ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Efficient Algorithms for Stochastic Optimization and Learning Algorithms for Uncertain Environments

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νικόλαος Ζαρίφης

Επιβλέπων: Δημήτρης Φωτάκης
Αναπληρωτής Καθηγητής ΕΜΠ

Αθήνα, Οκτώβρης '18

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Efficient Algorithms for Stochastic Optimization and Learning Algorithms for Uncertain Environments

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νικόλαος Ζαρίφης

Επιβλέπων: Δημήτρης Φωτάκης
Αναπληρωτής Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την Οκτώβρης '18

....	....	....
Δημήτρης Φωτάκης	Νικόλαος Παπασπύρου	Αριστείδης Παγουρτζής
Αναπληρωτής Καθηγητής ΕΜΠ	Αναπληρωτής Καθηγητής ΕΜΠ	Αναπληρωτής Καθηγητής ΕΜΠ

Αθήνα, Οκτώβρης '18

....

Νικόλαος Ζαρίφης
(Διπλωματούχος Ηλεκτρολόγος Μηχανικός & Μηχανικός Υπολογιστών Ε.Μ.Π.)

Περίληψη

Σε αυτή την διπλωματική μελετάμε πόσο αποδοτικά μπορούμε να λύσουμε προβλήματα όπως το Σακίδιο ή το Συντομότερο μονοπάτι , στην στοχαστική τους μορφή . Μελετάμε δύο τύπους αυτών τον προβλημάτων. Indyk et al, μελέτησαν το πρόβλημα του στοχαστικού σακιδίου με διάφορες παραλλαγές και έπειτα η Nikolova μελέτησε το στοχαστικό Σύντομο μονοπάτι. Και οι δυο δείξαν ότι όταν τα βάρη ακολουθούν bernoulli κατανομή τότε υπάρχει ένας QPTAS . Εμείς σε αντίθεση δείχνουμε πως μπορείς να επεκτείνεις τον αλγόριθμο και να βελτιώσεις τα αποτελέσματα σε εναν EPTAS.Κι επίσης είδαμε πιο γενικές παραλλαγές όπως χωρίς να έχεις υπόθεση για είδος κατανομής. Στην συνέχεια μελετήσαμε την δουλειά του Gupta et al όπου δίνουν αλγόριθμους που μαθαίνουν την βέλτιστη λύση σε συνδυαστικά προβλήματα σε αβέβαιο περιβάλλον και χρησιμοποιούμε τους αλγόριθμους τους στο συντομότερο μονοπάτι για πιο γενικές συναρτήσεις κόστους.
Λέξεις κλειδιά: Μάθηση, Στοχαστική Βελτιστοποίηση , Poisson προσέγγιση,Δειγματοληψία

Abstract

In this thesis, we study how efficiently we can solve certain problems like Knapsack or Shortest Path ,in their stochastic variation.We study two variants of these problems. Indyk et al, studied the Stochastic Knapsack with several configurations and Nikolova studied the Stochastic Shortest Path problem.Both of them showed that when the weights follow Bernoulli Random variables there exists a QPTAS. We, on the other hand, find an extension of their algorithm which can improve their results to an EPTAS algorithm. After we extend our configuration to a more general where we can have every probability distribution for our weights where we prove similar results.After that we study the work of [Gupta et al] where they give sampling algorithms for combination pure exploration and we use this work to develop algorithms when the weights are unknown Random Variables.

Keywords:Learning, Sampling, Stochastic Optimization, Poisson Approximation

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κ. Φωτάκη όπου με έκανε να αγαπήσω την θεωρητική πληροφορική.Μου έδειξε έναν νέο δρόμο στην επιστήμη και με βοήθησε να ξεκινήσω να πραγματοποιώ τα όνειρα μου.Κατά την διάρκεια της διπλωματικής μου προσέφερε καθοδήγηση, άπειρες συμβουλές , στήριξη σε προσωπικό επίπεδο κατά τις διαφορές δύσκολες περιόδους και μου αφιέρωσε πολύ χρόνο. Με βοήθησε να αναπτυχθώ επιστημονικά κι να είμαι σήμερα έτοιμος να επιτύχω τους στόχους μου.

Θα ήθελα να ευχαριστήσω επίσης την οικογένεια μου που με στήριξε, και αφιέρωσαν την ζωή τους για να σπουδάσω χωρίς να μου λείψει τίποτα.Τους φίλους μου και κοντινούς μου ανθρώπους , που έκαναν την ζωή μου ομορφότερη και ήταν κοντά μου στις δυσκολίες,ζήσαμε αξιοζήλευτες εμπειρίες , στιγμές που δεν θα ξεχάσω πότε και όπως λέει κι σε μια από της αγαπημένες μου ταινίες it's a wonderfull life: "no man is a failure who has friends" , τους (τυχαία σειρά,συγνώμη αν ξέχασα κάποιον): Νίκο Κ,Βασίλη Κ,Σαντρα Σ,Γιάννη Ο,Βασίλη Γ,Κατερίνα Κ,Δημήτρη Κ, Γιώργο Δ, Αργύρη Ο,Χρήστο Λ,Άσπα Σ, Σωτηρία Σ,Ευαγγελία Γ, Αλέξανδρο Τ, Κωνσταντίνα Δ,Μπέττυ Γ,Πέτρο και Δημήτρη Γ,Νίκο Η,Γρηγόρη Ι,Μελκον Χ και Αννα κ Ευη Α. Καθώς κι τους φίλους μου από το ΜΟΠ-corelab Στρατή κι Λούκα για τις ωραίες ώρες έρευνας κι συζήτησης. Φυσικά δεν πρέπει να ξεχνάμε κι τους ανθρώπους που χωρίς αυτούς δεν θα ήμασταν εδώ, γιαυτό ευχαριστώ τον κ. Λάκκα. Κι σαν αυτός ο κύκλος της ζωής μου έκλεισε με το τέλος αυτής της εργασίας, ένας νέος αρχίζει.Είμαι ευγνώμων σε όλους, γιατί χωρίς τον κάθε ένα ξεχωριστά τίποτα δεν θα ήταν το ίδιο.

Τέλος θα ήθελα να ευχαριστήσω τον κ. Λουλάκη για την βοήθεια που μου προσέφερε και για την εμπιστοσύνη του με τους διαγωνισμούς μαθηματικών. Τον κ Παγουρτζή και κ. Ζάχο για την καθοδήγηση,τις συμβουλές τους που μου προσέφεραν και για την προσπάθεια τους να κάνουν το εργαστήριο μια οικογένεια. Και τον κ. Παπασπύρου για τις συμβουλές του, βοήθεια και χρόνο που αφιέρωσε.

Περιεχόμενα

Κεφάλαιο 1

Introduction

Computer science is no more about computers than astronomy is about telescopes

E.Dijkstra

1.1 Purpose of this thesis

From an introductory class on Algorithms everyone is confident enough with the definition of optimization . Several problems can be thought as the fundamental of the discrete optimization. The first problem that will come in your mind, will be the shortest path problem. This is defined as the choice of certain edge in a way to construct a path from s to t and their weigh is minimized. Although this problem admits an easy solution when the weights are positive numbers ,it becomes harder as we erase assumptions.To provide an example as pertinent evidence , if we erase the assumption that the weights are positive values , the problem becomes harder, in fact it is a **NP-HARD** problem. In real life if we know the distance from a point to another we can not be sure about the time that is needed to cross it.There are several variables that make the environment noisy.For example if the edge is the road ,then you know that you can cross it in t time most of the times, but there are times when the traffic is highly increased which increase the time needed to cross it. Scientist can model this distortion of the environment with the use of probabilities.This leads to a new road of research rather than the deterministic one where you are fully aware of the environment to one that you have unlimitless information. This extend the definition of the deterministic optimization to the stochastic optimization.

Stochastic optimization refers to a collection of methods for optimizing an objective function when randomness is present.Randomness usually enters the problem either in the environment or in the objective function. Like the deterministic optimization there does no exist a single solution or method to solve every problem, but a toolbox is developed which made this problems tractable.

One common example of stochastic optimization is the bandit problem which is a general model for stochastic problems.A common implementation which makes it easier to understand this: imagine a gambler at a row of slot machines(arms) who has to decide in which one to play based on limitless information and he wants to maximize his rewards.In this problem you have to decide which one to play and how many times.Several scientists started to consider this problem as it is close to sequence testing [1].

Stochastic shortest path problem(first defined in [2]) is defined as the original shortest path but we do not know a priori our weights ,only a model for them. We can also define Stochastic Knapsack proportionally . In this configuration we only know a model of the weights of the items and we know their value which we want to maximize. These two problems are very similar as both admit a pseudo-polynomial algorithm.

One generalization of the stochastic optimization problems is the ability to see the item when you are choosing it and then adapt your actions. These problems are NP-hard as it is similar to Markov Decision process. In [3],[4] and [5] one can find several results in this direction for the stochastic knapsack problem.And in [6] several results were made in utility maximization.

On the other hand another problem that we face is what you can do when you can only sample the edges and know nothing about the environment. This problem is closed to hypothesis testing. In these problem you have a set of hypothesis and an oracle which you use to sample. The objective is to find a member of the hypothesis which is the closest to the one you sample. These problems have got the attention of many scientists nowadays. Several central limits theorems have risen in the surface which show a way to efficiently approximate several distributions.One of the first papers in central limits was the one developed by Barry-Essen [7] [8] which showed that the sum of several random variables can be approximated by a Gaussian distribution.Although this was the first central limit , several others have arisen.After that a scientist named Charles Stein provided a more formal method for proving central limits theorems which after the years turn to be the most important tool used by mathematicians. On [9] one can find several theorems about steins method, notably one of my favorites is that a sum of random variables $X = \sum X_i$ is close to a Gaussian with error rate of $1/Var(X)$.Paul and Gregory Valiant [10] showed a more general central limit theorem for multidimentional distributions and [Daskalakis et all] [11] improved their bounds.

Most of the times it may seem that Gaussian estimation are adequate for most of our applications but they only help when your model has proportionally big variance.On the other hand one may ask if there is a theorem for small variance models.In 1960's L.Cam [12] proved that one can approximate binomial distribution with poisson ones. After,by the use of Stein's method A. D. Barbour, L. Holst and S. Janson [13] proved that you can approximate small variance model using Poisson distribution who provide a sub-optimal bound.In later work Daskalakis and Papadimitriou [14] used the previous central limits theorems to provide an ϵ-Cover for PBDs.[Daskalakis et all] [11] also developed a more general framework to approximate using Poisson when the model is a vector.

Afterwards, statistical techniques used to solve several discrete problems using the minimum number of samples. This algorithms called 'natural' and it is believed that they are the only ones that will survive due to big data.In this settings we have a set of distributions and we want to find the one that is more closed to a known one.Although, there is a straightforward solution by using maximum likelihood, there are plenty of other which achieve a better overall result.The first problem that someone can describe is Finding the most biased coin. This problem is one of a wider collection of bandit problems.The problem first considered in 50s, in a work of Robbins[1] which derives strategies that asymptotically attain an average reward that converges in the limit to the reward of the best arm.Then [15] provided a review of the classical results on multi armed bandit problem. Lower bounds for different variants of the multi-armed bandit have been studied by several authors. For the expected regret model,the seminal work of Lai and Robbins [16] provides tight bounds in terms of the Kullback-Leibler divergence between the distributions of the rewards of the different arms.For several results in adversarial multi-armed bandit problem in which there are no probabilistic assumptions was first considered in [17] [18], and the regret grows proportionally to the square root of steps.Then in [19] it is showed in the probabilistic bandit problem that there are needed $O(n/\epsilon^2 \log(1/\delta))$ samples to find the ϵ-optimal arms with probability δ which improved the previous known bound of:$O(n/\epsilon^2 \log(n/\delta))$ .First , Martin et al [20] provide a lower bound of 2 bandits which can be viewed as the lower bound to distinguishes 2 coins which is $\Omega(\log(\delta^{-1})/\epsilon^2)$. Furthermore, Tsitsiklis et al , in [21] proved a more efficiently lower bound in terms of the arms $\Omega(\log(1/\delta)\sum_i 1/\Delta_i^2)$ where the Δ_i is the gap between the optimal arm and the ith. Richard Karp and Chandrasekaran then provided an suboptimal algorithm for finding the most biased coin [22]. After that [23] improved the previous results in a more general setting.After that much work has been done into finding the top-k arms,in [24].Several work has been done into finding the best arm [25] and then they improved their results in the best known today in [26].

Afterwards, one question that was raised is if these results can be used in combinatorial optimization problems. In [27] [Gupta et all] studied the way for someone to export the best basis in a matroid problem using as less samples as needed, after they extend their techniques in more general problems [28] which we are going to use in our problems.We are going to analyze the shortest path problem in a PAC module. We will show how to sub-optimal find paths that optimize certain functions.

1.1.1 Formal definition of our problems

In the first sections we consider a variant of some well known problems from combination optimization.The first one is the shortest path . In the original problem we need to find a path that minimizes the weight and connects two vertices s and t. In our variation the weights are not known a priori but we now a model for them.The model may be from knowing the distribution or knowing the variance and the mean value of each edge.We assume that the variables are independent, if they are not Nikolova [29] describes a way to transform dependent random variables to independent ones using covariance.Formally it is defined as:

Given a graph G(V,E,W) ,where $W = \{E_1, \dots E_n\}$ are the weights of each edge E, of |E| independent random variables following some distribution and an overflow probability δ, we have to choose a $P \subseteq W$ which forms a path eg: $(s, v_1, v_2, ...t)$ s.t: $P[\sum\limits_{i \in P} E_i > 1] \leq \delta$

On the other hand, we also consider a variation of the knapsack problem which is the ability to find a set of items that maximizes our profit and it is less than a threshold value of weight. In our version the profit is a deterministic variable (we know apriori the values of the profit) and we only know a model of the weigh values. We can define the problem as:

Given a set S of n independent random variables $\{X_1, \dots, X_n\}$ following some distribution with associated profits $\{p_1 \dots, p_n\}$ and an overflow probability δ, we have to choose a $S' \subseteq S$ which will maximize out profit $(Profit(S') \geq Profit(S''))$ and $P[\sum\limits_{i \in S'} X_i > 1] \leq \delta$.

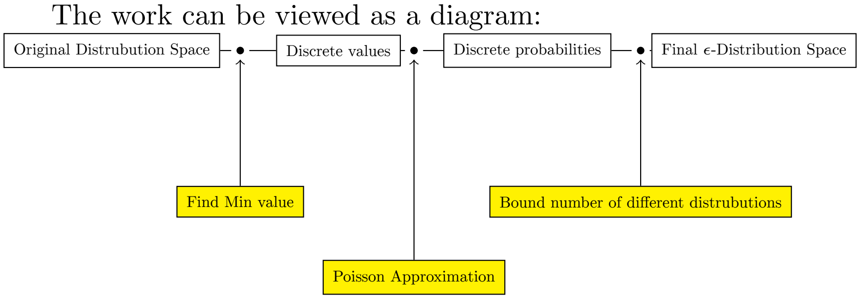
In our final section we are trying to solve the same problem but without knowing the model of our weights. In this setting we have a graph and we have an access to an oracle $O(e)$ which given an edge e it returns the weight of the edge in this particular moment(sampling). Our goal is to find the path that from s,t which optimizes the expected value of our objective function. Formally:

Given a graph G(V,E,W) ,where $W = \{E_1, \dots E_n\}$ are the weights of each edge E, of |E| independent random variables following some distribution and given access to a sampling oracle O , we have to choose a $P \subseteq W$ which forms a path eg: $(s, v_1, v_2, \dots .t)$ s.t we optimizes the $E(f(P))$ and we use the minimum number of samples.

1.1.2 Contribution

We provide a EPTAS for the stochastic shortest path problem and stochastic knapsack.We use a modification of Indyk,Goel [30] algorithm and we will use the Poisson approximation to round our values. Similar with us [31] provided a PTAS for the same problems using again the poisson approximation technique.

The work can be viewed as a diagram:



We also study the learning problem of stochastic path and we show lower bounds on their sample complexity. We are trying to extend the results of [28] when we have different variance between the edges and we provide bounds for our claims.We also provide a way to optimize more general objective functions with the usage of an oracle. We are going to show that there exist a bound of samples that no algorithms can be optimal by taking less. Several scientists before have proved lower bounds for these problems .Tsitsiklis et al [32] provided a very nice lower bound based on the difference on their mean value . Afterwards in [23] it proved that the lower bound of samples needed to classify a random variable to one of 2 categories in dominated by the log-likelihood lower bound , they proved that they need more than $\frac{1}{KL(D_1,D_2)}$.Other works include the ones from : [33] and [34].

Remark: Parallel to this work, we worked on online optimization. Specifically, we worked on optimizing facility location with switching cost.Although , we have some very interesting results we do not include them here as for the size limitations.This work will be uploaded to the arxiv in the following days.

1.2 Organisation of this thesis

In the chapter 2 2 we provide the mathematical background needed to understand our work. We provide several proofs for a variety of lemmas. Also , we provide a method for proving lower bounds due to [35].

In chapter 3 3 we show our results. We provide the proof of the EPTAS for the stochastic knapsack and shortest path. In chapter 4 4 we define the learning problem and we present a method by Gupta et al [28] and we provide several results for more general settings.

Κεφάλαιο 2

Mathematical Background

2.1 Introduction

Everyone can argue that without mathematics nothing is possible.Most of us are aware that most of rigorous proofs needs simple and fundamental mathematical equations.In these thesis we are going to prove most of our theorems with simple and most of the times elegant proofs. Therefore we are gonna introduce some basic lemmas and theorems which are vital to our proofs.We are going to describe the basic distances in probability theorem and several properties of them. Also we are going to describe the poisson approximation technique .Lastly we are going to describe Kaufman's work on the lower bound of bandit problem.

2.2 Useful Inequalities

We are going to start with the most basic inequality in probability theory.

Proposition 1. *Markov inequality : Let X be a no-negative Random Variable and $t > 0$ then*

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

We are going to provide an one line proof as it shows some interesting facts about Random Variables.

Απόδειξη. For every non-negative Random Variable it holds $t\mathbb{1}_{(X \geq t)} \leq X$ by using the linearity of expectation we get

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

□

Although Markov inequality is a fundamental inequality , it is not as strong as someone can expect. This lead mathematicians to a new toolbox of inequality which are called Chernoff Bounds.The first kind that we are going to use is the following.

Proposition 2. *[36] Chernoff Bounds : Let X_i be independent Random Variable satisfying $X_i \leq M$, $\forall 0 \leq i \leq n$. Let $X = \sum_{i=1}^{n} X_i$. Then*

$$\Pr[\sum_{i=1}^{n} X_i \leq \mathbb{E}[X] + \lambda] \leq \exp(-\frac{\lambda^2}{2(\sum_{i=1}^{n} \mathbb{E}[X_i^2] + M\lambda/3)})$$

The second kind that is needed is:

Proposition 3. *[36] Chernoff Bounds : Let X_i be non-negative independent Random Variable,we have the following bounds for the sum $X = \sum_{i=1}^{n} X_i$:*

$$\Pr[\sum_{i=1}^{n} X_i \leq \mathbb{E}[X] - \lambda] \leq \exp(-\frac{\lambda^2}{2\sum_{i=1}^{n} \mathbb{E}[X_i^2]})$$

2.3 Poisson Approximation

2.3.1 Metrics

One important question someone may ask is how can someone determinate how close 2 probability distributions are.All metric spaces have a metric that show the "distance" between 2 elements. But how can someone define this to more complex structures such as probability distributions.In fact there exists several "metrics" to show how close are 2 measures. We are going to explain 3 distances. The KL distance, Total Variation and the Kolmogorov distance.

Definition 1. *Total Variation distance: Let P_1,P_2 be 2 probability measures. We define Total Variation distance as:*

$$d_{tv}(P_1, P_2) = \sup_{A \in \Omega} |P_1(A) - P_2(A)|$$

One can see the total variation distance as the difference between the histograms of the two probability distributions.One can argue that d_{tv} is the closest one in approximation of the discrete problems as it provides the maximal error in some events.

Definition 2. *Kolmogorov distance: Let P_1,P_2 be 2 probability measures.Let F_1,F_2 be the cdf. We define Kolmogorov Distance as:*

$$d_k(P_1, P_2) = \sup_{A \in R} |F_1(A) - F_2(A)|$$

Kolmogorov distance shows the maximal difference between cfd of the random variables.This distance have several application in stochastic differential equations and in systems. As one may want to optimize a stochastic function under the constraints that $X < 1$ or the $\sum a_i X_i < 1$, if the constraints are exponentially large one can find a cover using this distance to make a polynomial one.

Lemma 1. *Let P_1,P_2 be 2 probability measures.Then it holds that:*

$$d_k(P_1, P_2) \leq d_{tv}(P_1, P_2)$$

Απόδειξη. One can argue that the set of events $A' = \{i | i \in \mathbb{R}, X \leq i\}$ is a subset of all probability events in Ω .

$$d_{tv}(P_1, P_2) = \sup_{A \in \Omega} |P_1(A) - P_2(A)| \geq \sup_{A \in A'} |P_1(A) - P_2(A)| = \sup_{A \in R} |F_1(A) - F_2(A)| = d_k(P_1, P_2)$$

$\square$

Definition 3. *Kullback–Leibler divergence: Let P_1,P_2 be 2 probability measures. We define KL divergence as:*

$$d_{kl}(P_1 \parallel P_2) = \int_X \log \frac{dP}{dQ} dP \qquad d_{kl}(P_1 \parallel P_2) = \int_{\mathbb{R}} P(x) \log \frac{P(x)}{Q(x)} dx$$

We are now present one important inequality for the distances which called data process inequality.

Lemma 2. *Let X, Y be two random variables on S. Let f be a function on S.Then :*

$$d_{tv}(f(X), f(Y)) \leq d_{tv}(X, Y)$$

There are a lot of ways to prove this, but one easy is to think that f will decrease the support of the random variable thus the difference may be smaller but never bigger.

Lemma 3. *Pinsker's inequality: Let P_1,P_2 be 2 probability measures then it holds:*

$$d_{kl}(P_1 \parallel P_2) \leq 2d_{tv}^2(P_1, P_2)$$

Lemma 4. *Let P_1, P_2 be 2 normal random variables each with m_1, m_2 mean and σ_1, σ_2 variance. Then:*

$$d_{kl}(P_1 \parallel P_2) \leq 1/2((\frac{\sigma_2^2 - \sigma_1^2}{\sigma_1^2})^2 + \frac{|m_2 - m_1|^2}{\sigma_1^2})$$

Απόδειξη.

$$\begin{aligned} d_{kl}(P_1 \parallel P_2) = (-\log(\sigma_2/\sigma_1) - 1/2 + \frac{|m_2 - m_1|^2 + \sigma_1^2}{2\sigma_2^2}) \leq 1/2(\frac{\sigma_2^2}{\sigma_1^2} - \log \frac{\sigma_2^2}{\sigma_1^2} - 1)\frac{|m_2 - m_1|^2}{2\sigma_2^2} \\ \leq 1/2((\frac{\sigma_2^2 - \sigma_1^2}{\sigma_1^2})^2 + \frac{|m_2 - m_1|^2}{\sigma_1^2}) \quad (2.1) \end{aligned}$$

where in last inequality we used that $x - \log x - 1 \leq (1 - x)^2$ $\square$

2.4 Poisson Approximation

Theorem 1. *Poisson Approximation : Let $X_1, \ldots, X_n$ be independent Bernoulli Random Variables $Be(p_i)$.Let $X = \sum_{i=1}^{n} X_i$ and $\lambda = \sum_{i=1}^{n} p_i$ Then*

$$d_{tv}(X, Poi(\lambda)) \leq \frac{\sum_{i=1}^{n} p_i^2}{\sum_{i=1}^{n} p_i}$$

This nice bound is proved using Steins Method. It tells us that if the parameters p_i are small enough then we can think our random variable as a Poisson random variable.Although this is a general tool which can be generalized for all random variables with small mean. In our sections we will use a more important tool which is for PMD random variables.

Lemma 5. *[11] For any $c \leq \frac{1}{2k}$, given access to the parameter matrix R for an $(n,k) - PMD$ M^R we can efficiently construct another $(n,k) - PMD$ $M^{\hat{R}}$, such that ,for all i,j , $\hat{R}(i,j) \notin (0,c)$, and*

$$d_{tv}(M^R, M^{\hat{R}}) < O(c^{1/2}k^{5/2}\log^{1/2}\frac{1}{ck})$$

2.5 Proving Lower Bounds

A fundamental problem in Learning theory is how can someone derive lower bounds on the number of samples for several problems. Although this is a very difficult problem in the previous years there has been a lot of scientific research which lead us to several theorem and frameworks that can help us to derive lower bounds. In our work we will mostly pay attention to the work of [Kauffman, et all] [35]. This work gave some interesting bounds in bandit problems which is a generalization of many other problems. We are going to present her proof shortly. From now on we will denote $d(a,b) = a\log(a/b) + (1-a)\log((1-a)/(1-b))$
Let a,a' be two bandit models or as in our problem , be a set of edges. In each time step t our algorithm will choose to sample for an edge i and we are going to name this action $A_t = i$ and Z_t be the outcome of each step. Lastly, let the f_a and $f_{a'}$ be the density of each edge. One can introduce the log-likelihood ratio by:

$$L_t = \sum_{a=1}^{K}\sum_{s=1}^{t} \mathbb{1}_{(A_s=i)} \log(\frac{f_a(Z_s)}{f'_a(Z_s)}) \tag{2.2}$$

One key lemma to the change of distribution is the following:

Lemma 6. *Let σ be ant stopping time with respect F_t . For every event $E \in F_\sigma$ then*

$$\Pr_{v'}(E) = \mathbb{E}_v[\mathbb{1}_E exp(-L_\sigma)]$$

This lemma shows a way to connect the 2 different distributions.

Lemma 7. *Let A be an algorithm that runs in n arms and let $C = (a_i)_{i=1}^n$ and $C' = (a'_i)_{i=1}^n$ be two sequences of n arms. Let the Random Variable t_i denote the number of samples taken from i-th arm. For any event E in F_t where t is a stopping time with respect the filtration $\{F_t\}_{t\geq 0}$, it holds*

$$\sum_{i=1}^{n} \mathbb{E}_{A,C}[t_i] KL(a_i, a'_i) \geq d(P_{A,C'}[E], P_{A,C}[E])$$

Απόδειξη. With a first glance this lemma states that when the information entropy of two distribution is close to each other , we need more samples to separate them. This derives from the fact that the Maximum Likelihood algorithm needs these much samples to be able to output the correct solution. Because the Expected value of of the log-likehood ratio of two random variables is in fact the KL distance. We are going to show this using the Wald's lemma.

$$L_t = \sum_{a=1}^{K} \sum_{s=1}^{N_a(t)} \log(\frac{f_a(Y_{a,s})}{f'_a(Y_{a,s})})$$

Also we have that

$$\mathbb{E}[\log(\frac{f_a(Y_{a,s})}{f'_a(Y_{a,s})})] = KL(V_a, V'_a)$$

Lemma 8. *Wald's Lemma: Let $X = \sum_{i=1}^{N} X_i$ where X_i are iid and N is a non negative random variable then:*

$$\mathbb{E}[X] = \mathbb{E}[N]\,\mathbb{E}[X_1]$$

It is easy to see that using 8 we have that

$$\mathbb{E}[L_t] = \sum_{a=1}^{K} N_a(t) KL(V_a, V'_a) \tag{2.3}$$

It is clear that our statements holds if $\Pr_v(E) = \{0, 1\}$ as the right hand side is equal to zero and the left hand size is non negative (KL distance is non negative and the samples are a positive number).

$$\begin{aligned}\Pr_{v'}(E) = \mathbb{E}_v[\mathbb{1}_E exp(-L_\sigma)] &= \mathbb{E}_v[\mathbb{1}_E\, \mathbb{E}_v[exp(-L_\sigma)|\mathbb{1}_E]] \quad \textit{(using Jensen inequality)} \\ &\leq \mathbb{E}_v[\mathbb{1}_E exp(\mathbb{E}_v[-L_\sigma|\mathbb{1}_E])] = \mathbb{E}_v[\mathbb{1}_E exp(\mathbb{E}_v[-L_\sigma|\mathbb{1}_E]\mathbb{1}_E)] \\ &= \mathbb{E}_v[\mathbb{1}_E exp(-\mathbb{E}_v[L_\sigma|E])] = exp(-\mathbb{E}_v[L_\sigma|E])P_v[E]\end{aligned}$$

We also used the very important property of expected value that is $\mathbb{E}[\mathbb{E}[X|Y]] = \mathbb{E}[X]$

And the same holds for E^c hence :

$$\mathbb{E}_v[L_\sigma|E] \geq \log\frac{P_v[E]}{P_{v'}[E]} \quad \mathbb{E}_v[L_\sigma|E^c] \geq \log\frac{P_v[E^c]}{P_{v'}[E^c]} \tag{2.4}$$

Therefore,using the total probability equality:

$$\begin{aligned}\mathbb{E}_v[L_\sigma] = \mathbb{E}_v[L_\sigma|E]P_v[E] + \mathbb{E}_v[L_\sigma|E^c]P_v[E^c] \geq P_v[E]\log\frac{P_v[E]}{P_{v'}[E]} + P_v[E^c]\log\frac{P_v[E^c]}{P_{v'}[E^c]} \\ = d(P_v[E], P_{v'}[E])\end{aligned}$$

□

Κεφάλαιο 3

Solving stochastic optimization problems

A straight line may be the shortest distance between two points, but it is by no means the most interesting

Doctor who (1971)

3.1 EPTAS for Stochastic Knapsack

3.1.1 Previous Work

At [37] [Tardos et al] defined the stochastic load balancing,stochastic bin packing and stochastic knapsack.They obtained an $O(1)$ - approximation algorithm for arbitrary distributions, under the assumption that $E = [m_i^X]$ is defined (Converges).Their techniques include showing how to approximate under the assumption of Bernoulli trials .Furthermore they showed that if they can solve that they can transform arbitrary distributions on an equivalent problem with Bernoulli trials and the result follows. Moreover Indyk and Goel [30], showed how to construct a PTAS for stochastic knapsack under the assumption of Exponential Jobs, furthermore they provided a QPTAS for the bernoulli trials.Their techniques included to provide a polynomial or subexpomential distribution space. We adapt many of their techniques in our work. Nikolova at her thesis [29] provided a new way to solve these problems. First of all , she defined the stochastic shortest path problem which she showed that it is NP-Hard to solve.Also, she showed a way to find the optimal solution in more general problems using the work of Cartensen[38] and showed some problems that are easy to find a solution. Using sublevel sets, she showed that if there exist an oracle that can answer if there exists a solution in this sublevel set, then she can construct an FPTAS. She showed this for a more general type of problems.These problems were quasi-convex maximization which is know to be NP-Hard. She used the work of Yannakakis and Papadimitrioy [39] to provide an algorithm for the shortest path under the assumption of Normal Distribution; she used $\epsilon-$pareto curves to approximate the oracle.Moreover , she also provided a QPTAS for the shortest path under the assumption of Bernoulli Trials.

In a parallel work with us, [31] Lin,Yuan , proposed a PTAS for Arbitrary random variables for the shortest path and knapsack problem. Their techniques involve like ours the poisson approximation and several tail inequalities . On the other hand our results provide an EPTAS which is the best possible .Unless P=NP there does not exist a FPTAS algorithm for this problem.

3.1.2 Introduction

In the following sections we are going to provide some algorithms which solve efficiently the Stochastic Knapsack problem and the Stochastic Shortest Path. We are gonna start by showing how we can discretize the space and build a polynomial state space for our dynamic programming algorithm.First of all we are going to prove that if the random variable has small expected value then we can use it as a deterministic value as the error is small.Then we are going to discretize the remaining random variables in a $O(poly(1/\epsilon))$ values. Then we are going to use the poisson approximation to prove that discrete random variables we small probabilities can be approximate by a poisson variable which will lead as an low bound for the probability which will lead us to an easy discretization. Lastly we are going to combine this results to build the state space and we will show the different algorithms that are needed to solve its problem.Furthermore it is easy to see that we can normalize our sum and if a value is bigger than 1 transform it to $1+\alpha$ which will lead to an overflow. We are going to start with some basic definitions of our problems.

Definition 4. *Given a set S of n independent random variables $\{X_1, \ldots, X_n\}$ following some distribution with associated profits $\{p_1 \ldots, p_n\}$ and an overflow probability δ, we have to choose a $S' \subseteq S$ which will optimize the program:*

$$\begin{aligned} &\textit{maximize} && \sum_{i \in S'} p_i \\ &\textit{s.t.} && P[\sum_{i \in S'} X_i > 1] \leq \delta \end{aligned}$$

Definition 5. *Given a graph G(V,E,W) where $W = \{E_1, \ldots E_n\}$ of n independent random variables following some distribution and an overflow probability δ, we have to choose a $P \subseteq W$ which form a Path s.t:*

$$P[\sum_{i \in P} E_i > 1] \leq \delta$$

These problems are NP-Hard to solve which is the reason that we are going to show a way to approximate efficiently.

Definition 6. *Given a set S of n independent random variables $\{X_1, \ldots, X_n\}$ following some distribution with associated profits $\{p_1 \ldots, p_n\}$, an overflow probability δ and a constant $\epsilon \geq 0$, we have to choose a $S' \subseteq S$ which will optimize the program:*

$$\begin{aligned} &\textit{maximize} && \sum_{i \in S'} p_i \\ &\textit{s.t.} && P[\sum_{i \in S'} X_i > 1+\epsilon] \leq \delta(1+\epsilon) \end{aligned}$$

Definition 7. *Given a graph G(V,E,W) where $W = \{E_1, \dots E_n\}$ of n independent random variables following some distribution , an overflow probability δ and a constant $\epsilon \geq 0$, we have to choose a $P \subseteq W$ which form a Path s.t:*

$$P[\sum_{i \in P} E_i > 1 + \epsilon] \leq \delta(1 + \epsilon)$$

In the following sections we are going to show how to construct EPTAS algorithms for these problems.We are going to define what is an EPTAS algorithm.

Definition 8. *We say that the algorithm is an Efficient PTAS when it is a PTAS and its running time is $O(n^c)$ for some constant c and independent of ϵ.*

This ensures that an increase in problem size has the same relative effect on runtime regardless of what ϵ is being used; however, the constant under the big-O can still depend on ϵ arbitrarily.

3.1.3 EPTAS Knapsack and SSP with Bernoulli Trials

In this section we are going to assume that our random variables are Bernoulli Trials. These variables takes a certain value α_i with some probability and zero otherwise, wlog values are non negative.Let all $\alpha_i \in [0, 1 + \alpha]$ where α is a small constant.Therefore when we get $1 + \alpha$ we have overflow our weight. Our techniques involve on how we can create an efficient cover for our variables and using this to find the optimal solution of our problem.

We are going to show that when our a_i are small enough we can assume that the wight is deterministic and its value is its expected value.

Lemma 9. *Let $t = \frac{\epsilon^3}{\log 1/\epsilon\delta}$ and let A be a subset of X_i where which $\alpha_i \leq t$ and $m = \sum\limits_{i \in A} E[X_i]$ then*

$$\Pr[\sum_{i \in A} X_i \geq (1 + \epsilon)m] \leq \epsilon\delta \tag{3.1}$$

Απόδειξη. Using chernoff bounds 2 , we set $\lambda = \epsilon E[X]$ and we have:

$$\Pr[\sum_{i \in A} X_i \geq (1 + \epsilon)E[X]] \leq \exp(-\frac{(\epsilon E[X])^2}{2(\sum E[X_i^2] + \epsilon E[X]M/3)})$$

Using that $M = t$,we have to prove that :

$$\exp(-\frac{(\epsilon E[X])^2}{2(\sum E[X_i^2] + \epsilon t E[X]/3)}) \leq \epsilon\delta \Leftrightarrow$$

$$\frac{(\epsilon E[X])^2}{2(\sum E[X_i^2] + \epsilon t E[X]/3)} \geq \log 1/\epsilon\delta$$

$$E[X]\frac{1}{2(\sum E[X_i^2]/E[X] + \epsilon t/3)} \geq \frac{\log 1/\epsilon\delta}{\epsilon^2} \tag{3.2}$$

We also have that :

$$E[X]\frac{1}{2(\sum E[X_i^2]/E[X] + \epsilon t/3)} \geq E[X]\frac{1}{2t(1 + \epsilon/3)} \tag{3.3}$$

Using (3.2) and (3.3) it is clear that we have to find a value t such as:

$$E[X]\frac{1}{2t(1+\epsilon/3)} \geq \frac{\log 1/\epsilon\delta}{\epsilon^2}$$

Using the fact that $E[X] \geq \epsilon$ we have:

$$\begin{aligned}\epsilon\frac{1}{2t(1+\epsilon/3)} &\geq \frac{\log 1/\epsilon\delta}{\epsilon^2} \Leftrightarrow \\ t &\leq \frac{2\epsilon^3}{\log 1/\epsilon\delta}\end{aligned}$$

□

Now we have to deal with the other values. We are going to truncate them to a value of the form $t(1+\epsilon)^k$ for some k. Let X_i' be the truncate random variable of X_i where if $\alpha_i \in [t(1+\epsilon)^k, t(1+\epsilon)^{k+1})$ then $a_i' = t(1+\epsilon)^k$. It is clear that for a subset A

$$\sum_{i\in A} X_i' \leq (1+\epsilon)\sum_{i\in A} X_i \tag{3.4}$$

Thus the number of different a_i' is order of $O(\log 1/t/\epsilon)$.

Now we are going to use Poisson Approximation 1 to bound the smallest probability. First of all we start we a simple lemma.

Lemma 10. *Let $a, b > 0$:*

$$d_{tv}(Poi(a), Poi(b)) \leq \frac{1}{2}(e^{|a-b|} - e^{-|a-b|})$$

Let's say we have a set A of the chosen X_i . We denote $X_{\alpha_i} = \sum_{j\in A:\Pr[X_j=\alpha_i]>0} X_j/\alpha_i$. We are going to prove that if the probability of a non zero event is small enough then we can think our variable as a Poisson with a parameter our expected value. Let C_i be a set of indexes where $\Pr[X_i > 0] \leq 1/k$. Using 1 we have:

$$d_{tv}(\sum_{j\in C_i} X_j, Poi(\sum_{j\in C_i} p_j)) \leq \frac{\sum_{j\in C_i} p_j^2}{\sum_{j\in C_i} p_j} \tag{3.5}$$

$$\leq \frac{1/k\sum p_i}{\sum p_i} \tag{3.6}$$

$$\leq 1/k \tag{3.7}$$

If we set $S = \sum_{j\in C_i} p_j$ and $r = \lfloor\frac{S}{1/k}\rfloor$ then we set $p_i' = 1/k\ \forall i \leq r$ and the rest $p_i' = 0$.

From 10, we can bound the distance by:

$$\begin{aligned}d_{tv}(Poi(\sum_{j\in C_i} p_j), Poi(\sum_{j\in C_i} p_j')) &\leq 1/2(e^{1/k} - e^{-1/k}) \\ &\leq 1.5/k\end{aligned}$$

Then we use the triangle inequality and we have:

$$d_{tv}(\sum X_i, \sum X_i') \leq 3.5/k \tag{3.8}$$

If we set $k = \frac{3.5}{\epsilon\delta}$ we have out result.

Our last step is to bound the number of different probabilities p_i. We again are going to round the values in a form of $\epsilon\delta(1+L)^k$. which means that $p_i' = \epsilon\delta(1+L)^k$ if $p_i \in [\epsilon\delta(1+L)^k, \epsilon\delta(1+L)^{k+1})$. We have to find a value of L that is sufficient for our bounds. First of all we can easily see that if A is a feasible solution then :

$$P[\sum_{i\in A} X_i > 1] \leq \delta \leq 1/(1+\epsilon) \tag{3.9}$$

The last inequality is from the fact that $(1+\epsilon)\delta < 1$, if not then every solution will be feasible.

We are going to prove a useful lemma that bounds the expected number of a feasible solution in terms of ϵ.

Lemma 11. *Let $E[X]$ be the mean value of $X = \sum\limits_{i\in A} X_i$ then for any feasible solution it holds that :*

$$E[X] < c\log(1/\epsilon)$$

Απόδειξη. Using chernoff bounds 3 by setting $\lambda = E[X] - 1$ we have:

$$\Pr[\sum_{i\in A} X_i \leq 1] \leq \exp(-\frac{(E[X]-1)^2}{2\sum\limits_{i\in A} E[X_i^2]})$$

And using (3.9) we get:

$$\frac{\epsilon}{1+\epsilon} \leq \Pr[\sum_{i\in A} X_i \leq 1] \leq \exp(-\frac{(E[X]-1)^2}{2\sum\limits_{i\in A} E[X_i^2]})$$

$$\Rightarrow \frac{\epsilon}{1+\epsilon} \leq \exp(-\frac{(E[X]-1)^2}{2\sum\limits_{i\in A} E[X_i^2]})$$

$$\Leftrightarrow \log\frac{\epsilon+1}{\epsilon} \geq \frac{(E[X]-1)^2}{2\sum E[X_i^2]}$$

$$\geq \frac{(E[X]-1)^2}{2(\alpha+1)\sum E[X_i]} = c_i(E[X] - \frac{1}{E[X]} - 2)$$

$$\Rightarrow c\log 1/\epsilon \geq E[X] \tag{3.10}$$

□

And now we are going to use the previous lemmas to prove our last lemma that will complete our cover.

Lemma 12. *Let $X = \sum_{i \in A} X_i$ be a feasible solution, and $E[X] \leq T$. Then for every other sequence $X' = \sum_{i \in A} X'_i$ such that $p'_i \leq p_i(1+L)$ we have*

$$P[X' \geq 1+\epsilon] \leq P[X \geq 1] + LT/\epsilon \tag{3.11}$$

Απόδειξη. Let $X'_i =_D X_i + X''_i$,(This means that they have identical distributions).And we set X''_i to be a random variable such as $P[X''_i = a_i | X_i = 0] = \frac{p'_i - p_i}{1-p_i}$ and $P[X''_i = a_i | X_i = a_i] = 0$. Then $P[X''_i = a_i] = p'_i - p_i \leq Lp_i$. From union-bound inequality we have :

$$P[X' \geq 1+\epsilon] \leq P[X \geq 1] + P[X'' \geq \epsilon]$$

Then using Markov inequality we have :

$$P[X'' \geq \epsilon] \leq E[X'']/\epsilon \leq LT/\epsilon$$

. □

We need $LT/\epsilon \leq \epsilon\delta$ thus we set $L = \epsilon^2\delta/T = \frac{\epsilon^2\delta}{c\log(1/\epsilon)}$. The order of different values of p_i is $O(\frac{\log(1/\epsilon)\log(1/\epsilon\delta)}{\epsilon^2\delta})$

Now we have all that we need to build our algorithm by showing how the DP table is constructed. We are going to start with the shortest path case. We are going to fill the dynamic programming table with rows correspond to the weight of small edges S , all the different types of large edges L and the poisson variables P. The columns of DP will correspond to vertices.We observe that ϵ/n can be an addictive error of the small variables as it increase the error rate by ϵ thus let $S = i\epsilon/n$ where $i = 0 \ldots n/\epsilon$ .For the poisson variables we are going to set $P_i = j\epsilon/n$ where $j = 0 \ldots n/\epsilon$ and the i correspond to a different value of α_i .Now we are going to calculate the different types of Large edges. By 11 we have that our large variables can not exceed a certain threshold of expected value.

Lemma 13. *For a chosen value of ϵ and δ we can have at most $O(\frac{\log(1/\epsilon)\log(1/\epsilon\delta)}{\epsilon^4\delta}))$ large edges.*

Απόδειξη. From 11,9and poison method we have that if we have A different values it should at worst satisfy the following inequality

$$\begin{aligned} A * t * p_{min} &\leq c\log(1/\epsilon) \\ \Leftrightarrow A &\leq c\frac{\log(1/\epsilon)\log(1/\epsilon\delta)}{\epsilon^4\delta}) \end{aligned} \tag{3.12}$$

□

We also need the number of different tuples (α_i, p_i) which is with a simple combinatorial argument $B = O(\frac{\log(1/\epsilon)^2\log(1/\epsilon\delta)}{\epsilon^3\delta})$.

Finally we conclude that the number of different types is $B^A = f(\epsilon, \delta)$.
The algorithm is using a similar idea from Bermann Ford Algorith. [40] [41]

Algorithm 1: Shortest Path algorithm

```
Data: G = (V, E) , constant δ , constant ε, start node u,end node v .
begin
    Let G' = (V, E') be the discetized graph.
    for i = 1 to n do
        for (u, v) ∈ E do
            foreach ({S, L, P}, u) do
                if (u, v) is a small edge then
                    if DP({S − e, L, P}, v) is non empty then
                        (DP({S, L, P}, u)) ⟵ v
                    end
                end
                if (u, v) is a small propability edge a_i then
                    if DP({S, L, P − e}, v) is non empty then
                        (DP({S, L, P}, u)) ⟵ v
                    end
                else
                    if DP({S, L − e, P}, v) is non empty then
                        (DP({S, L, P}, u)) ⟵ v
                    end
                end
            end
        end
    end
    output ⟵ CalculatePath(DP)
end
```

Theorem 3.1.1. *Given a constant $\epsilon \geq 0$ there exist a EPTAS for finding a set P for the stochastic shortest path such as:*

$$P[\sum_{i \in P} E_i > 1 + c_1\epsilon] \leq \delta(1 + c_2\epsilon)$$

Απόδειξη. It is easy to see that the loop in (3) and (4) will run n^3 and update $f(\epsilon, \delta)P(\epsilon) * n^2$ values where $P(\epsilon)$ is the poisson and smallest variable space thus we have $n^5 g(\epsilon\delta)$ running time which lead us to a EPTAS. □

We observe that for the Knapsack problem we can use the idea for DP but with different columns.

Algorithm 2: Knapsack algorithm

```
   Data: (W_i, Z_i) item values, constant δ , constant ε.
1  begin
2      Let (W'_i, Z_i) be the discetized values.
3      for i = 1 to n do
4          foreach {S, L, P} do
5              if W'_i is a small item then
6                  if DP({S − W'_i, L, P}) is non empty then
7                      (DP({S, L, P}, u)) ⟵
                         max(DP({S − W'_i, L, P}) + Z_i, DP({S, L, P}))
8                  end
9              end
10             if W'_i is a small propability item a_i then
11                 if DP({S, L, P − W'_i}) is non empty then
12                     (DP({S, L, P})) ⟵
                         max(DP({S, L, P − W'_i}) + Z_i, DP({S, L, P}))
13                 end
14             else
15                 if DP({S, L − W'_i, P}) is non empty then
16                     (DP({S, L, P}, u)) ⟵
                         max(DP({S, L − W'_i, P}) + Z_i, DP({S, L, P}))
17                 end
18             end
19         end
20     end
21     output ⟵ CalculateKnap(DP)
22 end
```

Theorem 3.1.2. *Given a constant $\epsilon \geq 0$ there exist a EPTAS for finding a set S' such as:*

$$\begin{aligned} & \textit{maximize} && \sum_{i \in S'} p_i \\ & \textit{s.t.} && P[\sum_{i \in S'} X_i > 1 + \epsilon] \leq \delta(1 + \epsilon) \end{aligned}$$

Απόδειξη. It is easy to see that the loop in (3) will run $n$and update $f(\epsilon, \delta)P(\epsilon) * n^2$ (4) values where $P(\epsilon)$ is the poisson and smallest variable space thus we have $n^3 g(\epsilon\delta)$ running time which lead us to a EPTAS. □

3.2 EPTAS Knapsack and SSP with General Random Variables

In this section we are going to show how to extend our results to a more general problem.In reality , we may have more complex models than simple bernoulli variables. Also there are case where the model of distribution is different for each weight as for example one way may be a poisson random variable and the other be a gamma distributed random variable. We are going to show that we can extend most of our proofs to work on this case.

First we are going to show how we can round all the variables in a way that the error will be small. For every random variable X let $f_X(x)$ be the pdf this means that $f_X(A) = \int_A f_X(x)dx$. Our rounding scheme assumes that we can create variables X' such that $p_{X'}(x) = \int_{x-\epsilon/n}^{x} f_X(y)dy$. This rounding increase the error with ϵ. This means that for a subset A we have

$$\sum_{i \in A} X_i' \leq \sum_{i \in A} X_i + \epsilon$$

Although we have created a discrete distribution for every variable with at most n/ϵ support it's clear that this is not enough to solve the entire problem.

It is clear that in this environment is not as easy as before to decrease the support . This is because there are two many options for every mean in the small support. Lucky for us there exist a way to transform the variable to a variable with smaller support.

Again we choose t as in 9. We are going to split the support in two spaces one where $X_i < t$ which we are gonna symbolize C and U where $U = (support(X) - C) \cup \{0\}$.

Also we are gonna denote $X_i|_C$ the projection of the variable X_i in C resp. $X_i|_C =_D X_i \mathbb{1}_{(X_i \leq t)}$.

Now we are going to prove one important lemma.

Lemma 14. *Let X_i be a random variable. Let X_i' be a random variable such as $X_i'|_C = 0, t$ and $\Pr[X_i'|_C = t] = E[X_i'|_C]/t$. Then:*

$$\Pr[\sum_{i \in A} X_i' > 1 + \epsilon] \leq \Pr[\sum_{i \in A} X_i > 1 + \epsilon] + \epsilon\delta$$

Απόδειξη.

$$\begin{aligned}\Pr[\sum_{i\in A} X_i' > 1+\epsilon] &= \sum_{A\subset[n]} P(A)P(\sum_{[n]-A} X_i + \sum_{A} X_i'|_C > 1+\epsilon)\\ &\le \sum_{A\subset[n]} (P(A)P[\sum_{[n]-A} X_i + \sum_{A} X_i|_C > 1+\epsilon] + \epsilon\delta)\\ &= P[\sum X_i > 1+\epsilon] + \epsilon\delta\end{aligned}$$

□

And we round the remaining support by $t(+1\epsilon)^k$ as previous and we again have $O(\log 1/t/\epsilon)$.

Now we have to bound the lowest probability. We are going to think our random variables as vector which will allow us to random them as PMD and then using data processing inequality we are going to have the same result for our setting.

Lemma 15. *There exist a constant $c > 0$ such that for every discrete random variable X_i with support k there exist a X_i' which has as the smallest probability the value of c and also*

$$d_{tv}(X_i, X_i') < O(c^{1/2}k^{5/2}\log^{1/2}\frac{1}{ck})$$

Απόδειξη. Let V_i be a vector which take with probability p_{ij} the canonical vector e_j of R^k . Then by 5 we can create a vector V_i' such that

$$d_{tv}(V_i, V_i') < O(c^{1/2}k^{5/2}\log^{1/2}\frac{1}{ck}) \tag{3.13}$$

Then let $f(x) = c^t x$ where $c = (a_i)_{i=1}^k$ then by 2 we have

$$d_{tv}(f(V_i), f(V_i')) \le d_{tv}(V_i, V_i') \tag{3.14}$$

But we see that $f(V_i) = X_i$, so by (3.13) and (3.14) we conclude our proof. □

We choose as $c = \frac{(\epsilon\delta)^{2+c_i}}{k^5}$ where c_i is a small non zero positive value, we have proved that $k = O(\log 1/t/\epsilon)$.

We are gonna again round the probability of every value with the following way. We gain are going to round the values in a form of $c\delta(1+L)^j$. which means that $p_i' = c(1+L)^j$ if $p_i \in [c(1+L)^j, c(1+L)^{j+1})$. So we need to prove our last lemma which is analogous to the 12.

Lemma 16. *Let $X = \sum_{i\in A} X_i$ be a feasible solution, and $E[X] \le T$. Then for every other sequence $X' = \sum_{i\in A} X_i'$ such that $p_i' \le p_i(1+L)$ we have*

$$P[X' \ge 1+\epsilon] \le P[X \ge 1] + LT/\epsilon \tag{3.15}$$

Απόδειξη. Let $X'_i =_D X_i + X''_i$ .And we set X''_i to be a random variable such as $P[X''_i = a_i | X_i = 0] = \frac{p'_i - p_i}{p_0}$ and $P[X''_i = a_j | X_i = a_j] = 0$ for all j.We set $p_0 = 1 - \sum_{j=1}^{k} p_{ij}$. Then $P[X''_i = a_i] = p'_i - p_i \leq Lp_i$. .From union-bound inequality we have :

$$P[X' \geq 1 + \epsilon] \leq P[X \geq 1] + P[X'' \geq \epsilon]$$

Then using Markov inequality we have :

$$P[X'' \geq \epsilon] \leq E[X'']/\epsilon \leq LT/\epsilon$$

. □

And now from 11 we have that we can set $L = O(\frac{\epsilon^2 \delta}{\log(1/\epsilon)})$. This lead us with $O(\log 1/c/L)$ different values for the p_i of each X_i.

Now we are going to calculate the cardinality of large variables.

Lemma 17. *For a chosen value of ϵ and δ we can have at most $O(\frac{\log(1/\epsilon)\log(1/\epsilon\delta)}{(\epsilon\delta)^{6+c}})$ large edges.*

Απόδειξη. From 11,9 and 15 we have that if we have A different values it should at worst satisfy the following inequality

$$A * t * p_{min} \leq c' \log(1/\epsilon)$$
$$\Leftrightarrow A \leq O(\frac{\log(1/\epsilon)\log(1/\epsilon\delta)}{(\epsilon\delta)^{6+c}})$$

□

Now we have to find the different number of vectors, which are $B = k^{\log(1/c)/L}$. Which lead us to the that all the different values of large vectors are $f(\epsilon, \delta) = B^A$.

The algorithm is similar to the one in the previous section. The difference is the line (3) where we change our values to adapt the smallest value , this means that we change every value to one that allows one to approximate with small error like the lemma 14.

Theorem 3.2.1. *Given a constant $\epsilon \geq 0$ and a set S with Random Variables there exist a EPTAS for finding a set P for the stochastic shortest path such as:*

$$P[\sum_{i \in P} E_i > 1 + c_1\epsilon] \leq \delta(1 + c_2\epsilon)$$

Απόδειξη. It is easy to see that the loop in (6) and (7) will run n^3 and update $f(\epsilon, \delta)P(\epsilon) * n^2$ values where $P(\epsilon)$ is the poisson and smallest variable space ,moreover, the line (3) runs on the edges which means n^2 thus we have $n^3 g(\epsilon\delta)$ running time which lead us to a EPTAS. □

Again the algorithm is quite similar to the previous section but with the adaption of the 14.

Algorithm 3: Shortest Path algorithm

Data: $G = (V, E)$, constant δ , constant ϵ, start node u,end node v .

begin
    Let $G' = (V, E')$ be the discetized graph.
    foreach $e \in E'$ **do**
        Set $\Pr[X'_e|_C = t] = E[X'_e|_C]/t$
    end
    for $i = 1$ *to* n **do**
        for $(u, v) \in E$ **do**
            foreach $(\{L, (P_i)_{i=1}^k\}, u)$ **do**
                if (u, v) *is a small propability edge* a_i **then**
                    if $DP(\{L, P = (\dots, P_i - e, P_{i+1}, \dots)\}, v)$ *is non empty* **then**
                        $(DP(\{L, P\}, u)) \longleftarrow v$
                    end
                end
                if $DP(\{L - e, P\}, v)$ *is non empty* **then**
                    $(DP(\{L, P\}, u)) \longleftarrow v$
                end
            end
        end
    end
    $output \longleftarrow CalculatePath(DP)$
end

Algorithm 4: Knapsack algorithm

```
Data: (W_i, Z_i) item values, constant δ , constant ϵ.
begin
    Let (W'_i, Z_i) be the discetized values.
    foreach i ∈ W' do
        Set Pr[X'_i|_C = t] = E[X'_i|_C]/t
    end
    for i = 1 to n do
        foreach {L, P} do
            if W'_i is a small propability item a_i then
                if DP({L, P = (..., P_i − W'_i, P_{i+1}, ...)}, v) is non empty then
                    (DP({L, P})) ⟵ max(DP({L, P =
                      (..., P_i − W'_i, P_{i+1}, ...)}) + Z_i, DP({L, P}))
                end
            end
            if DP({L − W'_i, P}) is non empty then
                (DP({L, P}, u)) ⟵ max(DP({L − W'_i, P}) + Z_i, DP({L, P}))
            end
        end
    end
    output ⟵ CalculateKnap(DP)
end
```

Theorem 3.2.2. *Given a constant $\epsilon \geq 0$ and a set S of Random Variables there exist a EPTAS for finding a set S' such as:*

$$\begin{aligned} &\textit{maximize} && \sum_{i \in S'} p_i \\ &s.t. && P[\sum_{i \in S'} X_i > 1 + \epsilon] \leq \delta(1+\epsilon) \end{aligned}$$

Απόδειξη. Same idea as the previous theorem. It is clear that the loop in (6) will run n times and the line (3) will also run n times, and update $f(\epsilon, \delta)P(\epsilon) * n^2(7)$ values where $P(\epsilon)$ is the poisson and smallest variable space thus we have $n^3 g(\epsilon\delta)$ running time which lead us to a EPTAS □

3.3 An Easy and Optimal Algorithm for Small Probability Events

In the previous section we saw that if the probability of a non zero event is small we can think our random variables as a Poisson process.This leads to a very nice algorithm that can provide a constant error and it's complexity time is as small as it can be. In fact poisson random variables are very special because they have some nice properties . One is the additive which states that the sum of two poisson is a poisson random variable with the sum of their parameters and the other one is stochastic dominance which is a way to say that a path is better than another one.

Definition 9. *We say that a distribution satisfies stochastic dominance if for parameters λ_1 λ_2 we have :*

$$\Pr[D(\lambda_1) \leq t] \leq \Pr[D(\lambda_2) \leq t] \Leftrightarrow \lambda_1 \leq \lambda_2$$

Also for poisson random variables we have the additive property which is:

Lemma 18. *Let X_1 ,X_2 be poisson random variables with parameters λ_1 λ_2 then we have that*

$$X = X_1 + X_2 \Rightarrow X = Poi(\lambda_1 + \lambda_2)$$

Απόδειξη. We know that the Moment generating function of a poisson random variable is $M_{X_1}(t) = exp(\lambda_1(e^t - 1))$ and also that $M_X(t) = M_{X_1}(t)M_{X_2}(t) = exp(()\lambda_1 + \lambda_2)(e^t - 1))$ and the prove completes with the inversion transform. □

Lemma 19. *Poisson Variables satisfy stochastic dominance.*

Απόδειξη. content... □

Using these 2 properties it is easy to prove that the best set is always the one with the lowest mean . This is proved in case of SSP as:

$$\Pr[\sum_{i \in S} W_i \leq t] = \Pr[\sum_{i \in S} Poi(\lambda_i) \leq t] = \Pr[Poi(\sum_{i \in S} \lambda_i) \leq t] \geq \Pr[Poi(\lambda') \leq t] \tag{3.16}$$

Where $\lambda' \geq \sum_{i \in S} \lambda_i$. This leads as that the Dijkstra algorithm using as weights the mean of each edge give us the optimal solution. Also from the previous section 1 we showed that if the probability of a non zero event is less than ϵ then we can assume that our variable is Poisson with error rate of ϵ^2.

Theorem 3.3.1. *Let $A = max_{X \in E} E[X]$ then there exist an A-approximate algorithm to find the optimal path which complexity is dominated by Dijkstra algorithm.*

Απόδειξη. Using 5 , we have that the d_{tv} error of using Poisson approximation is A. Using the fact that

$$d_{tv}(A, B) \leq \delta \Rightarrow \Pr[A \in X] \leq \Pr[B \in X] + \delta$$

This leads to

$$\Pr[\sum_{i \in S} X_i > 1 + \epsilon] < \Pr[\sum_{i \in S'} X_i > 1 + \epsilon] + A$$

So we solve the deterministic problem with weights the mean values and the Path we get is the best within an error rate of A. The algorithm only runs Dijkstra algorithm that means that the complexity time is dominated by Dijkstra. □

Κεφάλαιο 4

Learning Discrete Settings in Unknown Environments

In the previous section , we showed how we will approximate the solution of the Stochastic Shortest Path if we know how the environment is modeled.In reality it is difficult to know how the environment is modeled although we can always approximate it.There are many ways to see this kind of problems. In our work we are going to see them as oracles where we sample their value in a particular model. As for an example in stochastic shortest path we can see the oracle as the time that it took a car to go through a particular edge or road and take the value from the GPS. This can help as to define a new type of algorithms called PAC. We are going to define $\delta-$correct algorithms and algorithm that it is correct with confidence $1-\delta$ this means that with probability less than δ the algorithm has output a wrong result. Also there exists the $(\delta, \epsilon)-$correct algorithms which are the algorithms that approximates the solution with in an error rate of ϵ and the probability to be wrong is at most δ. In fact in real world data every value is approximated between an error rate as we can not have infinity number of bits that needed to calculate the precision of measures in fact with $\log(1/\epsilon)$ we can have at most ϵ error. Either the way we are gonna assume that our samples are the real ones without the approximation error that may or may not have. Most of this work is based on some results of [Gupta et all][28] and we are gonna generalize some results for the stochastic paths for more general costs functions using Nikolova's [29] results.
We are going to start with some basic definitions.We start by define $\delta-$correct algorithm for shortest path.

Definition 10. *We say Algorithm A is a δ-correct algorithm for stochastic shortest path on a graph $G(V, E)$ if given a function $f : R \to R$ we want to find a path P:(s,t) such for any other path P′:(s,t):*

$$\Pr[E[f(\sum_{i\in P} E_i)] \leq E[f(\sum_{i\in P'} E_i)]] \geq 1-\delta$$

As for example f may be just the sum of edges or it may define the value at risk or even more complex functions. At first we are going to start with the function $f(x) = x$.

4.1 Lower Bound for the Expected Value

Given an instance of the shortest path problem $G(V, E)$ and let P be the set of all paths from s,t. We will assume that all paths have are following Gaussian distribution with mean m_i and variance 1.

Now we are going to prove that give the optimal solution O the Lower bound (Low (G(V,E)))is the solution of the following program:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} t_i \\ \text{s.t.} \quad & \sum_{i \in A \triangle O} 1/t_i \leq (E[A] - E[O])^2 \forall A \in P \\ & t_i > 0 \end{aligned} \tag{4.1}$$

Theorem 2. *Let C be an instance of the shortest path problem and let P be all the paths from (s,t) . For any $\delta(0, 0.01)$ and any δ-correct algorithm A for the problem , A takes $\Omega(Low(C) \log 1/\delta)$ samples on Expectation on C.*

Απόδειξη. Fix δ , an instance C and an algorithm A. Let n_i be the expected number of samples drawn for the i-th Path. Let $a = d(\delta, 1 - \delta)/2$ and $t_i = n_i/a$. We have to show that it is a feasible solution for 4.1 .

$$\sum_{i=1}^{n} n_i = a \sum_{i=1}^{n} t_i \geq aLow(C) \in \Omega(Low(c) \log(1/\delta))$$

Now we fix a path A and let $\Delta_i = c/n_i$ where

$$c = \frac{2(E[O] - E[A])}{\sum\limits_{i \in A \triangle O} 1/n_i}$$

Now we change our path such that each mean of edge in $O \setminus A$ is decreased by Δ_i while the mean of edges in $A \setminus O$ is increased by Δ_i . So we have that

$$E[O'] - E[A'] = (E[O] - \sum_{i \in O \setminus A})\Delta_i) - (E[A] + \sum_{i \in A \setminus O})\Delta_i)) = E[O] - [A] - c \sum_{i \in A \triangle O} 1/n_i = -E[O] + E[A] < 0$$

This means that the O is no longer optimal in C'. Let E be the event that the algorithm returns O as the optimal solution.Now using the 7 and that A is a δ-correct algorithm we have that $P_{A,C}[E] > 1 - \delta$ and $P_{A,C'}[E] < \delta$:

$$\sum_{i \in A \triangle O} n_i KL(e_i, e_i') = \sum_{i \in A \triangle O} n_i/2\Delta_i^2 \geq d(P_{A,C}[E], P_{A,C'}[E]) \geq d(1 - \delta, \delta)$$

$$\sum_{i \in A \triangle O} n_i c^2/n_i^2 = \frac{4(E[O] - E[A])^2}{(\sum\limits_{i \in A \triangle O} 1/n_i)^2} \sum_{i \in A \triangle O} 1/n_i = \frac{4(E[O] - E[A])^2}{\sum\limits_{i \in A \triangle O} 1/n_i} \geq 2d(\delta, 1 - \delta)$$

which follows

$$\sum_{i \in A \triangle O} 1/t_i \leq (E[A] - E[O])^2$$

□

4.1.1 Finding the optimal path

In order to find to find a Path that minimizes an objective function , we have to use some sort of binary search . But when our weights are random variables it is difficult to say that this path is better than the other. For this reason we can create confidence intervals for each path and cross out the ones that with high probability are not bellow a threshold.A high level idea is that in each step we are going to decrease the range of our confidence interval while we are going to increase our probability of be correct. In fact in every step we have the intervals that are approximately correct within an error rate.This means that if we stop the algorithm in a particular moment we are going to have an approximately correct solution.One problem that we have is that the paths of a graph are expomential at a rate of n^n while the edges are at most n^2 . There comes the pareto curves which is a framework developed by Yianakakis and Papadimitriou [39] where we can create a FPTAS to find a path for a combination optimization problem. Combining all of them we have our algorithm . We are now briefly describe the main function of their method. **Remark:**In the following section we are not going to present the verify method as it is technical and it does not needed to prove our claims. The verify function adds a $\log \delta^{-1}$ in our algorithms. The proof of this method is in [28] and it is the same proof in all the situations here.

Algorithm 5: SimulEst

Data: U, accurancy parameter ϵ and confidence level δ

Result: A vector m indicating the number of samples of each edge

Let $m = (m_1, \ldots, m_E)$ be the optimal solution of this program:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{E} m_i \\ \text{s.t.} \quad & \sum_{i \in A \triangle B} 1/m_i \le \frac{\epsilon^2}{2\ln(2/\delta)} \forall A, B \in U \\ & m_i > 0 \end{aligned}$$

return m

This function provides a way to minimize the samples we needed to be able to distinguish between sub-optimal ones and not.

In high level this algorithm does a binary search until only one set remains. It is clear that if we stop in the step r we get accuracy 2^{-r}.

We are going to explain the basic ideas which show why this works. We omit the verify procedure and the correctness proof . First of all we are going to explain the choice of $\frac{\epsilon^2}{2\ln(2/\delta)}$ in Algorithm 5.

Lemma 20. *Given a set of Gaussian random variables $a_1, a_2, \ldots a_k$ with unit variance and means $m_1, m_2, \ldots, m_k$ suppose we take t_i samples from the i-th variable , and let X_i be its empirical*

Algorithm 6: NaiveGapElim

Data: Instance (C, F) and confidence level δ
Result: Best Path

$F_1 \leftarrow F$, $\delta_0 \leftarrow 0.01$, $\lambda \leftarrow 10$
for $r = 1$ *to* ∞ **do**
  if $|F_r| = 1$ **then**
    Verification
    return F_r
  end
  $\epsilon_r \leftarrow 2^{-r}$, $\delta_r \leftarrow \delta_0/(10r^2|F|^2)$
  $m^r \leftarrow SimulEst(F_r, \epsilon_r/\lambda, \delta_r)$ $\hat{m^r} \leftarrow samples(m^r)$
  $opt_r \leftarrow min_{A \in F_r} \hat{m}^r(A)$
  $F_{r+1} \leftarrow \{A \in F_r : \hat{m}^r(A) \leq opt_r + \epsilon_r/2 + 2\epsilon_r/\lambda\}$
end

mean. Then we have

$$\Pr[|\sum_{i=1}^{k} X_i - \sum_{i=1}^{k} m_i| \geq \epsilon] \leq 2 \exp - \frac{\epsilon^2}{2 \sum_{i=1}^{k} 1/t_i}$$

Απόδειξη. The variable $X = \sum_{i=1}^{k} X_i$ follows a Gaussian distribution with mean $m = \sum_{i=1}^{k} m_i$ and variance $\sum_{i=1}^{k} 1/t_i$ thus we can use the tail bound of Gaussian distribution and the result follows. $\square$

Now it is clear that we need to get enough samples to lower our variance and then we can have enough error. It is clear that SimulEst outputs the number of samples that is needed to have +-ϵ error in the mean value. One problem that we have is that our failure probability is independent in each Path. This means that $\sum_j \Pr[|\sum_{i=I_j} X_i - \sum_{i=1}^{k} m_i| \geq \epsilon] \leq |F|\delta$ thus we need to tweak the delta by setting $\delta = \delta/|F|$ to have our result. Also the $|F|$ can be exponentially large, but as the δ is in a logarithm it increase our samples polynomially (around $nlogn$).

4.1.2 Learning for general functions

We have seen a general method of how to find the path with the lowest mean using sub-optimal samples. Now we are going to see how to find paths with more objectives such as: minimization of tail objective , portfolio maximization and and minimization of polynomials of degree 3. First of all the problem that we have here is that variance effects the cost function , so we need to develop a method that also takes as an input the variance. We are going to show a sub optimal algorithm in a case where we have Normal Random variable. We are going to show that in fact we have to find to separate the sub-level sets until there exists only one Path which can be done with the use of Pareto curves [39].

First of all we see that when our random variables are following Normal distribution , we have that by additive property that

$$\Pr[\sum_{i=1}^{n} X_i \leq t] = \Phi(\frac{t - E[X]}{\sqrt{var[X]}})$$

.

So we have the following program:

$$\begin{aligned} &\text{maximize } \tfrac{t-E[X]}{\sqrt{var[X]}} \\ &\text{s.t.} \qquad\qquad X \in P \end{aligned} \tag{4.2}$$

This program is quasi-convex maximization which is NP-Hard to solve. Although we will try to find some methods to approximate .It is known by [29] that we can efficiently approximate the solution. We are going to use some of her methods to solve the problem when the mean and the variance are unknown . In fact we are going to show that this problem is equivalent to portfolio minimization 4.3.

$$\begin{aligned} &\min E[X] + k\sqrt{Var[X]} \\ &\text{s.t.} \qquad\qquad X \in P \end{aligned} \tag{4.3}$$

To show the equivalence we see that if we have an oracle for 4.2 then we can answer the question:

$$\frac{t - E[X]}{\sqrt{var[X]}} > k \Leftrightarrow t - (E[X] + k\sqrt{var[X]}) > 0$$

Now we are going to solve the problem in steps. First we are going to solve it when the variance of each edge is known and then when it is not.

4.1.2.1 Unknown mean and known variance

In this section we are going to show how we can efficiently sample our distributions in a way that it will help us to find the optimal distribution.

The following lemma is essential to our proof:

Lemma 21. *Let Y be a random variable with variance σ^2 , there exist an estimator $\hat{Y}$ which with $O(\log(1/\delta)/\epsilon^2)$ samples satisfies:*

$$|E[Y] - \hat{Y}| \leq \epsilon\sigma$$

Απόδειξη. Let $\hat{Y} = \sum_{i=1}^{n} Y_i/n$ where Y_i is the i-th sample of Y. It is clear that $E[\hat{Y}] = E[Y]$ and also that $\hat{Y} \sim N(E[Y], \sigma^2/n)$. Using tail bounds for Normal Variables we have that:

$$\Pr[|\hat{Y} - E[Y]| > \epsilon] \leq 2exp(\frac{-\epsilon^2}{2\sigma^2/n})$$

By setting $n = \frac{\sigma^2}{\epsilon^2}\log(1/\delta)$ we complete the proof. □

Now we are going to define the sublevel sets which we are going to use to do our binary search in our set.

Definition 11. *We define L_λ as the sublevel set of the function f which is $L_\lambda = \{x \in P | f(x) \leq \lambda\}$*

Definition 12. *We define the set of breakpoints V as the values of λ that separates two solutions. $V = \{\lambda_i \in R | |L_{\lambda_{i+1}}| - |L_{\lambda_i}| < 0\}$*

This notation helps as to provide a way to compare two different solutions.In fact if we have a way to find all the set V then easily we will have a way to find the optimal solution. We argue that every algorithm that finds the optimal solution must have enough samples to separate two different values of breakpoints. Using the previous technique one can prove that the minimum samples that are required to find the optimal solution are the solution of the Program P_3 4.3.

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} t_i \\ \text{s.t.} \quad & \sum_{i \in A \triangle O} \sigma_i^2/t_i \leq (f(O) - f(A))^2 \forall A \in P \\ & t_i > 0 \end{aligned} \tag{4.4}$$

Now we have to prove that this program is the Lower bound that we need. We are going to use the same method proposed by Gupta. To prove following:

Theorem 3. *Let C be an instance of the shortest path problem and let P be all the paths from (s,t) . For any $\delta(0, 0.01)$ and any δ-correct algorithm A for the problem , A takes $\Omega(P_3(C))\log 1/\delta)$ samples on Expectation on C.*

Απόδειξη. Fix δ , an instance C and an algorithm A. Let n_i be the expected number of samples drawn for the i-th Path. Let $a = d(\delta, 1-\delta)/2$ and $t_i = n_i/a$. We have to show that it is a feasible solution for 4.3 .

$$\sum_{i=1}^{n} n_i = a \sum_{i=1}^{n} t_i \geq aLow(C) \in \Omega(Low(c)\log(1/\delta))$$

Now we fix a path A and let $\Delta_i = c\sigma_i^2/n_i$ where

$$c = \frac{2(f(O) - f(A))}{\sum_{i \in A \triangle O} \sigma_i^2/n_i}$$

Now we change our path such that each mean of edge in $O \setminus A$ is decreased by Δ_i while the mean of edges in $A \setminus O$ is increased by Δ_i . So we have that

$$f(O')-f(A') = (f(O)-\sum_{i\in O\setminus A})\Delta_i)-(f[A]+\sum_{i\in A\setminus O})\Delta_i)) = f(O)-f(A)-c\sum_{i\in A\triangle O}\sigma_i^2/n_i = -f(O)+f(A) < 0$$

This means that the O is no longer optimal in C'. Let E be the event that the algorithm returns O as the optimal solution.Now using the 7 and that A is a δ-correct algorithm we have that $P_{A,C}[E] > 1-\delta$ and $P_{A,C'}[E] < \delta$:

$$\sum_{i\in A\triangle O} n_i KL(e_i, e_i') = \sum_{i\in A\triangle O} n_i \Delta_i^2/2\sigma_i^2 \geq d(P_{A,C}[E], P_{A,C'}[E]) \geq d(1-\delta,\delta)$$

$$\sum_{i\in A\triangle O} n_i * \sigma_i^2 \frac{c^2}{n_i^2/\sigma_i^4} = \frac{4(f(O)-f(A))^2}{(\sum_{i\in A\triangle O}\sigma_i^2/n_i)^2}\sum_{i\in A\triangle O}\sigma_i^2/n_i = \frac{4(f(O)-f(A))^2}{\sum_{i\in A\triangle O}\sigma_i^2/n_i} \geq 2d(\delta, 1-\delta)$$

which follows

$$\sum_{i\in A\triangle O}\sigma_i^2/t_i \leq (f(A)-f(O))^2$$

□

Now we present our algorithms.

Algorithm 7: SimulEstP

Data: U, accurancy parameter ϵ and confidence level δ

Result: A vector m indicating the number of samples of each edge

Let $m = (m_1, \ldots, m_E)$ be the optimal solution of this program:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{E} m_i \\ \text{s.t.} \quad & \sum_{i\in A\triangle B}\sigma_i^2/m_i \leq \tfrac{\epsilon^2}{2\ln(2/\delta)} \forall A, B \in U \\ & m_i > 0 \end{aligned}$$

return m

Algorithm 8: NaiveGapElimPortfolio

Data: Instance (C, F)

Result: Best Path

$F_1 \leftarrow F$, $\delta_0 \leftarrow 0.01$, $\lambda \leftarrow 10$

for $r = 1$ *to* ∞ **do**

  if $|F_r| = 1$ **then**

    Verification

    return F_r

  end

  $\epsilon_r \leftarrow 2^{-r}$, $\delta_r \leftarrow \delta_0/(10r^2|F|^2)$

  $m^r \leftarrow SimulEstP(F_r, \epsilon_r/\lambda, \delta_r)$

  $\hat{m^r} \leftarrow samples(m^r)$

  $opt_r \leftarrow min_{A \in F_r} \hat{m}^r(A) + k\sigma_A$

  $F_{r+1} \leftarrow \{A \in F_r : \hat{m}^r(A) + k\sigma_A \leq opt_r + \epsilon_r/2 + 2\epsilon_r/\lambda\}$

end

It is clear that with probability $1-\delta_r$ in each round r , we know the items with approximation ratio $+\epsilon_r$ due to 21. The rest of the proof follows like in [28] .

Theorem 4.1.1. *For every instance C, NaiveGapPortfolio takes*

$$O(P_3(C)(\log \delta^{-1} + \log \Delta^{-1}(\log \log \Delta^{-1} + \log P))$$

where P is the number of paths and $\Delta = f(O) - max_{A \in (P-O)} f(A)$.

Απόδειξη. In each step r let $a = 16\lambda^2 \log(2/\delta_r)$ and $m_i = at_i$ and $fix A, B \in P$, SimulEstP takes

$$\begin{aligned}\sum_{i \in A \triangle B} \sigma_i^2/m_i \leq a^{-1}(\sum_{i \in A \triangle O} \sigma_i^2/t_i + \sum_{i \in B \triangle O} \sigma_i^2/t_i) \\ \leq a^{-1}((f(O) - f(A))^2 + (f(O) - f(B))^2) \\ \leq 2a^{-1}\epsilon_{r-1}^2 \\ \leq \frac{\epsilon_r^2}{2\ln(2/\delta_r)}\end{aligned}$$

Where in the second line we used the fact that due the step of our algorithm, every path that is still inside the set F has $f(O) - f(X) \leq \epsilon_{r-1}$

So we have that : $\sum_{i \in P} m_i = a \sum_{i \in P} t_i = O(P_3(C) \log 1/\delta_r) = O(P_3(C)(\log r + \log F))$. Now we sum through r and we have that :

$$O(\sum_{r=1}^{r^*} P_3(C)(\log r + \log F)) \leq O(r^* P_3(C)(\log r^* + \log F))$$

Then we now that our algorithm stops when we are at a distance Δ^{-1} and we need $\log \Delta^{-1}$ to get there. This means that:

$$O(r^* P_3(C)(\log r^* + \log F)) \leq O(\log \Delta^{-1} P_3(C)(\log \log \Delta^{-1} + \log F))$$

□

4.2 Unknown Variance

The ability to know the variance of each edge is very advantageous as any estimator that learns the variance needs a lot of samples. In this section we will show how to optimize functions of random variables that are polynomials of third degree.

Lemma 22. *Let f be a polynomial of third degree and X a random variable then :*

$$E[f(X)] = f[E[X]] + 1/2 f''(E[X]) Var[X]$$

Απόδειξη. By expanding f using Taylor in the point $E[X]$ series we have:

$$f(X) = f(E[X]) + f'(E[X])(X - E[X]) + 1/2 f''(E[X])(X - E[X])^2$$

By taking the expected value the proof concludes. □

This lemma show that we do not need an estimator for f to find its mean value but we only need an estimator for X.

Lemma 23 (from [42] Lemma 6)**.** *There exist an algorithm A that takes* $O(\log(1/\delta)/\epsilon^2)$ *samples and gives with probability* $1 - \delta$ *outputs estimates* $\hat{m}, \hat{\sigma^2}$ *such that:*

$$|m - \hat{m}| \leq \epsilon\sigma \quad |\sigma^2 - \hat{\sigma}^2| \leq 2\epsilon\sigma^2$$

This is slightly modified as excess kurtosis for normal distributions is 0. So the question which remains is how many samples do we want for finding the best path.
As our functions are polynomials of third degree their derivative is has at most 2 roots. We also have that

$$f(E[X]) - f([E[Y]]) = f'(\xi)(E[X] - E[Y])$$

This means that we can assume that for small differences the sign for $f(E[X]) - f([E[Y]])$ depends on $E[X] - E[Y]$ this leads us that the previous bounds for mean value still holds.
Now we are going to prove bounds for variance. We argue that the program P_4:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} t_i \\ \text{s.t.} \quad & \sum_{i \in A \triangle O} \sigma_i^4 / t_i \leq (f''(E[O]))(Var(O) - Var(A))^2 \quad \forall A \in P \\ & \sum_{i \in A \triangle O} \sigma_i^2 / t_i \leq (f''(O) - f''(A))^2 \quad \forall A \in P \\ & \sum_{i \in A \triangle O} \sigma_i^2 / t_i \leq (f(O) - f(A))^2 \quad \forall A \in P \\ & t_i > 0 \end{aligned} \tag{4.5}$$

Lemma 24. *Let C be an instance of the shortest path problem and let P be all the paths from (s,t) . For any* $\delta(0, 0.01)$ *and any* δ*-correct algorithm A for the problem , A takes* $\Omega(P_4(C)) \log 1/\delta)$ *samples on Expectation on C.*

Απόδειξη. Fix δ , an instance C and an algorithm A. Let n_i be the expected number of samples drawn for the i-th Path. Let $a = d(\delta, 1-\delta)/2$ and $t_i = n_i/a$. We have to show that it is a feasible solution for 4.5 .

$$\sum_{i=1}^{n} n_i = a \sum_{i=1}^{n} t_i \geq a P_4(C) \in \Omega(P_4(c)\log(1/\delta))$$

Now we fix a path A and let $\Delta_i = c\sigma_i^4/n_i$ where

$$c = \frac{2(f''(E[X])(Var(O) - Var(A))}{\sum\limits_{i \in A \cup O} \sigma_i^2/n_i}$$

Now we change our path such that each variance of edge in $O \setminus A$ is decreased by Δ_i while the mean of edges in $A \setminus O$ is increased by Δ_i . So we have that

$$f(O') - f(A') = (f(O) - \sum_{i \in O \setminus A} \Delta_i) - (f[A] + \sum_{i \in A \setminus O} \Delta_i)) = f(O) - f(A) - c \sum_{i \in A \triangle O} \sigma_i^2/n_i = -f(O) + f(A) < 0$$

This means that the O is no longer optimal in C'. Let E be the event that the algorithm returns O as the optimal solution.Now using the 7 and that A is a δ-correct algorithm we have that $P_{A,C}[E] > 1-\delta$ and $P_{A,C'}[E] < \delta$,using 4 we have:

$$\sum_{i \in A \triangle O} n_i KL(e_i, e_i') = \sum_{i \in A \triangle O} n_i \Delta_i^2/\sigma_i^4 \geq d(P_{A,C}[E], P_{A,C'}[E]) \geq d(1-\delta, \delta)$$

$$\begin{aligned} \sum_{i \in A \triangle O} n_i * \sigma_i^4 \frac{c^2}{n_i^2/\sigma_i^8} &= \frac{4(f''(E[X])(Var(O) - Var(A)))^2}{(\sum\limits_{i \in A \triangle O} \sigma_i^8/n_i)^2} \sum_{i \in A \triangle O} \sigma_i^4/n_i \\ &= \frac{4(f''(E[X])(Var(O) - Var(A)))^2}{\sum\limits_{i \in A \triangle O} \sigma_i^2/n_i} \geq 2d(\delta, 1-\delta) \end{aligned}$$

which follows

$$\sum_{i \in A \triangle O} \sigma_i^2/t_i \leq (f''(E[O])(Var(O) - Var(A)))^2$$

□

This means that the variance dominates the sample complexity. We are going to propose the two algorithms that needed.

Algorithm 9: SimulEstG

Data: U, accurancy parameter ϵ and confidence level δ

Result: A vector m indicating the number of samples of each edge

Let $m = (m_1, \ldots, m_E)$ be the optimal solution of this program:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{E} m_i \\ \text{s.t.} \quad & \sum_{i \in A \triangle B} \hat{\sigma_i}^2 / m_i \leq \tfrac{\epsilon^2}{2\ln(2/\delta)} \forall A, B \in U \\ & \sum_{i \in A \triangle B} \hat{\sigma_i}^4 / m_i \leq \tfrac{\epsilon^2}{2\ln(2/\delta)} \\ & m_i > 0 \end{aligned}$$

return m

Algorithm 10: NaiveGapElimGeneral

Data: Instance (C, F)

Result: Best Path

$F_1 \leftarrow F$, $\delta_0 \leftarrow 0.01$, $\lambda \leftarrow 10$

Let $\hat{\sigma}^2$ be a 2-approximation vector of the vector σ of variances

for $r = 1$ *to* ∞ **do**

  if $|F_r| = 1$ **then**

    Verification

    return F_r

  end

  $\epsilon_r \leftarrow 2^{-r}$, $\delta_r \leftarrow \delta_0/(10r^2|F|^2)$

  $m^r \leftarrow SimulEstG(F_r, \epsilon_r/\lambda, \delta_r/2, \hat{\sigma}^2)$

  $\hat{m^r} \leftarrow samples(m^r)$

  $opt_r \leftarrow min_{A \in F_r} f(\hat{m}^r(A)) + f''(\hat{m}^r(A))(\hat{\sigma}^r(A))$

  $F_{r+1} \leftarrow \{A \in F_r : f(\hat{m}^r(A)) + f''(\hat{m}^r(A))(\hat{\sigma}^r(A)) \leq opt_r + \epsilon_r/2 + 2\epsilon_r/\lambda\}$

end

First of all it is clear by 23 that SimulEstG outputs an $+-\epsilon$ approximation of the mean value and the variance. This is done because we get a 2-approximation of the variance. That only increase our samples by a factor 8.Furthermore , this means that we can always have a good approximation of our mean value.

Theorem 4.2.1. *For every instance C, NaiveGapPortfolio takes*

$$O(P_4(C)(\log \Delta^{-1}(\log \log \Delta^{-1} + \log P))$$

where P is the number of paths and $\Delta = minV$ which means that Δ is the minimum break point.

Απόδειξη. Using the same ideas as before . In each step r let $a = 16\lambda^2 \log(2/\delta_r)$ and $m_i = at_i$ and $fixA, B \in P$, SimulEstPE takes

$$\begin{aligned} \sum_{i \in A \triangle B} \sigma_i^4/m_i \leq a^{-1}(\sum_{i \in A \triangle O} \sigma_i^4/t_i + \sum_{i \in B \triangle O} \sigma_i^2/t_i) \\ \leq a^{-1}((f''(E[O]))(Var(O) - Var(A))^2 + (f''(E[O]))(Var(O) - Var(B))^2) \\ \leq 2a^{-1}\epsilon_{r-1}^2 \\ \leq \frac{\epsilon_r^2}{2\ln(2/\delta_r)} \end{aligned}$$

Also we have that :

$$\begin{aligned} \sum_{i \in A \triangle B} \sigma_i^2/m_i \leq a^{-1}(\sum_{i \in A \triangle O} \sigma_i^2/t_i + \sum_{i \in B \triangle O} \sigma_i^2/t_i) \\ \leq a^{-1}((f(O) - f(A))^2 + (f(O) - f(B))^2) \\ \leq 2a^{-1}\epsilon_{r-1}^2 \\ \leq \frac{\epsilon_r^2}{2\ln(2/\delta_r)} \end{aligned}$$

And the same equation for $f''(O)$.

So we have that : $\sum_{i \in P} m_i = a \sum_{i \in P} t_i = O(P_3(C) \log 1/\delta_r) = O(P_4(C)(\log r + \log F))$. Now we sum through r and we have that :

$$O(\sum_{r=1}^{r^*} P_4(C)(\log r + \log F)) \leq O(r^* P_4(C)(\log r^* + \log F))$$

Then we now that our algorithm stops when we are at a distance Δ^{-1} of our error as there we will only have on solution ahead thus we need $\log \Delta^{-1}$ to get there. This means that:

$$O(r^* P_4(C)(\log r^* + \log F)) \leq O(\log \Delta^{-1} P_4(C)(\log \log \Delta^{-1} + \log F))$$

$\square$

Βιβλιογραφία

[1] Herbert Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5):527–535, 09 1952.

[2] Evdokia Nikolova, Jonathan A. Kelner, Matthew Brand, and Michael Mitzenmacher. Stochastic shortest paths via quasi-convex maximization. In Yossi Azar and Thomas Erlebach, editors, *Algorithms – ESA 2006*, pages 552–563, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[3] 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*, SODA '11, pages 1647–1665, Philadelphia, PA, USA, 2011. Society for Industrial and Applied Mathematics.

[4] Asaf Levin and Aleksander Vainer. The benefit of adaptivity in the stochastic knapsack problem with dependence on the state of nature. *Discrete Optimization*, 10(2):147 – 154, 2013.

[5] Anupam Gupta, Ravishankar Krishnaswamy, Marco Molinaro, and R. Ravi. Approximation algorithms for correlated knapsacks and non-martingale bandits. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 827–836, Washington, DC, USA, 2011. IEEE Computer Society.

[6] Jian Li and Amol Deshpande. Utility maximization under uncertainty. *CoRR*, abs/1012.3189, 2010.

[7] Andrew C. Berry. The accuracy of the gaussian approximation to the sum of independent variates. *Transactions of the American Mathematical Society*, 49(1):122–136, 1941.

[8] C. G. Esseen. A moment inequality with an application to the central limit theorem. *Scandinavian Actuarial Journal*, 1956(2):160–170, 1956.

[9] L.H.Y. Chen, L. Goldstein, and Q.M. Shao. *Normal Approximation by Stein's Method*. Springer Verlag, 2010.

[10] Gregory Valiant and Paul Valiant. Estimating the unseen: An n/log(n)-sample estimator for entropy and support size, shown optimal via new clts, 2011.

[11] Constantinos Daskalakis, Gautam Kamath, and Christos Tzamos. On the structure, covering, and learning of poisson multinomial distributions. *CoRR*, abs/1504.08363, 2015.

[12] J. L. Hodges and Lucien Le Cam. The poisson approximation to the poisson binomial distribution. *Ann. Math. Statist.*, 31(3):737–740, 09 1960.

[13] L. Heinrich. A. d. barbour, l. holst and s. janson: Poisson approximation. oxford university press 1992, 277 pp., isbn 0-19-852235-5. *Biometrical Journal*, 35(4):511–511.

[14] Constantinos Daskalakis and Christos H. Papadimitriou. Sparse covers for sums of indicators. *CoRR*, abs/1306.1265, 2013.

[15] D. Stoyan. Berry, d. a. and b. fristedt: Bandit problems. sequential allocation of experiments. monographs on statistics and applied probability. chapman and hall, london/new york 1985, 275 s. *Biometrical Journal*, 29(1):20–20.

[16] T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4 – 22, 1985.

[17] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 322–331, Oct 1995.

[18] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, January 2003.

[19] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Pac bounds for multi-armed bandit and markov decision processes. In Jyrki Kivinen and Robert H. Sloan, editors, *Computational Learning Theory*, pages 255–270, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[20] Martin Anthony and P. L. Bartlett. *Neural network learning:theoretical foundations.* Cambridge University Press, Cambridge, U.K., 1999.

[21] Shie Mannor and John N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. In *Journal of Machine Learning Research*, 2003.

[22] Karthekeyan Chandrasekaran and Richard M. Karp. Finding the most biased coin with fewest flips. *CoRR*, abs/1202.3639, 2012.

[23] Matthew L Malloy, Gongguo Tang, and Robert D Nowak. Quickest search for a rare distribution. In *Information Sciences and Systems (CISS), 2012 46th Annual Conference on*, pages 1–6. IEEE, 2012.

[24] Wei Cao, Jian Li, Yufei Tao, and Zhize Li. On top-k selection in multi-armed bandits and hidden bipartite graphs. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1036–1044. Curran Associates, Inc., 2015.

[25] Lijie Chen and Jian Li. On the optimal sample complexity for best arm identification. *CoRR*, abs/1511.03774, 2015.

[26] Lijie Chen, Jian Li, and Mingda Qiao. Towards instance optimal bounds for best arm identification. *CoRR*, abs/1608.06031, 2016.

[27] Lijie Chen, Anupam Gupta, and Jian Li. Pure exploration of multi-armed bandit under matroid constraints. *CoRR*, abs/1605.07162, 2016.

[28] Lijie Chen, Anupam Gupta, Jian Li, Mingda Qiao, and Ruosong Wang. Nearly optimal sampling algorithms for combinatorial pure exploration. *CoRR*, abs/1706.01081, 2017.

[29] Evdokia Nikolova. *Strategic algorithms.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2009.

[30] A. Goel and P. Indyk. Stochastic load balancing and related problems. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pages 579–586, Oct 1999.

[31] J. Li and W. Yuan. Stochastic Combinatorial Optimization via Poisson Approximation. *ArXiv e-prints*, November 2012.

[32] Shie Mannor, John Tsitsiklis, Kristin Bennett, and Nicol O Cesa-bianchi. The sample complexity of exploration in the multi-armed bandit problem. 07 2004.

[33] Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 3212–3220. Curran Associates, Inc., 2012.

[34] Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1238–1246, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[35] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best arm identification in multi-armed bandit models. *CoRR*, abs/1407.4443, 2014.

[36] Fan Chung and Linyuan Lu. *Complex Graphs and Networks (Cbms Regional Conference Series in Mathematics).* American Mathematical Society, Boston, MA, USA, 2006.

[37] Jon Kleinberg, Yuval Rabani, and Éva Tardos. Allocating bandwidth for bursty connections. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC '97, pages 664–673, New York, NY, USA, 1997. ACM.

[38] Patricia June Carstensen. The complexity of some problems in parametric linear and combinatorial programming. 1983.

[39] C. H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 86–92, Nov 2000.

[40] Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.

[41] LR Ford. D. r. fulkerson, flows in networks. *The RAND Corporation*, 1962.

[42] Constantinos Daskalakis, Ilias Diakonikolas, and Rocco A. Servedio. Learning poisson binomial distributions. *CoRR*, abs/1107.2702, 2011.