COMBINATORIAL IDENTIFICATION IN MULTI-ARMED BANDITS

NISHANT D. GURNANI

Advisor: Sébastien Bubeck

Abstract. In this thesis we study the problem of combinatorial identification in a multi-armed bandit game. Our proposed solutions rely on extensions of work by Audibert et al., 2010 and Bubeck et al., 2013. Specifically, we propose two parameter free algorithms, Combinatorial SAR & SR, to the problem of finding a combinatorial structure with maximal mean given a finite number of evalutions. We discuss theoretical aspects of the algorithms and run experiments on the specific problem of finding a maximum weighted spanning tree. Ultimately, we show that our algorithmic contributions do not prove to be a natural framework to solve the combinatorial identification problem.

This thesis represents my own work in accordance with University regulations.

Nishant D. Gurnani

Date: May 6th 2013.
Senior Thesis for the Department of Mathematics, Princeton University.

Acknowledgements

First and foremost I'd like to thank my advisor Sébastien Bubeck, without whose patience, guidance, and insight this thesis would simply not exist. Additionally I'd like to thank Michael Damron for taking time out of his very busy schedule to represent the Mathematics Department as the second reader for this thesis.

I'd like to thank all my teachers at Brunswick School for putting me on the path that eventually led here. Thank you to my friends for the most wonderful four years of my life, without you I wouldn't have survived. Terrace F. Club, thank you for your food, thank you for your love.

To my family, thank you for believing in me all these years. Without your constant support and love I would not be here today.

Contents

1. Introduction

The *multi-armed bandit problem* is a sequential decision problem where, at each stage, an agent (or forecaster) faces of a choice between a fixed number of d stochastic arms and receives a random reward according to the distribution of the chosen arm[1]. His goal is simply to maximize the cumulative sum of rewards.

Introduced by Robbins (1952) [9], the stochastic bandit problem and its variations have been extensively studied over the years. It provides a useful model of what is known as the *exploration-exploitation tradeoff*, a fundamental problem in many areas such as statistics, economics and machine learning. The tradeoff arises from the fact that the agent requires more information about his environment in order to be able to take good actions. The agent can continue to exploit the arm he knows to have performed best so far or explore other arms in the hope of finding one with higher expected reward.

In this thesis we study a variation of the classical multi-armed bandit problem described above, where there is no explicit trade-off between exploration and exploitation. Our setting is as follows: the agent is allowed to sample the arms a fixed number, n, times after which he must output a recommendation (identify some subset of the arms) corresponding to some pre-specified criterion. This is the so-called *pure-exploration problem* that was introduced by Bubeck et al., 2009 [3], where the objective was to identify the distribution with maximal mean. In this scenario, the agent is evaluated by the difference between the average payoff of the best arm and the average payoff obtained by his recommendation.

The pure-exploration problem is a natural framework for applications where one needs to design strategies that make best possible use of limited resources in order to optimize the performance of some decision-making task. An example from [3] concerns channel allocation from mobile phone communications. During the short time before the communication starts, a cellphone has a limited number of evaluations to explore the set of channels to find the best one to operate. The cellphone's objective is to find the best channel (one with least noise) given the limited evaluations. More generally, the pure-exploration problem applies to situations with a preliminary exploration phase in which costs are not measured in terms of rewards but in terms of resources that come in limited budget (i.e. time to connect for channel allocation).

For this pure-exploration problem, Audibert et al., 2010 [1] proposed an optimal parameter-free algorithm, called SR (Successive Rejects). In particular they showed that the algorithm requires $n = \mathcal{O}(Hlog^2K)$ evaluations to be able to find the best arm. Furthermore the authors introduced a notion of *best arm identification complexity*, denoted H, which characterizes the hardness of identifying the best distribution in a specific set of K distributions. In [4], this setting was extended to study the *m-best arm identification problem*, in which the objective was to find the m arms with highest means. The authors in the latter paper introduced a new algorithm SAR (Successive Accept and Rejects), that required only $\widetilde{\mathcal{O}}\left(H^{\langle m \rangle}\right)$ evaluations to find the top m arms. Additionally they extended the complexity to apply to the m-best setting, denoted $H^{\langle m \rangle}$.

In this thesis, we extend the ideas in [1] and [4] to apply to the combinatorial indentification problem, where the objective is to find the combinatorial structure with maximal mean given a fixed number of samples. We propose two algorithms, Combinatorial SAR & SR, and define notions of gap and complexity for the combinatorial setting. We also run some experiments on the specific setting of finding a maximum weighted spanning tree. Ultimately, through discussion and numerical results we suggest that extensions of SAR and SR are not suited for the problem of combinatorial identification.

[1]The terminology of "arms" and "bandits" originates from slot machines which are often colloquially referred to as "one-armed bandits".

The organization of the thesis is as follows. In Section 2 we setup our problem, introduce relevant notation and define complexity. In Section 3 we describe our proposed algorithms Combinatorial SAR & SR and discuss two specific combinatorial settings: finding maximum weighted bipartite matchings and maximum weighted spanning trees. In Section 4 we propose some simple experiments to compare our two algorithms against uniform sampling, for the specific combinatorial setting of finding a maximum weighted spanning tree. Finally we conclude in Section 5 with a discussion of possible and desirable future directions.

2. Problem Setup

2.1. **Problem Statement.** We are interested in the following situation: An agent faces d arms and has a budget of n evaluations (or *pulls*). For each arm $i \in \{1, \dots, d\}$ there is an associated probability distribution ν_i with mean μ_i (we denote $\mu = (\mu_1, \dots, \mu_d) \in \mathbb{R}^d$). These distributions are unknown to the agent, but we assume that they are sub-gaussian. At each round $t = 1, \dots, n$, the agent chooses an arm I_t, and observes a reward drawn from ν_{I_t} independently from the past given I_t. The agent's goal is to identify the best subset of arms satisfying some given combinatorial structure. More precisely, the agent is given a set $\mathcal{C} \subset \{0,1\}^d$ where the combinatorial set $\mathcal{C}$ is a a subset of the d-dimensional hypercube $\{0,1\}^d$ such that $\forall\ c \in \mathcal{C},\ \|c\|_1 = m$. In other words, each element in the set $\mathcal{C}$ has m arms corresponding to a combinatorial structure e.g. m edges in a spanning tree. At the end of n evaluations, the agent selects $c_n \in \mathcal{C}$ based on his observations. His objective is that c_n corresponds to the set of arms with maximal rewards.[2]

Parameters: number of rounds n, number of arms d, combinatorial set $\mathcal{C}$.

Parameters unknown to agent: the reward distributions $\nu_1, \dots, \nu_d$.

For each round $t = 1, 2, \dots, n$:

(1) the agent chooses an arm $I_t \in \{1, \dots, d\}$.

(2) the environment draws a reward $X_{I_t, T_{I_t}(t)}$ from ν_{I_t} independently from the past given I_t

At the end of n rounds, the agent outputs a recommendation c_n (with m arms) based on his observations.

Figure 1: The pure exploration problem for multi-armed bandits in a combinatorial setting

For each arm i we denote by $T_i(t)$ the number of times that arm i was pulled from rounds 1 to t. Subsequently we denote the sequence of rewards for a given arm i as $X_{i,1}, \dots, X_{i,T_i(t)}$. Thus the empirical mean of arm i after s pulls is $\widehat{X_{i,s}} = \frac{1}{s}\sum_{t=1}^{s} X_{i,t}$.

2.2. **Complexity Measures.** Let $c^* = \operatorname{argmax}_{c \in \mathcal{C}} c^\top \mu$, denote the optimal structure. We evaluate the performance of the agent's strategy by the probability of misidentification,

$$e_n = \mathbb{P}\left(c_n \neq \operatorname*{argmax}_{c \in \mathcal{C}} c^\top \mu\right).$$

While finer measures of performance can be proposed (such as simple regret $r_n = [c^* - \mathbb{E}c_n]$), as argued in [1] for a first order analysis we can simply focus on the quantity e_n.

Based on the complexity measures defined previously in [1] and [4], we introduce the following gaps:

$$\Delta_i = \left| \max_{c \in \mathcal{C} \text{ s.t. } c_i = 1} c^\top \mu - \max_{c \in \mathcal{C} \text{ s.t. } c_i = 0} c^\top \mu \right|,$$

[2] To simplify our analysis, we will assume that the rewards are in $[0, 1]$ and that there is a unique optimal structure within a combinatorial setting.

and the corresponding hardness measures:

$$H_1 = \sum_{i=1}^{d} \frac{1}{\Delta_i^2} \qquad \text{and} \qquad H_2 = \max_{i \in \{1,\dots,d\}} i\Delta_i^{-2}$$

Furthermore note that the two complexity measures are equivalent up to a logarithmic factor, below is the proof from [1] included for the sake of completeness.

Proposition 2.1. *The two complexity measures H_1 and H_2 satisfy the following inequality:*

$$H_2 \leq H_1 \leq log(2d)H_2.$$

Proof. The left inequality holds as: for any $i \in \{1, \dots, d\}$, $H_1 = \sum_{k=1}^{d} \Delta_{(k)}^{-2} \geq \sum_{k=1}^{i} \Delta_{(i)}^{-2} \geq i\Delta_{(i)}^{-2}$.

To prove the right inequality, first let $\overline{\log}(d) = \frac{1}{2} + \sum_{i=2}^{d} \frac{1}{i}$ and note that $log(d+1) - \frac{1}{2} \leq \overline{\log}(d) \leq log(d) + \frac{1}{2} \leq log(2d)$. Then the inequality follows as: $\sum_{i=1}^{d} \Delta_{(i)}^{-2} = \Delta_{(2)}^{-2} + \sum_{i=2}^{d} \frac{1}{i} i\Delta_{(i)}^{-2} \leq \overline{\log}(d) \max_{i \in \{1,\dots,d\}} i\Delta_{(i)}^{-2}$.

□

We define two complexity measures largely because we've found, based on previous literature, that the quantity H_2 proves to be a very useful substitute for H_1 when proving upper bounds on e_n. Furthermore, while we do not prove any bounds for our problem using the complexity measures, we argue that these quantities are indeed characteristic of the hardness of the problem. We suggest that intuitively any strategy needs a budget n of order at least H_1 to find the optimal combinatorial structure c^*.

Consider a fixed arm i, and assume that we know the values μ_j, for any $j \neq i$. In this scenario one faces a hypothesis testing problem for arm i needing to decide whether its value μ_i is large enough to include instead of the corresponding arm in the optimal structure c^*. Let ξ_i be the threshold value for this hypothesis testing problem (note that ξ_i depends on $\mu_1, \dots, \mu_{i-1}, \mu_{i+1}, \dots, \mu_d$). To ensure no mistake in the selection of the optimal structure, we need to sample arm i at least $\frac{1}{(\mu_i - \xi_i)^2}$ times[3]. The key observation is to note that $|\mu_i - \xi_i| = \Delta_i$. The value $|\mu_i - \xi_i|$ is exactly how much must be added (or subtracted) to the value μ_i such that i becomes part of the optimal structure (is no longer part of the optimal structure) and this matches our definition of Δ_i.

3. Combinatorial best arm identification

In this section we propose two algorithms for combinatorial identification: Combinatorial SAR and Combinatorial SR. First we define our benchmark algorithm, uniform sampling, and discuss our motivation for trying to do better. Then we describe the algorithms in detail and discuss their shortcomings. Finally, we conclude with a discussion in 3.4 of combinatorial identification in two specific settings: finding the maximum weighted bipartite matching and maximum weighted spanning tree.

3.1. **Uniform Sampling.** (See Fig. 2)

The uniform sampling algorithm serves as an important theoretical benchmark to which all other algorithms are compared. The algorithm proceeds by sampling each arm $\lfloor n/d \rfloor$ times and then outputs the m arms with maximal mean corresponding to the given combinatorial structure.

The algorithm is the simplest way to sample the arms and our motivation for proposing other algorithms stems from the fact that there is potential to sample more intelligently. Given the constraint of a finite number, n, of samples, we believe that by sampling more intelligently we can get a better output as less samples are wasted on suboptimal arms.

[3]This fact follows from the Neyman-Pearson lemma in Statistics, which provides a bound on the power of a hypothesis test.

For each round $t = 1, 2, \ldots, n$:

(1) Sample each arm $\lfloor n/d \rfloor$ times

(2) Calculate $\text{argmax}_{c \in \mathcal{C}} \, c^\top \hat{\mu}$

Output: $c_n \in \mathcal{C}$ such that $\|c_n\|_1 = m$

Figure 2: Uniform Sampling for combinatorial identification

Theorem 3.1. *The probability of error of Uniform Sampling in the combinatorial identification problem satisfies*

$$e_n \leq m \exp(-\lfloor \frac{n}{d} \rfloor (\Delta^2))$$

Proof. We apply Hoeffding's inequality and a union bound.

$\mathbb{P}\{\hat{\mu}_{i,n} - \hat{\mu}_{i^*,n} \geq 0\} = \mathbb{P}\{(\hat{\mu}_{i,n} - \hat{\mu}_{i^*,n}) - (-\Delta_i) \geq \Delta_i\} \leq \exp(-\frac{2\lfloor \frac{n}{d} \rfloor^2 \Delta_i^2}{2\lfloor \frac{n}{d} \rfloor}) = \exp(-\lfloor \frac{n}{d} \rfloor \Delta_i^2)$

□

3.2. **Combinatorial SAR Algorithm.** (See Fig. 3) The SAR (Successive Accept and Rejects) algorithm was proposed by Bubeck et al., 2013 in [4] as a way to solve the m-best arms identification problem. The main idea behind the algorithm is its ability to Accept (Reject) an arm if it's confident enough that the arm is within (not within) the top m arms. In our variation for combinatorial identification, we retain the original algorithm while including an additional step where we calcuate an estimate of the maximum combinatorial structure, $\hat{c} = \text{argmax}_{c \in \mathcal{C}} \, c^\top \hat{\mu}$, during each phase.

Informally the algorithm proceeds as follows. First it divides the n rounds into $d - 1$ phases and maintains an intial active arms set A that contains all the d arms. During each phase, it samples each arm equally often and calculates an estimate of the optimal combinatorial structure $\hat{c}$ (e.g. in the maximum spanning tree problem it'll calculate a max. spanning tree of size m based on the empirical means at the end of that phase). Next it orders the empirical means of the arms in $\hat{c}$ (via a bijection $\sigma_{\hat{c}}$), then it orders the empirical means not in $\hat{c}$ (bijection $\sigma_{\hat{c}^\perp}$). Finally it creates a total ordering of the arms (via $\sigma_k = \sigma_{\hat{c}} + \sigma_{\hat{c}^\perp}$) by combining the two orderings. Then for each active arm it calculates an estimate for the gaps and removes the arm i_k with the highest gap from the active set. If the empirical mean of removed arm i_k is greater than the $(m(k) + 1)^{th}$ best empirical mean (as deterimined by our ordering) then we accept the arm i_k otherwise we reject it. In other words if we find that the arm with the largest gap is within our estimate for the top m arms in c_n we are reasonably confident that it belongs to the optimal structure and so we accept it. Similarly if the arm with the largest gap is not within our estimate for the top m arms we are reasonably confident that it does not belong in the optimal structure and we reject it. After the $d - 1$ phases, we output the m arms in our accepted set, where each arm $i \in c_n$.

The length of the phases are the same as in the original SR algorithm [1], and are chosen carefully to obtain an optimal (up to logarithmic factor) convergence rate.

$$\sum_{k=1}^{d-1} n_k = n_1 + n_2 + \ldots + n_{d-1} + n_{d-1} \leq d + \frac{n-d}{\overline{\log}(d)} (\frac{1}{2} + \sum_{k=1}^{d-1} \frac{1}{d+1-k}) = n.$$

It is important to realize that there are some issues present with this algorithm in a combinatorial setting. Unlike in the case where we're trying to find the m best arms, finding the optimal combinatorial structure is far more restrictive. This is because in the m best case the number of options for accepting other arms remain unlimited, whereas once an arm is chosen in the combinatorial case we restrict ourselves to simply the permutations of combinatorial structures containing that specific arm. As a result, optimizing locally has sizeable effects on our global optimization problem. Simply consider the case where you have two potential arms with equal means to choose from. In this scenario given that the corresponding gaps of the arms are the same one is equally likely to

choose either. However if it turns out that accepting the first arm restricts you to the subset of structures that contains the optimal structure and accepting the second arm does not, then the penalty (or reward) for choosing one arm is exceptionally high. Consequently in scenarios where there are many bad arms to choose from it is likely that Combinatorial SAR will not work efficiently.

Let $A_1 = \{1, \ldots, d\}, m(1) = m, \overline{\log}(d) = \frac{1}{2} + \sum_{i=2}^{d} \frac{1}{i}$, $n_0 = 0$ and for $k \in \{1, \ldots, d-1\}$,

$$n_k = \left\lceil \frac{1}{\overline{\log}(d)} \frac{n-d}{d+1-k} \right\rceil.$$

For each phase $k = 1, 2, \ldots, d-1$:

(1) for each active arm $i \in A_k$, select arm i for $n_k - n_{k-1}$ rounds.

(2) Calculate $\hat{c} = \text{argmax}_{c \in \mathcal{C}}\, c^\top \hat{\mu}$

(3) Let $\sigma_k = \sigma_{\hat{c}} + \sigma_{\hat{c}^\perp} : \{1, \ldots, d+1-k\} \to A_k$ be the bijection that orders the empirical means such that $\sigma_{\hat{c}}$ orders the empirical means of the m arms in $\hat{c}$ and $\sigma_{\hat{c}^\perp}$ orders the empirical means of the $(d+1-k)-m$ arms in $\hat{c}^\perp$. Combining the two orderings we get a total ordering σ_k over the $d+1-k$ arms in the phase.

(4) Given an ordering σ_k of the empirical means $\hat{\mu}_{\sigma_k(1),n_k} \geq \ldots \hat{\mu}_{\sigma_k(d+1-k),n_k}$. For $1 \leq r \leq d+1-k$, define the empircal gaps

$$\hat{\Delta}_{\sigma_k(r),n_k} = \begin{cases} \hat{\mu}_{\sigma_k(r),n_k} - \hat{\mu}_{\sigma_k(m(k)+1),n_k} & \text{if} \quad r \leq m(k) \\ \hat{\mu}_{\sigma_k(m(k)),n_k} - \hat{\mu}_{\sigma_k(r),n_k} & \text{if} \quad r \geq m(k)+1 \end{cases}$$

(5) Let $i_k \in \text{argmax}_{i \in A_k} \hat{\Delta}_{i,n_k}$ (ties broken arbitraily). Deactivate arm i_k, that is set $A_{k+1} = A_k \setminus \{i_k\}$.

(6) If $\hat{\mu}_{i_k,n_k} > \hat{\mu}_{\sigma_k(m(k)+1),n_k}$ then arm i_k is accepted and set $m(k+1) = m(k) - 1$

Output: The m accepted arms such that each arm $i \in c_n$.

Figure 3: SAR for combinatorial identification

3.3. **Combinatorial SR Algorithm.** (See Fig. 4) The SR (Successive Rejects) algorithm was proposed in Audibert et al., 2010 [1] as a parameter-free algorithm to find the arm with maximal mean. We propose a combinatorial version of this algorithm, where the key difference from the original algorithm is that we do not maintain an accepted arms set. More specifically the algorithm proceeds as follows. First it divides the n rounds into $d-1$ phases and maintains an intial active arms set A that contains all d arms. During each phase, it samples each arm equally often and removes the arm with the lowest empirical mean from A. The key difference is that instead of rejecting an arm altogether, we merely stop sampling it further. After the $d-1$ phases we calculate the maximum combinatorial structure c_n, taking into account the empirical means of all arms including the ones we rejected (i.e. the ones we simply stopped sampling further).

We propose this specific algorithm as a way to address the potential issues of using combinatorial SAR. Note that unlike in combinatorial SAR, we're not accepting arms nor outright rejecting arms. Consequently the potential combinatorial structures we can choose from at the end is unlimited. Instead we rely on the assumption that if after a number of samples the empirical mean of an arm looks bad, it's reasonable to stop further sampling the arm.

3.4. **Applications.**

Let $A_1 = \{1, \ldots, d\}, m(1) = m, \overline{\log}(d) = \frac{1}{2} + \sum_{i=2}^{d} \frac{1}{i}$, $n_0 = 0$ and for $k \in \{1, \ldots, d-1\}$,

$$n_k = \left\lceil \frac{1}{\overline{\log}(d)} \frac{n-d}{d+1-k} \right\rceil.$$

For each phase $k = 1, 2, \ldots, d-1$:

(1) for each active arm $i \in A_k$, select arm i for $n_k - n_{k-1}$ rounds.

(2) Let $A_{k+1} = A_k \setminus \text{argmin}_{i \in A_k} \hat{X}_{i,n_k}$. (remove only one element from A_k, if there is a tie, select randomly the arm to dismiss among the worst arms

Output: Calculate $c_n = \text{argmax}_{c \in \mathcal{C}} c^\top \hat{\mu}$, taking into account empirical means of all d arms.

Figure 4: SR (Successive Rejects) for combinatorial identification

3.4.1. *Maximum Weighted Bipartite Matching.* A particular combinatorial setting of interest is finding a maximum weighted bipartite matching (often referred to as the Assignment Problem). Consider the complete bipartite graph $K_{m,m}$ (two sets of vertices X and Y of size m where the set of edges consists of all possible links from one set to another) with distributions as edge weights. Let $\mathcal{C}$ contain all the perfect matchings of size m (a perfect matching is an injective mapping from $\{1, \ldots, m\} \in X$ to $\{1, \ldots, m\} \in Y$) thus $|\mathcal{C}| = m!$ matchings. The edges in $K_{m,m}$ correspond to the arms in a multi-armed bandit, where $d = m^2$. Thus our objective is to sample the arms on $K_{m,m}$ a total of n times and find the optimal combinatorial structure - the maximum weighted matching.

[4]
This combinatorial setting is motivated by a specific real world application - Ad placement. Given m ad's and m websites, how do we go about assigning ad's to websites such that the expected click-through rate (how many times the ad will be clicked on) of each ad will be as high as possible. Clearly we can model this problem as finding a maximum weighted matching on a complete bipartite graph, where the arms correspond to the click-through rate distribution of a specific ad placed on a specific website.

While there exist algorithms for finding the maximum weighted bipartite matching in an offline setting (most notably the Hungarian Algorithm see [10]), no efficient online versions exist for our specific setting where the edge weights are changing dynamically (one exists but is largely intractable to simulate see [8]). All three algorithms we propose (Uniform, Combinatorial SAR, Combinatorial SR) represent a reasonable approach to solving the problem, while being easy to implement and tractable for large simulations.

Another approach to formulating the problem is to state it in terms of online linear optimization. As shown in [2], we can use the Birkhoff-von Neumann Theorem to show that the convex hull of matchings on a bipartite graph is easily described. This is useful as it means we can state the problem in terms of linear optimization, as maximizing a linear function is the same as maximizing over the convex hull. While this formulation did not lead to any insights, we include it here as an interesting alternative approach to our problem.

Using the Birkhoff-von Neuman Theorem (Every doubly stochastic matrix is a convex combination of permutation matrices) we can describe the convex hull for bipartite matchings as follows.

Proposition 3.2. *Let $\mathcal{C}$ be the set of matchings of size m on $K_{m,m}$, then*

$$Conv(\mathcal{C}) = \left\{ \sum_{j=1}^{m} \sum_{i=1}^{m} x(i,j) = 1, \forall i, j \in \{1, \ldots, m\} \right\}$$

As a result our problem can be written as a linear optimization problem where we are maximizing over the convex hull

$$\max_{x \in Conv(\mathcal{C})} \langle x, \mu \rangle$$

3.4.2. *Maximum Weighted Spanning Tree.* Another combinatorial setting of interest is finding the maximum weighted spanning tree. Consider the complete graph K_{m+1} (set of vertices of size $m+1$ such that every pair of distinct vertices is connected by a unique edge) with distributions as edge weights. Let $\mathcal{C}$ contain all the spanning tree's of K_{m+1} (a spanning tree is a path through the entire graph that contains all the vertices and no cycles), Cayley's formula (see [10]) gives us $|\mathcal{C}| = m+1^{m-1}$. The edges in K_{m+1} correspond to the arms in a multi-armed bandit where $d = \frac{(m+1)(m}{2}$. Our objective is to sample the arms on K_{m+1} and find the maximum weighted spanning tree.

In related liteature this problem is often framed as finding the minimum weighted spanning tree (the MST problem) for which several efficient offline algorithms exist. Once again no efficient algorithms exist for our specific setting of dynamically changing edge weights. We run simulations of this particular setting in Section 4 where we use Kruskal's algorithm (see Fig. 5) to calculate our final optimal structure. Kruskal's algorithm is a greedy algorithm in graph theory for finding the minimum weighted spanning tree, to find the maximum weighted spanning tree simply multiply all edge weights by -1 and run the algorithm.

Input: non-null connected graph G and numbers $w(e)$ for every $e \in E(G)$

Let $m+1 = |V(G)|$

For each $i = 1, \ldots, m+1$:

(1) choose an edge e_i of G with $w(e_i)$ minimum such that $e_{i+1} \neq e_1, \ldots, e_i$ and $\{e_1, \ldots, e_i\} \cup \{e_{i+1}\}$ contains no cycles

Output: Spanning tree with edge set $\{e_1, \ldots, e_m\}$.

Figure 5: Kruskal's Algorithm

4. Experiments

In this section we run some simple experiments similar to those in Audibert et al., 2010 [1] for the problem of finding a maximum weighted spanning tree. Our objective in running these numerical simulations is to better inform and guide our theoretical analysis. We compare Uniform sampling to the performance of our two proposed algorithms, where in each case we use Kruskal's Algorithm at the end to calculate our final spanning tree.

In our experiments we consider only Bernoulli distributions, and the optimal arm always has parameter $\frac{1}{2}$. As outlined in Section 3.4.2 we only consider complete graphs K_m, with $\frac{m(m-1)}{2}$ edges which correspond to the number of arms d. We consider four different experiments where each experiment corresponds to a different situation for the gaps, either being clustered into a few group or distributed according to arthimetic or geometric progression. We run each experiment $T = 1000$ times and plot the probability of misidentification against the number of rounds n sampled. The parameters for the experiments are as follows:

- Experiment 1: Two groups of bad arms, $m = 7$, $d = 21$, $\mu_{2:6} = 0.42$, $\mu_{7:21} = 0.38$.
- Experiment 2: Geometric progression, $m = 4$, $d = 6$, $\mu_i = 0.5 - (0.37)^i$, $i \in \{2, 3, 4, 5, 6\}$.
- Experiment 3: Arithmetic progression, $m = 6$, $d = 15$, $\mu_i = 0.5 - 0.025i$, $i \in \{2, \ldots, 12\}$.

- Experiment 4: Three groups of bad arms, $m = 9$, $d = 36$, $\mu_{2:6} = 0.45$, $\mu_{7:20} = 0.43$, $\mu_{21:30} = 0.38$.

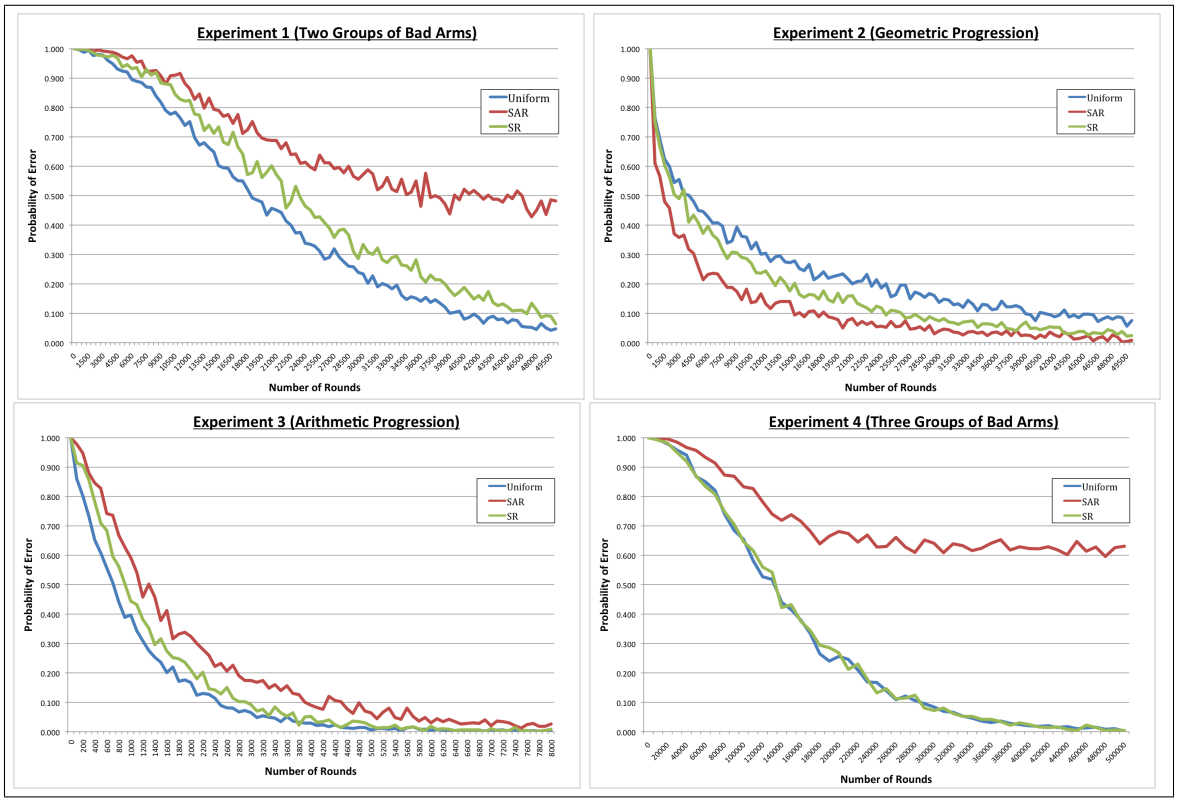

Figure 6: Simulation Results for Maximum Weighted Spanning Tree

Looking at the results in Fig. 6 we see that our algorithms performed much worse than expected. On the whole Combinatorial SAR performs the worst between the two, however it does perform the best for Experiment 2. Additionally as we suggested, SAR struggles in scenarios where it has many bad arms to differentiate amongst, most notably in Experiment 4.

On the other hand, Combinatorial SR performs much better than SAR. Our strategy to not explicitly accept or reject arms seems to have worked as SR performs well in the two experiments with groups of bad arms. Ultimately though, it at beast does as well as Uniform but never better.

5. Conclusion

In this thesis we have studied combinatorial identification in multi-armed bandits. Our goal has been to find a generalized sampling strategy for dynamically changing edge-weights on combinatorial structures, that performs better than uniform sampling. Specifically we proposed two algorithms, Combinatorial SAR & SR, which were extensions of previous work in [1] and [4]. Given the sucess of the two algorithms in the settings of *best* and *mbest* identification, we suspected they might extend to the general combinatorial setting. Despite finding the converse to be true, we consider our analysis and experimental results to be useful indicators of research direction. This particular problem has yet to be solved in any of the specific combinatorial settings that we considered, thus an attempt to solve the problem in general is not a trivial matter.

Given our limited theoretical understanding, we propose that further experiments should be considered. We believe that a proposed algorithmic solution will arrive from attempting several

variations of existing sampling strategies and working backwards to develop a theoretical understanding. We dismiss the claim that one cannot do better than uniform sampling as clearly one can improve one's recommendation by not wasting samples on suboptimal arms. Ultimatley though, based on our results we find that an entirely new algorithm must be formulated.

Appendix A.

In this Appendix we prove several results stated earlier.

A.1. **Hoeffding's Inequality.** We use a simply stated version of Hoeffding's Inequality.

Theorem A.1. *Let $X_1, \ldots, X_n$ be independent identically distributed random variables with mean μ, and bounded in the range [0,1], then*

$$\mathbb{P}(|\hat{X}_n - \mu| \geq t) \leq 2e^{-2nt^2} = 2\exp(-2nt^2)$$

For proofs and other statements of Hoeffding's inequality see [7].

A.2. **Kruskal's Algorithm.** We provide a simpler proof of Kruskal's Algorithm than in [10].

Definition A.2. *Let T be a spanning tree of G, and let $f \in E(G) - E(T)$. A cycle C of G with $f \in E(C)$ such that $C \setminus f$ is a path of T is called a* ***fundamental cycle of f with respect to T****.*

Proof of Kruskal's Algorithm.

Proof. Let $e_1, \ldots, e_m$ be the edges generated by Kruskal's algorithm. Choose $i \in \{1, \ldots, m\}$ maximum such that there is a minimum weighted spanning tree (mst) containing all of $e_1, \ldots, e_{i-1}$. We claim that $i = n$. Suppose to the contrary that $i \neq n$. Let T be a mst containing $e_1, \ldots, e_{i-1}$. Then $e_i \notin E(T)$ from the maximality of i. Let C be the fundamental cycle of e_i with respect to T. Since the algorithm chose edge e_i, there is no cycle in $\{e_1, \ldots, e_i\}$, and so some edge e of C is not in $\{e_1, \ldots, e_i\}$. There is no cycle included in $\{e_1, \ldots, e_{i-1}, e\}$ since all these edges belong to T. Furthermore since the algorithm chose e_i rather than e, it follows that $w(e_i) \leq w(e)$. But since e is an edge of the fundamental cycle of e_i with respect to T, we have that $w(e_i) \geq w(e)$, hence $w(e_i) = w(e)$. Now note that we can construct a spanning tree T' with edge-set $(E(T) \setminus e) \cup \{e_i\}$ and since $w(e_i) = w(e)$ it follows that T' is a mst. Hence $e_1, ..., e_i \in E(T')$ which contradicts the maximality of i. □

References

[1] J.-Y. Audibert, S. Bubeck, and R. Munos. Best arm identification in multi-armed bandits. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*, 2010.

[2] S. Bubeck. Introduction to online optimization. Lecture Notes, 2011.

[3] S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in multi-armed bandits problems. In *Proceedings of the 20th International Conference on Algorithmic Learning Theory (ALT)*, 2009.

[4] S. Bubeck, T. Wang, and N. Viswanathan. Multiple identifications in multi-armed bandits. In *Proceedings of the Thirtieth International Conference on Machine Learning (ICML)*, 2013.

[5] N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 2012.

[6] V. Gabillon, M. Ghavamzadeh, A. Lazaric, and S. Bubeck. Multi-bandit best arm identification. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[7] L. Koralov and Y. Sinai. *Probability Theory.* Springer, 2007.

[8] G. Ayorkor Mills-Tetty, A. Stentz, and M. Bernardine Dias. The dynamic hungarian algorithm for the assignment problem with changing costs. Carnegie Mellon Robotics Institute Technical Report 149, 2007.

[9] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematics Society*, 58:527–535, 1952.

[10] D. West. *An Introduction to Graph Theory.* Prentice Hall, 2001.