Errata

Johan Hallberg Szabadváry

20 augusti 2022

Page 13

In Assumption 3.1, first line should read

There exists a (strictly) convex function ψ on $\mathbb{R}$ such that for all $\lambda \geq 0$

Page 14

From line 16, remove the following:

Note in particular that Assumption 3.1 implies that, $\psi(0) = 0$, which in turn means that $\psi^*(0) = \sup_\lambda(\lambda \cdot 0 - \psi(\lambda)) = \sup_\lambda(-\psi(\lambda)) = 0$ since ψ is convex and $(0) = 0$. Moreover,

The next sentence should just be:

The LF transform is always convex irrespective of the shape of the original function.

Page 19

From line 3, remove the following:

Note however that these bounds follow with ψ^* and $\nu = 1$ if all distributions satisfy Assumption 3.1. Therefore, we will assume that all distributions satisfy that there exists a convex function ψ so that (3.5) holds. To be explicit our assumption implies, for all i, m

$$\mathbb{P}(|\overline{X}_{i,s}^{(m)} - \mu_i^{(m)}| > \varepsilon) < e^{-s\psi^*(\varepsilon)}. \tag{3.15}$$

Page 23

Figures 2 and 3. Legend is wrong in MF-UCB. Blue should be $m = 3$, green is $m = 2$ and red is $m = 1$

Page 57

From line 20, remove the following:

and that $(\psi^*)^{-1}$ is a convex function with $(\psi^*)^{-1}(0) = 0$ (see eq. (3.6)). In particular, this means that $c_{t,s}$ increases with t, and decreases with s.

SJÄLVSTÄNDIGA ARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

Single- and Multi-Fidelity Bandits in Monte Carlo Tree Search - from the Casino to Mobile Network Optimization

av

Johan Hallberg Szabadváry

2022 - No M5

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET, 106 91 STOCKHOLM

Single- and Multi-Fidelity Bandits in Monte Carlo Tree Search - from the Casino to Mobile Network Optimization

Johan Hallberg Szabadváry

Självständigt arbete i matematik 30 högskolepoäng, avancerad nivå

Handledare: Maxime Bouton, Alessandro Previti and Yishao Zhou

2022

Abstract

Mobile networks are made of several base stations, each one with a number of antennas, that require the optimization of several parameters in order to provide good service. Automatic optimization of these parameters is difficult as the size of the search space is exponential in the number of antennas, which is typically large. The recent success of Monte Carlo Tree Search methods in large and complex games, such as Go, suggests that similar methods might be used in large discrete black-box optimization, as they can be modeled as sequential decision-making problems. In this thesis, we study and apply a version of the famous upper confidence bound for trees (UCT) algorithm to black-box optimization. We state and prove a form of the UCT algorithm that allows for more general distributions. Moreover, we suggest a novel *Multi-fidelity* UCT algorithm based on the multi-fidelity bandit problem. Experimental results indicate that this new algorithm finds at least as good solutions in a shorter time, given an existing low-fidelity approximation. We discuss and empirically test different low-fidelity approximations.

Sammanfattning

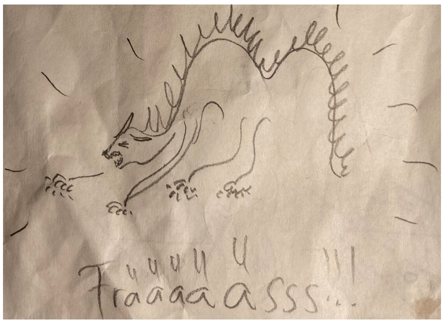
Mobilnätverk består av flera basstationer, var och en med ett antal antenner som kräver optimering av ett flertal parametrar för att leverera bra service. Automatisk optimering av dessa parametrar är svår, eftersom storleken på sökrummet ökar exponentiell med antelet antenner, som vanligen är stort. På senare tid har Monte Carlo trädsökning skördat framgångar i stora och komplicerade spel såsom Go, vilket antyder att liknande metoder skulle kunna användas till storskalig diskret *black-box* optimering. I den här uppsatsen analyseras och appliceras en version av den berömda upper övre konfidensgräns för träd (UCT) algoritmen på *black-box* optimering. Vi formulerar och bevisar en form av UCT som tillåter mer generella sannolikhetsdistributioner. Vidare föreslår vi en ny *Multi-fidelity* UCT algoritm, baserad på en multi-fidelity variant av det flerarmade banditproblemet. Experimenten visar att den nya algoritmen hittar minst lika bra lösningar på kortare tid, given en low-fidelity approximering. Vi disskuterar och prövar olika low-fidelity approximationer empiriskt.

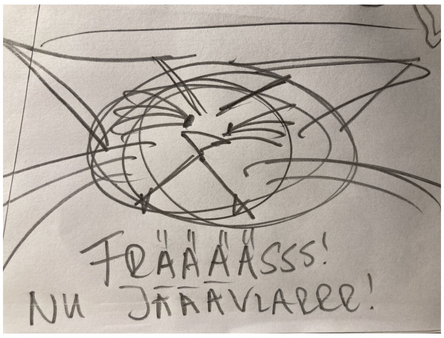
Acknowledgements

First, I want to thank my supervisors at Ericsson Research; Maxime Bouton and Alessandro Previti. Thank you for your great patience with a mathematician's attempt at programming. You have pointed me in the right direction and kept me on track in our weekly meetings. I have enjoyed working with you, and I am grateful for your input and practical help. I am a much better programmer coming out of this process than going in.

I also want to thank Yishao Zhou, my academic supervisor. Thank you for your endless support and encouragement. You push and inspire me to reach higher than I thought possible. Not just in this project, but throughout my education. I am ever grateful for having you as a mentor.

Finally, my wife, Kinga Szabadváry. I could not have done this without you. You have supported me in every way possible during my education. In particular, thank you for the inspirational cat drawings.





Contents

1 Introduction

A major factor in determining the performance of a mobile network is the configuration of base station antennas, and in particular the tilt angle. Poorly configured antennas in a mobile network can interfere with nearby antennas and deteriorate the signal of users that could otherwise have good coverage.

Existing methods for mobile network organization typically fall into one of four categories. Either, optimization is done by hand by an engineer with expert knowledge. This is often both difficult and costly, as the engineer has to anticipate many traffic conditions and possible sources of interference. Alternatively, one may use some kind of self organizing network (SON) [1]. They use rule-based methods and typically suffer from issues with scalability. Methods relying on mathematical optimization are also available [2], but typically requires so much simplification that they can not capture interference between antennas. Finally, reinforcement learning (RL) methods [3–8] are more robust, but due to the typically, large scale complexity, they usually treat network entities as independent agents, again failing to take into account the benefit of configuring interacting entities together. The resulting solutions are suboptimal despite the method's scalability. A challenge with using RL methods is that they typically need a lot of data. The lack of coordination can be addressed by e.g. a message passing algorithm, [9] which however still needs lots of data, and also has to model the network using some approximation.

A black-box function is a function for which no analytic expression is known, and black-box optimization is the problem of optimizing such a function. Note that a black-box function is not necessarily an irregular function. It could well be both smooth and convex and as nice as can be. This information is however not available to us in black-box optimization. Examples of black-box functions include for example legacy computer code that is not available for examination and real world experiments such as chemical reactions, the number of clicks on a website ad et cetera. Depending on the cost of evaluating the function, we speak of cheap and expensive black-box optimization, where these terms are somewhat arbitrary. In cheap black-box optimization, we have more options available that rely on many function evaluations, such as methods that use gradient approximations. Expensive black-boxes are more challenging, and typically one uses some type of surrogate model to approximate the expensive function.

The expected outcome of making a move in a board game like Chess or Go, for example, can be viewed as a black-box function. Monte Carlo Tree Search (MCTS) is a family of algorithms originally designed for automatic game play. They are an important part of DeepMind's *AlphaGo* [10] that plays Go at superhuman level. Essentially, playing Go successfully is nothing other than repeatedly maximizing the black-box function of the expected outcome of each legal move from the current game state. The success of MCTS in game playing therefore suggests that similar methods could be applied to black-box optimization. One challenge that arises in this is that in a game like Go there is a natural order in which moves are made, which means that MCTS can consider the long term effect, and prioritize moves that are promising in the long run over short term greedy moves. In black-box optimization, a move corresponds to assigning a value to a variable, which can be done in any order. However, MCTS works sequentially, which imposes an arbitrary order in which values are assigned. Essentially it is as we play a rather strange game, where the order of the moves is not set. This introduces the problem of choosing an order of the variables. Since variables can interact in complicated ways that can be hard to anticipate, the order may have a big impact on the performance of MCTS in black-box optimization.

Optimizing the tilt angles of base station antennas can be viewed as black-box optimization. The black-box function to be optimized maps an antenna tilt configuration to some measure of network performance. We will use a measure that essentially is the average quality of service for every user in the network. Using a network simulator, the computation time of this is roughly linear in the number of users, which makes the function expensive given enough users.

This thesis studies the theoretical foundations for the famous *upper confidence bound for trees* (UCT) algorithm, which was proposed by Kocsis and Szepesvári (2006) [11]. The UCT algorithm iteratively builds a search tree by modeling node selection as a separate *stochastic multi-armed bandit problem*. We propose to apply UCT to mobile network optimization, treating the problem as *black-box optimization*. Moreover, building on the work of Kandasamy, Dasarathy, Schneider and Póczos (2016) [12] on the *multi-fidelity multi-armed bandit problem* we introduce a novel multi-fidelity version of UCT (MF-UCT) that uses one or more computationally cheaper approximations of an expensive function to be optimized, in order to exclude poor configurations early in the search, thereby decreasing the search space. This approach corresponds to using surrogate models in more traditional black-box optimization methods.

Experimental results show that MF-UCT performs at least as well as UCT on an antenna tilt optimization problem. Given a pre-existing approximation of the reward function, it reaches good results faster. Both UCT and MF-UCT imposes an arbitrary ordering of the antennas. This order impacts the convergence time of the algorithms, and the difference can be quite large. We suggest two ways to find a good order in this work.

The main contributions of this thesis are to adapt and apply Monte Carlo tree search to discrete black-box optimization, and suggesting ways to handle the imposed arbitrary ordering of the variables. Moreover, we suggest and experimentally evaluate a new multi-fidelity algorithm, allowing for faster computation times in some situations. Finally, we prove a version of the UCT algorithm that allows for more general probability distributions.

The thesis is organized as follows. Section 2 presents some probability inequalities that will be needed later. Section 3 introduces the classical UCB algorithm for solving the stochastic multi-armed bandit problem. Following Bubeck and Cesa-Bianchi [13], we present the more general (α, ψ)-UCB algorithm. We also introduce the multi-fidelity multi-armed bandit problem, and a Multi-fidelity UCB algorithm. Section 4 gives a brief introduction to the Monte Carlo tree search family of algorithms. We also present the classical UCT algorithm and prove its convergence and consistency under more general assumptions than is usually seen. Finally, a new Multi-fidelity version of UCT (MF-UCT) is presented and discussed. We formulate more precisely the problem of optimizing a mobile network in Section 5, and discuss how to modify UCT for black-box optimization. Our experimental results are shown together with a discussion on antenna ordering. Finally, we discuss the results and some directions for future research in Section 6. Most technical proofs and a section on convex analysis can be found in the appendices.

2 Some background on probability

This section presents some concepts from probability theory that we will require. In particular we introduce concentration inequalities Lemma 2.6 that we require to prove the convergence of the UCT algorithm.

Recall that the *expectation* of the random variable X, defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where the set Ω is the sample space, the σ-algebra $\mathcal{F}$ on Ω is the event space, and $\mathbb{P}$ is a probability measure on the measurable space $(\Omega, \mathcal{F})$ is defined as the Lebesgue integral

$$\mathbb{E}[X] := \int_{\Omega} X d\mathbb{P} = \int_{\Omega} X(\omega)\mathbb{P}(d\omega).$$

We will also need to consider *conditional expectation.* We introduce this concept with a theorem due to Kolmogorov.

Theorem 2.1. *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, and X a random variable with $\mathbb{E}[|X|] < \infty$. Then, for any sub-σ-algebra $\mathcal{G}$ of $\mathcal{F}$, there exists a random variable Y that is $\mathcal{G}$ measurable. Moreover $\mathbb{E}[|Y|] < \infty$, and for every set $G \in \mathcal{G}$,*

$$\int_G Y d\mathbb{P} = \int_G X d\mathbb{P}.$$

If $\tilde{Y}$ is another random variable with these properties, $\tilde{Y} = Y$ almost surely (i.e. with probability 1). A random variable with these properties is called a version of the conditional expectation $\mathbb{E}[X|\mathcal{G}]$ *of X given $\mathcal{G}$.*

The proof of this fundamental theorem can be found e.g. in chapter nine in Williams' book Probability with martingales [14]. Williams also proves two important properties of conditional expectation that we will need: The *tower law* says that if $\mathcal{H}$ is a sub-σ-algebra of $\mathcal{G}$ (both are sub-σ-algebras of $\mathcal{F}$), then $\mathbb{E}[\mathbb{E}[X|\mathcal{G}]|\mathcal{H}] = \mathbb{E}[X|\mathcal{H}]$ almost surely. Secondly, the *"taking out what is known"* property says that if Z is a bounded $\mathcal{G}$ measurable random variable, then $\mathbb{E}[ZX|\mathcal{G}] = Z\mathbb{E}[X|\mathcal{G}]$ almost surely.

Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, an increasing family of sub-σ-algebras $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \subseteq \mathcal{F}$ of $\mathcal{F}$ is called a *filtration,* and $(\Omega, \mathcal{F}, \{\mathcal{F}_n\}, \mathbb{P})$ is called a filtered space. A stochastic process $\{X_n\}$ is called *adapted* to the filtration $\{\mathcal{F}_n\}$, or $\{\mathcal{F}_n\}$-adapted if, for all n, X_n is $\mathcal{F}_n$-measurable.

Definition 2.1 (Martingale)**.** *A stochastic process $\{X_n\}$ is called a* martingale *relative to the filtration $\{F_n\}$, if $\{X_n\}$ is $\{F_n\}$-adapted, $\mathbb{E}[|X_n|] < \infty$ for all n, and the conditional expectation*

$$\mathbb{E}[X_n|\mathcal{F}_{n-1}] = X_{n-1}$$

almost surely (i.e. with probability 1) for all $n \geq 1$. If the equality sign is exchanged for "$\leq$" or "$\geq$", $\{X_n\}$ is called a supermartingale *or* submartingale *respectively.*

For example, if X is a random variable with finite expectation, and $\{\mathcal{F}_t\}$ is a filtration over the same probability space, then the sequence of conditional expectations, $Y_t = \mathbb{E}[X|\mathcal{F}_t]$ is a martingale. In particular, this is called a Doob martingale. To see that it is a martingale, note that

$$\begin{gathered}\mathbb{E}[|Y_t|] = \mathbb{E}[|\mathbb{E}[X|\mathcal{F}_t]|] \leq \mathbb{E}[\mathbb{E}[|X||\mathcal{F}_t]] = \mathbb{E}[|X|] < \infty, \quad \text{and} \\ \mathbb{E}[Y_t|\mathcal{F}_{t-1}] = \mathbb{E}[\mathbb{E}[X|\mathcal{F}_t]|\mathcal{F}_{t-1}] = \mathbb{E}[X|\mathcal{F}_{t-1}] = Y_t \;\; \text{since } \mathcal{F}_{t-1} \subseteq \mathcal{F}_t.\end{gathered}$$

The Kullback-Leibler divergence (KL-divergence) is an important concept in information theory and machine learning [15]. It is closely related to relative entropy, and is a popular measure way to measure the distance between two probability distributions [16]. Let P, Q be probability distributions of continuous random variables, defined on the same probability space. We define

$$D_{KL}[P||Q] := \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} dx \tag{2.1}$$

where p, q are the probability densities of P and Q. In machine learning, the KL-divergence is used, for example in the Cross entropy method [17].

We move on to establish some standard inequalities from probability theory.

Theorem 2.2 (Markov's inequality)**.** *If X is a non-negative random variable and $t > 0$, then*

$$\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}[X]}{t}.$$

where $\mathbb{E}$ denotes the expected value of X.

Proof. Let $\{\Pi(x)\}$ be the indicator function of the event $\Pi(x)$ for any predicate Π. Then

$$\mathbb{P}(X \geq t) = \mathbb{E}[\{X \geq t\}] \leq \mathbb{E}\left[\frac{X}{t}\{X \geq t\}\right] = \frac{1}{t}\mathbb{E}[X\{X \geq t\}] \leq \frac{1}{t}\mathbb{E}[X].$$

□

The second inequality follows as a corollary of Markov's inequality

Corollary 2.1 (Chernoff's Bounding Method)**.** *Let X be a real random variable. Then, for all $t > 0$ it holds that*

$$\mathbb{P}(X \geq t) \leq \inf_{s>0} e^{-st}\mathbb{E}[e^{sX}].$$

Proof. Fix $s > 0$, Markov's inequality yields

$$\mathbb{P}(X \geq t) = \mathbb{P}(sX \geq st) = \mathbb{P}(e^{sX} \geq e^{st}) \leq e^{-st}\mathbb{E}[e^{sX}].$$

The result follows as s was arbitrary. □

Theorem 2.3 (Hoeffding's lemma)**.** *Let X be a real valued stochastic variable such that $X \in [a,b]$ almost surely, and $\mathbb{E}[X] = 0$. Then, for all $\lambda \in \mathbb{R}$,*

$$\ln(\mathbb{E}[e^{\lambda X}]) \leq \frac{\lambda^2(b-a)^2}{8}.$$

Proof. The function $x \mapsto e^{\lambda x}$ is convex, so, since $X \in [a,b]$ almost surely, we have

$$e^{\lambda X} \leq \frac{b-X}{b-a}e^{\lambda a} + \frac{X-a}{b-a}e^{\lambda b}$$

almost surely. Taking expectation on both sides and recalling that $\mathbb{E}[X] = 0$ yields

$$\mathbb{E}[e^{\lambda X}] \leq \frac{be^{\lambda a} - ae^{\lambda b}}{b-a} = e^{g(\lambda(b-a))},$$

where $g(h) = \frac{ha}{b-a} + \ln(1 + \frac{a(1-e^h)}{b-a})$, which is somewhat tedious but straightforward to verify. The function g is twice differentiable, and we compute

$$g'(h) = \frac{a}{b-a} - \frac{ae^h}{b-ae^h}, \quad \text{and} \quad g''(h) = -\frac{abe^h}{(b-ae^h)^2}.$$

Thus $g(0) = g'(0) = 0$. Moreover, we have $g''(h) \leq \frac{1}{4}$ for all $h \in \mathbb{R}$, since if this was not true, then there would exist some real h_0 such that $-4abe^{h_0} > b^2 - 2abe^{h_0} + a^2e^{2h_0}$ which is equivalent to $0 > (b + ae^{h_0})^2$, a contradiction. So we have $g''(h) \leq \frac{1}{4}$ for all real h. By Taylor's theorem, there is some $\theta \in [0,1]$ such that

$$g(h) = g(0) + hg'(0) + \frac{1}{2}h^2g''(h\theta) = \frac{1}{2}h^2g''(h\theta) \leq \frac{h^2}{8}.$$

Since the exponential function is strictly monotonically increasing, taking logarithms at both sides of the inequality $\mathbb{E}[e^{\lambda X}] \leq e^{\frac{1}{8}\lambda^2(b-a)^2}$ finishes the proof. □

Remark Hoeffding's lemma can easily be extended to the case when the expectation is conditional. If $\mathcal{F}$ is a σ-algebra such that $\mathbb{E}[X|\mathcal{F}] = 0$, then, replacing $\mathbb{E}[X]$ by $\mathbb{E}[X|\mathcal{F}]$ everywhere in the proof yields

$$\ln(\mathbb{E}[e^{\lambda X}|\mathcal{F}]) \leq \frac{\lambda^2(b-a)^2}{8}. \tag{2.2}$$

Theorem 2.4 (Hoeffding-Azuma inequality)**.** *Let $\{X_i\}_{i=0}^n$ be a martingale relative to some filtration $\{\mathcal{F}_n\}$, such that for all $i > 0$, there is a constant c_i such that $|X_i - X_{i-1}| \leq c_i$. Then*

$$\mathbb{P}(X_n - X_0 \geq \varepsilon) \leq e^{-\frac{\varepsilon^2}{2\sum_{i=1}^n c_i^2}}.$$

Proof. Markov's inequality and some algebraic manipulation yields, for any real λ,

$$\begin{aligned}\mathbb{P}(X_n - X_0 \geq \varepsilon) &= \mathbb{P}(e^{\lambda(X_n - X_0)} \geq e^{\lambda\varepsilon}) \\ &\leq e^{-\lambda\varepsilon}\mathbb{E}[e^{\lambda(X_n - X_0)}] \\ &= e^{-\lambda\varepsilon}\mathbb{E}\left[e^{\lambda \sum_{i=1}^n (X_i - X_{i-1})}\right] \\ &= e^{-\lambda\varepsilon}\mathbb{E}\left[\prod_{i=1}^n e^{\lambda(X_i - X_{i-1})}\right].\end{aligned}$$

Using the tower property of conditional expectation, we get

$$\mathbb{P}(X_n - X_0 \geq \varepsilon) \leq e^{-\lambda\varepsilon}\mathbb{E}\left[\mathbb{E}\left[\prod_{i=1}^n e^{\lambda(X_i - X_{i-1})}\middle|\mathcal{F}_{n-1}\right]\right].$$

Using the "Taking out what is known" property, we therefore have

$$\mathbb{P}(X_n - X_0 \geq \varepsilon) \leq e^{-\lambda\varepsilon}\mathbb{E}\left[\prod_{i=1}^{n-1} e^{\lambda(X_i - X_{i-1})}\mathbb{E}\left[e^{\lambda(X_n - X_{n-1})}\middle|\mathcal{F}_{n-1}\right]\right].$$

Now, since $\{X_n\}$ is a martingale, $\{X_n - X_{n-1}\}$ is a martingale difference sequence, so in particular $\mathbb{E}[X_n - X_{n-1}|\mathcal{F}_{n-1}] = 0$, and by assumption we have $|X_n - X_{n-1}| < c_n$. We can therefore use the conditional version of Hoeffding's lemma to conclude that

$$\mathbb{E}[e^{\lambda(X_n - X_{n-1})}|\mathcal{F}_{n-1}] \leq e^{\frac{\varepsilon^2(2c_n)^2}{8}} = e^{\frac{\varepsilon^2 c_n^2}{2}},$$

which means that

$$\mathbb{P}(X_n - X_0 \geq \varepsilon) \leq e^{-\lambda\varepsilon}e^{\frac{\varepsilon^2 c_n^2}{2}}\mathbb{E}\left[\prod_{i=1}^{n-1} e^{\lambda(X_i - X_{i-1})}\right].$$

Repeating the argument $n-1$ times yields

$$\mathbb{P}(X_n - X_0 \geq \varepsilon) \leq e^{-\lambda\varepsilon}e^{\frac{\lambda^2}{2}\sum_{i=1}^n c_i^2} = e^{\frac{\lambda^2 \sum_{i=1}^n c_i^2}{2} - \varepsilon\lambda}.$$

This is true for any real λ, but we want the exponent to be $-\frac{\varepsilon^2}{2\sum_{i=1}^n c_i^2}$. Hence, we solve the quadratic equation

$$\frac{\lambda^2 \sum_{i=1}^n c_i^2}{2} - \varepsilon\lambda = -\frac{\varepsilon^2}{2\sum_{i=1}^n c_i^2}$$

which has the unique solution $\lambda = \frac{\varepsilon}{\sum_{i=1}^n c_i^2}$. With this choice of λ, the theorem follows. □

Let $\mathcal{F}_t$ denote a filtration over some probability space. To establish tail inequalities for stopped martingales, we prove some simple bounds here

Lemma 2.1. *Let N be an integer-valued random variable and let S_t be an $\mathcal{F}_t$-adapted real-valued process (not necessarily a martingale) which is centered, meaning that $\mathbb{E}[S_t] = 0$. Pick any $\varepsilon > 0$ and integers $0 \leq a < b$. Then*

$$\mathbb{P}(S_N \geq \varepsilon N) \leq (b-a+1) \max_{a \leq t \leq b} \mathbb{P}(S_t \geq \varepsilon t) + \mathbb{P}(N \notin [a,b]),$$
$$\mathbb{P}(S_N \leq -\varepsilon N) \leq (b-a+1) \max_{a \leq t \leq b} \mathbb{P}(S_t \leq -\varepsilon t) + \mathbb{P}(N \notin [a,b]).$$

Proof. Note that

$$\begin{aligned}\mathbb{P}(S_N \geq \varepsilon N) &\leq \mathbb{P}(S_N \geq \varepsilon N, a \leq N \leq b) + \mathbb{P}(N \notin [a,b]) \\ &= \underbrace{\mathbb{P}(S_N \geq \varepsilon N | a \leq N \leq b)\mathbb{P}(a \leq N \leq b)}_{p} + \mathbb{P}(N \notin [a,b]).\end{aligned}$$

Let $\{\Pi(x)\}$ be the indicator function of the event $\Pi(x)$ for any predicate Π, that is, $\{\Pi(x)\} = 0$ if $\Pi(x)$ is true and 0 otherwise. Then we have

$$\begin{aligned}\mathbb{P}(S_N \geq \varepsilon N | a \leq N \leq b) &= \mathbb{E}[\{S_N \geq \varepsilon N\} | a \leq N \leq b] \\ &\leq \mathbb{E}\left[\sum_{i=a}^{b} \{S_i \geq \varepsilon i\} | a \leq N \leq b\right] \\ &= \sum_{i=a}^{b} \mathbb{P}(S_i \geq \varepsilon i | a \leq N \leq b).\end{aligned}$$

Hence

$$p \leq \sum_{i=a}^{b} \mathbb{P}(S_i \geq \varepsilon i) \leq (b-a+1) \max_{a \leq t \leq b} \mathbb{P}(S_t \geq \varepsilon t)$$

which yields the desired inequality. The other inequality is shown analogously.

□

The next result is a corollary of this lemma and the Hoeffding-Azuma inequality. It generalizes the Hoeffding-Azuma inequality for S_N when N is an integer-valued random variable.

Lemma 2.2 (Hoeffding-Azuma inequality for stopped martingales)**.** *Assume that S_t is a centered martingale ($\mathbb{E}[S_t] = 0$), such that the corresponding martingale difference process is uniformly bounded by $C > 0$. Then, for any $\varepsilon > 0$ and integers $0 \leq a < b$, the following inequalities hold:*

$$\mathbb{P}(S_N \geq \varepsilon N) \leq (b-a+1)e^{-\frac{a\varepsilon^2}{2C^2}} + \mathbb{P}(N \notin [a,b]),$$
$$\mathbb{P}(S_N \leq -\varepsilon N) \leq (b-a+1)e^{-\frac{a\varepsilon^2}{2C^2}} + \mathbb{P}(N \notin [a,b]).$$

Lemma 2.3. *Let $\{Z_i\}, i = 1, \dots, n$ be a sequence of random variables such that Z_i is conditionally independent of $Z_{i+1}, \dots, Z_n$ given $Z_1, \dots, Z_{i-1}$, and let f be a C-Lipschitz function of n variables, i.e. there is a constant $C > 0$ such that*

$$|f(x_1, \dots, x_n) - f(x_1, \dots, x_{i-1}, x_i', \dots, x_n)| \leq C$$

holds for all $x_1, \dots, x_n, x_i'$ in the domain of f. Then the Doob martingale, $X_i = \mathbb{E}[f(Z_1, \dots, Z_n)|Z_1, \dots, Z_i]$ has bounded differences. In particular,

$$|X_{i+1} - X_i| \leq C.$$

Proof. Let us write $Z = (Z_1, \dots, Z_n)$ for a more compact notation. Let Z_{i+1}' be an independent copy of Z_{i+1}, and let $Z' = (Z_1, \dots, Z_i, Z_{i+1}', \dots, Z_n)$. Then,

$$\begin{aligned} |X_{i+1} - X_i| &= |\mathbb{E}[f(Z)|Z_1, \dots, Z_{i+1}] - \mathbb{E}[f(Z)|Z_1, \dots, Z_i]| \\ &= |\mathbb{E}[f(Z)|Z_1, \dots, Z_{i+1}] - \mathbb{E}[f(Z')|Z_1, \dots, Z_i]| \\ &= |\mathbb{E}[f(Z)|Z_1, \dots, Z_{i+1}] - \mathbb{E}[f(Z')|Z_1, \dots, Z_{i+1}]| \\ &= |\mathbb{E}[f(Z) - f(Z')|Z_1, \dots, Z_{i+1}]| \\ &\leq \mathbb{E}[|f(Z) - f(Z')||Z_1, \dots, Z_{i+1}] \\ &\leq C, \end{aligned}$$

where we used the independence of the Z_i in lines two and three, and the linearity of conditional expectation in line four. Finally, we used that f is C-Lipschitz in the last line. □

Lemma 2.4. *Let $N = \sum_{i=1}^n Z_i$, where $Z_1, \dots, Z_n$ are random variables that take the values 0 or 1. We assume that Z_i is adapted to the filtration $\{\mathcal{F}_i\}_t$ and that Z_{i+1} is conditionally independent of $Z_{i+2}, \dots, Z_n$, given $\mathcal{F}_i$. Then we have, for $\varepsilon > 0$*

$$\mathbb{P}(N - \mathbb{E}[N] > \varepsilon) \leq e^{-\frac{\varepsilon^2}{2n}}, \quad \textit{and} \quad \mathbb{P}(N - \mathbb{E}[N] < -\varepsilon) \leq e^{-\frac{\varepsilon^2}{2n}}.$$

Proof. The function $f(Z_1, \dots, Z_n) = \sum_{i=1}^n Z_i$ is 1-Lipschitz. Hence, the Doob martingale $X_i = \mathbb{E}[N|Z_1, \dots, Z_i]$ is a bounded difference martingale with bound 1 by Lemma 2.3. Applying the Hoeffding-Azuma inequality to the centered martingale $X_i - \mathbb{E}[N]$ we get the desired result. □

Lemma 2.5. *Let Z_i be as in Lemma 2.4, and $N_n = \sum_{i=1}^n Z_n$. If a_n is an upper bound on $\mathbb{E}[N_n]$, then for all $\Delta > 0$, if n is such that $a_n \leq \Delta/2$, then*

$$\mathbb{P}(N_n \geq \Delta) \leq e^{-\frac{\Delta^2}{8n}}.$$

Proof. We have

$$\mathbb{P}(N_n \geq \Delta) = \mathbb{P}(N_n > \mathbb{E}[N_n] + \Delta - \mathbb{E}[N_n]) \leq \mathbb{P}(N_n > \mathbb{E}[N_n] + \Delta/2),$$

since $\mathbb{E}[N_n] \leq a_n \leq \Delta/2$ by assumption. By lemma 2.4 we obtain the bound as stated. □

Lemma 2.6. *Let $Z_i, \mathcal{F}_i, a_i$ be as in Lemma 2.5. Let $\{X_i\}$ be a sequence of mutually independent random variables, all with the same probability distribution with mean μ. We say that $\{X_n\}$ is an i.i.d. sequence. Let $\{Y_i\}$ an $\mathcal{F}_i$-adapted process with $X_i, Y_i \in [0,1]$. Let*

$$S_n = \sum_{i=1}^{n} \Big((1 - Z_i)X_i + Z_i Y_i \Big).$$

and $\Delta = 9\sqrt{2n \ln(2/\delta)}$ for fixed $\delta > 0$, and

$$R_n = \mathbb{E}\Big[\sum_i X_i \Big] - \mathbb{E}[S_n].$$

Then for all n such that $a_n \leq \Delta/9$ and $|R_n| \leq (4/9)\Delta$, we have

$$\mathbb{P}(S_n \geq \mathbb{E}[S_n] + \Delta) \leq \delta \quad \textit{and} \quad \mathbb{P}(S_n \leq \mathbb{E}[S_n] - \Delta) \leq \delta.$$

Proof. We shall only prove the first inequality, since the second can be shown similarly. We have $S_n = \sum_{i=1}^n X_i + \sum_{i=1}^n Z_i(Y_i - X_i) \leq \sum_{i=1}^n X_i + 2\sum_{i=1}^n Z_i$. Therefore,

$$\mathbb{P}(S_n \geq \mathbb{E}[S_n] + \Delta) \leq \mathbb{P}\Big(\sum_{i=1}^n X_i + 2\sum_{i=1}^n Z_i \geq \mathbb{E}\Big[\sum_{i=1}^n X_i \Big] - R_n + \Delta \Big) \tag{2.3}$$

Using the trivial inequality $\mathbb{P}(A + B \geq \Delta) \leq \mathbb{P}(A \geq \alpha\Delta) + \mathbb{P}(B(1-\alpha)\Delta)$, which holds for any $A, B \geq 0$ and any $\alpha \in [0,1]$, the right-hand side of (2.3) is bounded above by

$$\mathbb{P}\Big(\sum_{i=1}^n X_i \geq \mathbb{E}\Big[\sum_{i=1}^n X_i \Big] + \Delta/9 \Big) + \mathbb{P}\Big(2\sum_{i=1}^n Z_i \geq \frac{8}{9}\Delta - R_n \Big).$$

Note that

$$\Bigg| \sum_{i=1}^n X_i - \mathbb{E}\Big[\sum_{i=1}^n X_i \Big] - \sum_{i=1}^{n-1} X_i + \mathbb{E}\Big[\sum_{i=1}^{n-1} X_i \Big] \Bigg| = |X_n - \mu| \leq 1$$

so by the Hoeffding-Azuma inequality,

$$\begin{aligned} \mathbb{P}\Big(\sum_{i=1}^n X_i - \mathbb{E}\Big[\sum_{i=1}^n X_i \Big] \geq \Delta/9 \Big) &= \mathbb{P}\Big(\sum_{i=1}^n X_i - \mathbb{E}\Big[\sum_{i=1}^n X_i \Big] \geq \sqrt{2n\ln(2/\delta)} \Big) \\ &\leq e^{-\frac{2n\ln(2/\delta)}{2n}} = \frac{\delta}{2}. \end{aligned}$$

Note further that, since by assumption $|R_n| \leq (4/9)\Delta$,

$$\mathbb{P}\Big(2\sum_{i=1}^n Z_i \geq \frac{8}{9}\Delta - R_n \Big) \leq \mathbb{P}\Big(2\sum_{i=1}^n Z_i \geq \frac{4}{9}\Delta \Big) = \mathbb{P}\Big(\sum_{i=1}^n Z_i \geq \frac{2}{9}\Delta \Big) \leq \frac{\delta}{2}$$

by Lemma 2.5 and the assumptions on a_n, finishing the proof of the first inequality. □

3 The multi-armed bandit problem

This section discusses the classical Multi-armed bandit problem, and how to solve it. This is useful to us, as the bandit problem is an efficient way to guide the search when optimizing expensive black-box functions. We also introduce the more recent Multi-fidelity multi-armed bandit problem of Kandasamy et al. [12], and its solution. We will then use these problems in the next section to model node selection in Monte Carlo tree search as bandit problems, leading to UCT and MF-UCT.

A player of slot machines in a casino wishes to identify the machine (or machines) with the highest expected payout. The means available to make this identification is to play different machines and observe the resulting rewards. The player is faced with the perhaps simplest form of the so-called exploration exploitation dilemma. In the face of sequential decision-making under uncertainty, one wishes to balance between the opposing needs of exploring untried actions in order to identify good choices, and taking the empirically best (so far) action as often as possible. For our casino player, this means that she wishes to simultaneously play new machines, and play the one that has so far yielded the highest mean payout. This is known as the *multi-armed bandit problem*, or simply bandit problem. The term "bandit" comes from the American slang for slot machine; "one-armed bandit", because in the long run, it is as efficient as a human bandit in separating the victim from their money. In the casino, our player faces many bandits at once, a "multi-armed bandit", which presents the sequential decision-making of which machine to play next. Introduced in 1933 by Thopmson [18], the original motivation was clinical trials when different treatments exist for the same disease. In this situation, the doctor faces the exploration exploitation dilemma in trying to identify the best available treatment while only being able to observe the outcomes on previous patients. Modern technology provides many new applications, and bandit problems are now important in many different domains. To give a few examples, ad placement is the problem of choosing which advertisement to display for the next user. This could be different ads or different designs of the same ad. Another example is hyperparameter fitting in machine learning models, where the Hyperband algorithm [19] outperforms many popular Bayesian methods. This section gives a brief introduction to the bandit problem and some methods to solve it.

There are three fundamental formulations of the bandit problem that depend on the nature of the reward process: Stochastic, adversarial and Markovian. We focus here on the stochastic bandit problem, and in particular the well known *upper confidence bound* (UCB) strategies introduced by Auer, Peter and Cesa-Bianchi in 2002 [20].

The stochastic K-armed bandit problem consists of $K \geq 2$ sequences of unknown rewards $X_{i,1}, X_{i,2}, \ldots$, where i is the index of an arm of the K-armed bandit. The set of arms is denoted $\mathcal{K} = \{1, \ldots, K\}$. Following the seminal work of Robbins [21] in 1952, we associate with each arm $i \in \mathcal{K}$ an unknown probability distribution θ_i on the interval $[0, 1]$. An algorithm $\mathcal{A}$ that chooses, for each time $t = 1, 2, \ldots$ the next arm I_t to play and observe the associated reward $X_{I_t,t}$ is called an *allocation strategy*, or just strategy.

Rather than trying to maximize the expected reward, a strategy $\mathcal{A}$ attempts to minimize the *regret*, defined as the loss due to not always playing the optimal arm, or arms since there could be more than one. In the stochastic setting, it is more natural to speak of the *pseudo-regret*, the *expected* loss due to not always playing the optimal arm. We will only be concerned with the pseudo-regret in this work, so for convenience we abbreviate it to just regret. With this in mind, the regret of a strategy $\mathcal{A}$ after n plays, $I_1, \ldots, I_n$ is defined to be

$$R_n(\mathcal{A}) = \max_{i \in \mathcal{K}} \mathbb{E}\left[\sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t}\right] = n\mu^* - \sum_{t=1}^{n} \mathbb{E}[\mu_{I_t}]. \tag{3.1}$$

From now on, any quantity marked by a "$*$" refers to an optimal arm $i^* \in \operatorname{argmax}_{i \in \mathcal{K}} \mu_i$. Letting $T_i(n)$ be the number of times the arm i has been played by $\mathcal{A}$ during the first n time steps, and defining $\Delta_i = \mu^* - \mu_i$, we may rewrite the regret (3.1) to the more compact form

$$R_n(\mathcal{A}) = \sum_{i=1}^{K} \Delta_i \mathbb{E}[T_i(n)]. \tag{3.2}$$

When the strategy is clear from context, we simply write R_n for the regret.

For a large family of reward distributions (see Appendix A), there is a well known lower bound on the regret, introduced by Lai and Robbins in their classical 1985 paper [22]. Specifically, they found strategies that satisfy

$$\mathbb{E}[T_i(n)] \leq \left(\frac{1}{I(p_i \| p^*)} + o(1)\right) \ln n \tag{3.3}$$

where $o(1) \to 0$ as $n \to \infty$, and

$$I(p_i \| p^*) = \int_{-\infty}^{\infty} p_i(x) \ln \frac{p_i(x)}{p^*(x)} d\nu(x) \tag{3.4}$$

is the Kullback-Leibler divergence between p_i and p^*, the respective probability density functions of θ_i and θ^* with respect to some measure ν. They also proved that the resulting regret is the best possible for any strategy, given some mild assumptions on the reward distributions θ_i. Specifically, for any suboptimal arm i, $\mathbb{E}[T_i] \geq (\ln n)/I(p_i \| p^*)$. The interested reader is referred to Appendix A for the precise statement and proof of this fundamental theorem. In light of this result, any strategy that has a regret that grows within a constant factor of this best regret, is said to resolve the exploration exploitation dilemma. The Kullback-Leibler divergence is commonly used to measure the distance between two probability distributions, so the bound of Lai and Robbins says that it is difficult to distinguish suboptimal arms whose distribution is "close" to that of an optimal arm. Moreover, it is not possible to entirely exclude even a very bad arm, as its expected number of plays grows at least as $\ln n$.

The strategies of Lai and Robbins computes for each arm a quantity called the *upper confidence index*, and chooses an arm with the highest index at each time step. This index requires the entire sequence of rewards obtained so far, which makes it hard to compute in general. A really attractive strategy would need to have a regret that solves the exploration exploitation dilemma, while being easy to compute.

3.1 Dealing with uncertainty

Any successful strategy must, in order to solve the exploration exploitation dilemma, in a way both explore and exploit at the same time. A simple heuristic principle for achieving this paradoxical behavior is the *optimism in face of uncertainty.* This is a general principle that can be used for sequential decision-making in an uncertain environment. We assume that a strategy has some data on the environment, and based on this it must choose an action to take. Then the strategy constructs a set of "plausible" environments which are "consistent" with the data and identifies the most "favorable" environment in this set. As the name suggests, the heuristic then says that we should choose the action which is optimal in this most favorable of the plausible environments. In short, "construct the best of all plausible worlds, and take the optimal action in this". Optimism in face of uncertainty gives simple and yet almost optimal strategies for the stochastic multi-armed bandit problem.

A family of strategies that rely on optimism in face of uncertainty is the UCB strategies. The most widely used of these is the UCB1 strategy that was introduced in 2002 by Auer, Cesa-Bianchi and Fischer [20]. We present here a more general version called (α, ψ)-UCB, presented by Bubeck and Cesa-Bianchi in [13].

3.2 Upper confidence bound

Throughout this thesis we make the assumption that the distribution of rewards X satisfy the following conditions:

Assumption 3.1. *There exists a convex function ψ on $\mathbb{R}$ such that for all $\lambda \geq 0$*

$$\ln\left(\mathbb{E}\left[e^{\lambda(X-\mathbb{E}[X])}\right]\right) \leq \psi(\lambda) \quad \textit{and} \quad \ln\left(\mathbb{E}\left[e^{-\lambda(X-\mathbb{E}[X])}\right]\right) \leq \psi(\lambda) \tag{3.5}$$

Making no other assumption on the distributions than $X \in [0,1]$, Hoeffding's lemma implies (3.5) with $\psi(\lambda) = \frac{\lambda^2}{8}$.

Following the optimism in face of uncertainty heuristic, we use (3.5) to construct an upper bound estimate on the mean of each arm at some fixed confidence level. The arm with the highest estimate is optimal in the best of plausible worlds, and is chosen to play next.

Let $\overline{X}_{i,s}$ be the mean value of rewards that result from playing s times at arm i. Since the rewards are i.i.d, we have that $\overline{X}_{i,s} = \frac{1}{s}\sum_{t=1}^{s} X_{i,t}$ in distribution. Using Chernoff's Bounding Method we derive concentration inequalities, (3.7) which are then used to construct our upper confidence bound. By Corollary 2.1,

$$\begin{aligned}\mathbb{P}(s\mu_i - s\overline{X}_{i,s} \geq s\varepsilon) &\leq \inf_{\lambda>0} e^{-s\varepsilon\lambda}\mathbb{E}[e^{-\lambda(s\overline{X}_{i,s}-s\mu_i)}] \\ &= \inf_{\lambda>0} e^{-s\varepsilon\lambda}\prod_{j=1}^{s}\mathbb{E}\left[e^{-\lambda(X_{i,s}-\mathbb{E}[X_{i,s}])}\right].\end{aligned}$$

Now, applying Assumption 3.1 we have

$$\mathbb{P}(s\mu_i - s\overline{X}_{i,s} \geq s\varepsilon) \leq \inf_{\lambda>0} e^{-s\varepsilon\lambda+s\psi(\lambda)} = e^{-s\psi^*(\varepsilon)}$$

where ψ^* is the Legendre-Fenchel (LF) transform of the convex function ψ in the assumption. The LF transform of ψ is defined as

$$\psi^*(\varepsilon) = \sup_{\lambda\in\mathbb{R}}(\lambda\varepsilon - \psi(\lambda)).$$

Note in particular that Assumption 3.1 implies that, $\psi(0) = 0$, which in turn means that

$$\psi^*(0) = \sup_{\lambda}(\lambda\cdot 0 - \psi(\lambda)) = \sup_{\lambda}(-\psi(\lambda)) = 0 \tag{3.6}$$

since ψ is convex and $\psi(0) = 0$. Moreover, the LF transform is always convex irrespective of the shape of the original function. The interested reader is referred to Appendix C for more details. Since ψ is convex by assumption, we are in the case when ψ^{**}, the LF transform of ψ^*, recover ψ. If in addition we assume that ψ is differentiable, the LF transform reduces to the Legendre transform commonly used in classical mechanics.

The above yields the concentration inequalities

$$\mathbb{P}(|\overline{X}_{i,s} - \mu_i| \geq \varepsilon) \leq e^{-s\psi^*(\varepsilon)} \tag{3.7}$$

where the second inequality is shown analogously. Hence, with probability at least $1-\delta$,

$$\overline{X}_{i,s} + (\psi^*)^{-1}\left(\frac{1}{s}\ln\frac{1}{\delta}\right) > \mu_i. \tag{3.8}$$

This is the upper confidence bound, or the measure of optimality in the best of plausible worlds of the optimism in face of uncertainty heuristic, and we have arrived at the (α,ψ)-UCB strategy where α is a parameter: At time t, play

$$I_t \in \operatorname{argmax}_{i\in\mathcal{K}}\left[\overline{X}_{i,T_i(t-1)} + (\psi^*)^{-1}\left(\frac{\alpha\ln t}{T_i(t-1)}\right)\right]. \tag{3.9}$$

3.3 The UCB algorithm

The (α, ψ)-UCB strategy is appealing because in contrast to the upper confidence index of Lai and Robbins, the upper confidence bound is easy to compute. It consists of two terms: $\overline{X}_{i,T_i(t-1)}$ is the average reward from arm i which biases exploitation, and $(\psi^*)^{-1}(\alpha \ln t / T_i(t-1))$ which biases selecting an arm that has not been played as often.Throughout this work we shall refer to this second term as the *exploration term*, and sometimes we write $c_{t,s} = (\psi^*)^{-1}(\alpha \ln t/s)$ for short. The strategy is summarized below in Algorithm 1.

Algorithm 1 (α, ψ)-UCB

```
**Initialization:** Play each arm once
**for** t = 1,2,... **do**
    Select $I_t \in \operatorname{argmax}_{i \in \mathcal{K}} \left[ \overline{X}_{i,T_i(t-1)} + (\psi^*)^{-1} \left( \frac{\alpha \ln t}{T_i(t-1)} \right) \right]$
```

Moreover, as the next theorem states, (α, ψ)-UCB resolves the exploration exploitation dilemma.

Theorem 3.1 (Regret of (α, ψ)-UCB)**.** *Assume that the reward distributions satisfy* (3.5)*. Then (α, ψ)-UCB with $\alpha > 2$ satisfies*

$$R_n \le \sum_{i:\Delta_i > 0} \Delta_i \left(\frac{\alpha \ln n}{\psi^*(\Delta_i/2)} + \frac{\alpha}{\alpha - 2} \right).$$

Assuming only that $X \in [0, 1]$ we note that taking $\psi(\lambda) = \frac{\lambda^2}{8}$ in (3.5), gives $\psi^*(\varepsilon) = 2\varepsilon^2$, which yields the regret bound

$$R_n \le \sum_{i:\Delta_i > 0} \left(\frac{2\alpha}{\Delta_i} \ln n + \frac{\alpha}{\alpha - 2} \Delta_i \right). \tag{3.10}$$

This is an important special case of (α, ψ)-UCB that is referred to simply as α-UCB. The original UCB1 algorithm introduced by Auer et al. is the case when $\alpha = 4$.

Proof of theorem 3.1. In order to bound $R_n = \sum_{i=1}^{K} \Delta_i \mathbb{E}[I_i(n)]$ it is enough to bound $\mathbb{E}[T_i(n)]$ for each suboptimal arm i. Any optimal arm i^* makes no contribution as $\Delta_{i^*} = 0$. Note that if (α, ψ)-UCB chooses arm $I_t = i \neq i^*$ at time t, then in particular

$$\overline{X}_{i^*,T_{i^*}(t-1)} + (\psi^*)^{-1} \left(\frac{\alpha \ln t}{T_{i^*}(t-1)} \right) < \overline{X}_{i,T_i(t-1)} + (\psi^*)^{-1} \left(\frac{\alpha \ln t}{T_i(t-1)} \right),$$

which implies that at least one of the following must hold:

$$\overline{X}_{i^*,T_{i^*}(t-1)} + (\psi^*)^{-1}\left(\frac{\alpha \ln t}{T_{i^*}(t-1)}\right) \leq \mu^*, \tag{3.11}$$

$$\overline{X}_{i,T_i(t-1)} \geq \mu_i + (\psi^*)^{-1}\left(\frac{\alpha \ln t}{T_i(t-1)}\right), \tag{3.12}$$

$$T_i(t-1) < \frac{\alpha \ln t}{\psi^*(\Delta_i/2)} \tag{3.13}$$

Indeed, if all three equations were false we would have

$$\begin{aligned}
&\overline{X}_{i^*,T_{k^*}(t-1)} + (\psi^*)^{-1}\left(\frac{\alpha \ln t}{T_{i^*}(t-1)}\right) \\
&> \mu^* \\
&= \mu_i + \Delta_i \\
&\geq \mu_i + 2(\psi^*)^{-1}\left(\frac{\alpha \ln t}{T_i(t-1)}\right) \\
&> \overline{X}_{i,T_i(t-1)} + (\psi^*)^{-1}\left(\frac{\alpha \ln t}{T_i(t-1)}\right)
\end{aligned}$$

which in particular implies that $I_t \neq i$, as i^* has a higher upper confidence bound.

For

$$T_i(t-1) \geq u = \left\lceil \frac{\alpha \ln n}{\psi^*(\Delta_i/2)} \right\rceil,$$

equation (3.13) is false. Let $\{\Pi(x)\}$ be the indicator function of the event $\Pi(x)$ for any predicate Π, that is, $\{\Pi(x)\} = 1$ if $\Pi(x)$ is true and 0 otherwise. Since the expected value of an event is just its probability of occurring, we get,

$$\begin{aligned}
\mathbb{E}[T_i(n)] &\leq u + \mathbb{E}\left[\sum_{t=u+1}^{n} \{I_t = i \text{ and } (3.13) \text{ is false}\}\right] \\
&\leq u + \mathbb{E}\left[\sum_{t=u+1}^{n} \{(3.11) \text{ or } (3.12) \text{ is true}\}\right] \\
&= u + \sum_{t=u+1}^{n} \Big(\mathbb{P}((3.11) \text{ or } (3.12) \text{ is true})\Big) \\
&\leq u + \sum_{t=u+1}^{n} \Big(\mathbb{P}((3.11) \text{ is true}) + \mathbb{P}((3.12) \text{ is true})\Big)
\end{aligned}$$

since the probability of at least one of two events occurring is bounded by the sum of their individual probabilities (union bound). It thus suffices to bound the probabilities of the events (3.11) and (3.12). Using (3.7) and the union bound again we see that

$$
\begin{aligned}
\mathbb{P}((3.11) \text{ is true}) &\leq \mathbb{P}\left(\exists s \in \{1, \dots, t\} : \overline{X}_{i^*,s} + (\psi^*)^{-1}\left(\frac{\alpha \ln t}{s}\right) \leq \mu^*\right) \\
&\leq \sum_{s=1}^{t} \mathbb{P}\left(\mu^* - \overline{X}_{i^*,s} \geq (\psi^*)^{-1}\left(\frac{\alpha \ln t}{s}\right)\right) \\
&\leq \sum_{s=1}^{t} e^{-s\psi^*\left((\psi^*)^{-1}\left(\frac{\alpha \ln t}{s}\right)\right)} \\
&= \sum_{s=1}^{t} e^{-s\frac{\alpha \ln t}{s}} = \sum_{s=1}^{t} e^{-\alpha \ln t} \\
&= \sum_{s=1}^{t} \frac{1}{t^{\alpha}} = \frac{1}{t^{\alpha-1}}.
\end{aligned}
$$

The same upper bound holds for (3.12) so we get that

$$
\mathbb{E}[T_i(n)] \leq u + \sum_{t=u+1}^{n} \frac{2}{t^{\alpha-1}} \leq u + 2\int_{u}^{\infty} \frac{ds}{s^{\alpha-1}} = u + \frac{2}{\alpha-2}u^{2-\alpha} \leq u + \frac{2}{\alpha-2}
$$

since $u > 1$ and $\alpha > 2$ by assumption. We now substitute this bound on expectations into the formula for the regret, giving

$$
\begin{aligned}
R_n = \sum_{i=1}^{K} \Delta_i \mathbb{E}[T_i(n)] &\leq \sum_{i:\Delta_i>0} \Delta_i \left(u + \frac{2}{\alpha-2}\right) \\
&\leq \sum_{i:\Delta_i>0} \Delta_i \left(\frac{\alpha}{\psi^*(\Delta_i/2)} \ln n + 1 + \frac{2}{\alpha-2}\right) \\
&= \sum_{i:\Delta_i>0} \left(\frac{\alpha \Delta_i}{\psi^*(\Delta_i/2)} \ln n + \frac{\alpha}{\alpha-2}\Delta_i\right),
\end{aligned}
$$

whence the theorem follows. □

As a final remark, we note that it is possible to relax the condition $\alpha > 2$. This can be done by replacing the union bound in the above proof by a “peeling” argument. This method lets one show logarithmic regret for $\alpha > 1$. See Bubeck [23] section 2.2 for details.

3.4 The multi-fidelity bandit problem

In their 2016 paper, Kandasamy et al. [12] introduced another upper confidence bound based algorithm to solve a variant of the classical stochastic K-armed bandit where observing the payoff of an arm is somehow costly, but we have access to one or more cheaper approximations of the payoffs. We call these approximations lower fidelity approximations, and they are used to excluding, at a lower cost, the worst arms. In doing so, we preserve capital for playing only a small subset of promising arms at higher fidelities. This will eventually result in the Multi-fidelity-UCB strategy (MF-UCB) which will be useful in antenna tilt optimization as computing the performance of the network given an antenna tilt configuration is expensive. Given access to one or more cheaper approximations of the network performance, we can treat the node selection in Monte Carlo tree search as a MF-bandit problem and thereby focus the computational resources on correctly estimating the most promising tilt configurations .

The multi-fidelity bandit problem differs from the classical bandit in that for each arm i we have access to $M-1$ approximate distributions $\theta_i^{(1)}, \theta_i^{(2)}, \ldots, \theta_i^{(M-1)}$ to the true distribution $\theta_i^{(M)} = \theta_i$. Of course, what we are trying to identify, is the arm (or arms) with the highest expectation $\mu^* = \max\{\mu_1^{(M)}, \ldots, \mu_K^{(M)}\}$. Thus, the lower fidelity approximations are useful to us exactly to the extent that they provide information about $\mu_i^{(M)}$ for each arm i. Driven by this observation, we make the assumption that for each fidelity $m = 1, \ldots, M-1$, the deviation in expectation $\mu_i^{(m)}$ from the true expectation $\mu_i^{(M)}$ is within some known quantity $\zeta^{(m)}$, and that this quantity decreases with m. That is,

$$|\mu_i^{(M)} - \mu_i^{(m)}| \leq \zeta^{(m)} \text{ for all } i \in \mathcal{K} \text{ and } m = 1, \ldots, M,$$

and

$$\zeta^{(1)} > \zeta^{(2)} > \cdots > \zeta^{(M)} = 0$$

are known. Note that in practice, it might be difficult to give tight bounds $\zeta^{(m)}$. The lower fidelities are only attractive if they are in some sense cheaper than the high fidelity M. To each fidelity m, we associate a cost $\lambda^{(m)}$ to playing an arm at fidelity m, and we have

$$\lambda^{(1)} < \lambda^{(2)} < \cdots < \lambda^{(M)}.$$

We extend our notation from the classical bandit problem to discuss the multi-fidelity bandit quite naturally. The number of plays at arm i at fidelity m after t time steps is denoted $T_{i,t}^{(m)}$, while $T_{i,t}^{(>m)}$ is the number of plays at the arm at all fidelities greater than m. The total number of plays at any arm at fidelity m is $Q_t^{(m)} = \sum_{i \in \mathcal{K}} T_{i,t}^{(m)}$. The mean of s samples drawn from $\theta_i^{(m)}$ is denoted $\overline{X}_{i,s}^{(m)}$, and $\Delta_i^{(m)} = \mu^* - \mu_i^{(m)} + \zeta^{(m)}$.

We aim towards an MF-UCB strategy to solve the multi-fidelity bandit problem. For this approach, we need the distributions $\theta_i^{(m)}$ to behave nicely. Kandasamy et al. make the following assumption:

Assumption 3.2. *All distributions satisfy concentration inequalities of the form*

$$\mathbb{P}(|\overline{X}_{i,s}^{(m)} - \mu_i^{(m)}| > \varepsilon) < \nu e^{-s\Phi(\varepsilon)} \tag{3.14}$$

for all $\varepsilon > 0$. Here $\nu > 0$ and Φ is an at least linearly increasing function with $\Phi(0) = 0$.

Note however that these bounds follow with ψ^* and $\nu = 1$ if all distributions satisfy Assumption 3.1. Therefore, we will assume that all distributions satisfy that there exists a convex function ψ so that (3.5) holds. To be explicit our assumption implies, for all i, m

$$\mathbb{P}(|\overline{X}_{i,s}^{(m)} - \mu_i^{(m)}| > \varepsilon) < e^{-s\psi^*(\varepsilon)}. \tag{3.15}$$

3.5 Multi-fidelity regret

A strategy $\mathcal{A}$ in the multi-fidelity setting chooses for each time t an arm $I_t \in \mathcal{K}$ to play, and a fidelity $m_t \in \{1, \dots, M\}$ at which to play the chosen arm. This results in observing a reward $X_{I_t,t}^{(m_t)}$. As with the classical bandit problem, a strategy attempts to minimize the regret (strictly speaking pseudo-regret), but we need to modify the notion of regret to fit the multi-fidelity setting. For a strategy $\mathcal{A}$ that is given Λ units of capital, that is we continue selecting a new arm as long as the total cost spent, $\sum_{t=1}^{s} \lambda^{(m_t)} < \Lambda$, we define the *instantaneous pseudo-reward* at time t as $q_t = \mu_{I_t}^{(M)}$, and let $r_t = \mu^* - q_t$ be the *instantaneous pseudo-regret.* Other choices for q_t are possible, but we stick with the one used by Kandasamy et al. We then define the regret of $\mathcal{A}$ to be

$$R(\Lambda, \mathcal{A}) = \Lambda\mu^* - \sum_{t=1}^{N} \lambda^{(m_t)} q_t = \underbrace{\left(\Lambda - \sum_{t=1}^{N} \lambda^{(m_t)}\right)\mu^*}_{\tilde{r}(\Lambda,\mathcal{A})} + \underbrace{\sum_{t=1}^{N} \lambda^{(m_t)} r_t}_{\tilde{R}(\Lambda,\mathcal{A})}. \tag{3.16}$$

where N is the random number of plays within capital Λ. As before, when the strategy is clear from context, we simply write $R(\Lambda)$ for the regret. In contrast to the classical bandit problem, the MF-regret depends on the total capital Λ, and not on the number n of plays so far. This is because N, the total number of rounds until the capital is spent, is a random variable in the MF-bandit. It will depend on which sequence of fidelities was chosen. A strategy that only plays at the highest fidelity would have a fixed number of plays within Λ capital, namely $N = \lfloor \Lambda / \lambda^{(M)} \rfloor$.

3.6 The multi-fidelity upper confidence bound (MF-UCB) strategy

As the name suggests, the MF-UCB strategy (or to be exact, the (α, ψ)-MF-UCB strategy), similar to (α, ψ)-UCB, uses an upper confidence bound for $\mu_i^{(m)}$ for each fidelity $m \in \{1, \dots, M\}$ and each arm $i \in \mathcal{K}$. The multi-fidelity upper confidence bounds are

$$\mathcal{B}_{i,t}^{(m)}(s) = \overline{X}_{i,s}^{(m)} + (\psi^*)^{-1}\big(\frac{\alpha \ln t}{s}\big) + \zeta^{(m)}, \quad \text{for all } m \in \{1, \dots, M\}, i \in \mathcal{K} \tag{3.17}$$

$$\mathcal{B}_{i,t} = \min_{m \in \{1,\dots,M\}} \mathcal{B}_{i,t}^{(m)}(T_{i,t-1}^{(m)}). \tag{3.18}$$

Similar to UCB, by (3.15) we have

$$\overline{X}_{i,s}^{(m)} + (\psi^*)^{-1}\left(\frac{1}{s}\ln\frac{1}{\delta}\right) > \mu_i^{(m)}. \tag{3.19}$$

with probability at least $1-\delta$, and since $\mu_i^{(m)} \geq \mu_i^{(M)} - \zeta^{(m)}$ we have that,

$$\overline{X}_{i,s}^{(m)} + (\psi^*)^{-1}\left(\frac{1}{s}\ln\frac{1}{\delta}\right) + \zeta^{(m)} > \mu_i^{(M)}.$$

with probability at least $1-\delta$. Hence, each $\mathcal{B}_{i,t}^{(m)}(T_{i,t-1}^{(m)})$ is a high probability upper bound on $\mu_i^{(M)}$. Kandasamy et al. show that the minimum, $\mathcal{B}_{i,t}$ gives the tightest bound. Similar to UCB, the MF-UCB strategy plays an arm that maximizes this bound, i.e. $I_t \in \text{argmax}_{i\in\mathcal{K}}\mathcal{B}_{i,t}$

Next, a fidelity is chosen. The condition $|\mu_i^{(M)} - \mu_i^{(m)}| \leq \zeta^{(m)}$ implies a constraint on the value of $\mu_i^{(M)}$:

$$\mu_i^{(m)} - \zeta^{(m)} \leq \mu_i^{(M)} \leq \mu_i^{(m)} + \zeta^{(m)},$$

and by (3.15), we have

$$\mathbb{P}\left(\overline{X}_{i,s}^{(m)} + (\psi^*)^{-1}\left(\frac{1}{s}\ln\frac{1}{\delta}\right) \geq \mu_i^{(m)}\right) < 1-\delta$$
$$\mathbb{P}\left(\overline{X}_{i,s}^{(m)} - (\psi^*)^{-1}\left(\frac{1}{s}\ln\frac{1}{\delta}\right) \leq \mu_i^{(m)}\right) < 1-\delta$$

so that $\mu_{I_t}^{(M)}$ lies in the interval

$$\left[\overline{X}_{I_t,T_{I_t,t-1}^{(m)}}^{(m)} - (\psi^*)^{-1}\left(\frac{\alpha \ln t}{T_{I_t,t-1}^{(m)}}\right) - \zeta^{(m)}, \overline{X}_{I_t,T_{I_t,t-1}^{(m)}}^{(m)} + (\psi^*)^{-1}\left(\frac{\alpha \ln t}{T_{I_t,t-1}^{(m)}}\right) + \zeta^{(m)}\right]$$

with high probability. Note that if $(\psi^*)^{-1}(\alpha \ln t / T_{I_t,t-1}^{(m)})$ is large, this interval that (highly probably) contains $\mu_{I_t}^{(M)}$ is large, implying that we could constrain the true mean more by playing at fidelity m. On the other hand, even exact knowledge of $\mu_{I_t}^{(m)}$ can only constrain $\mu_{I_t}^{(M)}$ to within a $\pm\zeta^{(m)}$ interval. So we would not want to play too many times at the mth fidelity. Because of this, we define for each low-fidelity $m = 1, \dots, M-1$ a threshold value $\gamma^{(m)}$ such that we choose the smallest fidelity m such that

$$(\psi^*)^{-1}(\alpha \ln t / T^{(m)}_{I_t,t-1}) \geq \gamma^{(m)}. \tag{3.20}$$

If no low-fidelity satisfies (3.20), we choose the high-fidelity M. The threshold values $\gamma^{(m)}$ are chosen by Kandasamy et al. as

$$\gamma^{(m)} = (\psi^*)^{-1}\bigg(\frac{\lambda^{(m)}}{\lambda^{(m+1)}}\psi^*(\zeta^{(m)})\bigg). \tag{3.21}$$

Their motivation for this choice is the following: Note that if $\Delta_i^{(m)} = \mu^* - \mu_i^{(m)} - \zeta^{(m)} > 0$ then we can say that arm i is not optimal. The second step of the algorithm attempts to ensure that arms with $\Delta_i^{(m)} \gtrsim \gamma^{(m)}$ are not played above fidelity m. If $\gamma^{(m)}$ is too large we would play too many suboptimal arms at high fidelities which is costly, whereas if it is too small we might play a suboptimal arm i (if $\mu_i^{(m)} > \mu^*$) too many times at fidelity m which may be less expensive but yields no useful information. The analysis in [12] reveal that, given our assumptions, (3.21) is an optimal tradeoff.

We have arrived at the MF-UCB algorithm, which is summarized in Algorithm 2.

Algorithm 2 (α, ψ)-MF-UCB

```
Initialization: Play each arm once
for t = 1,2,... do
    Select $I_t \in \text{argmax}_{i \in \mathcal{K}} \mathcal{B}_{k,t}$                                    ▷ See eq. (3.18).
    $m_t = \min_m\{m : (\psi^*)^{-1}(\alpha \ln t / T^{(m)}_{I_t,t-1}) \geq \gamma^{(m)} \vee m = M\}$   ▷ See eq. (3.21)
    Play $X \sim \theta^{(m_t)}_{I_t}$
```

Now, using (3.21), we see that (3.20) really says that if the previous play at arm T_t was at fidelity m, we play again at fidelity m if

$$\frac{\alpha \lambda^{(m+1)}}{\lambda^{(m)}\psi^*(\zeta^{(m)})} \ln t \geq T_{I_t,t-1}.$$

Equivalently, there is a constant C, depending on the costs $\lambda^{(m)}, \lambda^{(m+1)}$ and the bound on deviation in expectations $\zeta^{(m)}$ so that we switch fidelity from m, for arm I_t if

$$C \ln t < T_{I_t,t-1}.$$

We see that it is possible to go back and play at a lower fidelity if we have not played an arm for a long time, as $\ln t$ increases.

Finally, we make one last assumption on the fidelities in order to avoid a situation where fidelities are “too close” to each other, which could cause unnecessary fidelity switching at no significant information gain.

Assumption 3.3. *All $\zeta^{(m)}$ are such that, for all $m < M$,*

$$\sum_{j=1}^{m} \frac{1}{\psi^*(\zeta^{(j)})} \leq \frac{1}{\psi^*(\zeta^{(m+1)})}.$$

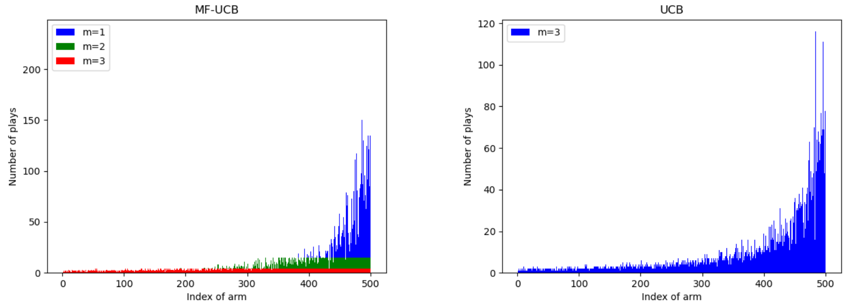
Note that we have assumed that $\zeta^{(m)}$, the upper bound of $|\mu^{(M)} - \mu^{(m)}|$ is known. This is not a realistic assumption for our purpose of mobile network optimization. We will rather try to use some method of function approximation to estimate the true network performance for a given antenna tilt configuration, which means that the best we can hope for is to estimate the deviation in means of the high- and low-fidelity scores by empirically computing the means on the training data. The network performance is computed by running an expensive simulation which can be configured with different number of users in the network. We also use a smaller number of users as a low-fidelity approximation, and estimate the deviation in means empirically. In their 2019 paper, Kandasamy et al. [24] discuss this problem and suggest a heuristic to adapt $\zeta^{(m)}$ which we describe in Section 5.

Simulation on synthetic problem

We compare UCB to MF-UCB on a simple synthetic problem to illustrate the difference. The problem is taken from Kandasamy et al. [12], and the setup is the following. We have $N = 500$ arms, and $M = 3$ fidelities. Let $\lambda = (\lambda^{(1)}, \lambda^{(2)}, \lambda^{(3)})$ and $\zeta = (\zeta^{(1)}, \zeta^{(2)}, \zeta^{(3)})$. The means of the high-fidelity distributions were chosen as a uniform grid in $(0, 1)$. All distributions are Gaussian with standard deviation $\sigma = 0.2$. The lower fidelity means were sampled uniformly within a $\pm\zeta^{(m)}$ band around $\mu_i^{(3)}$. The costs were chosen as $\lambda = (1, 10, 100)$ and we chose $\zeta = (0.2, 0.1, 0)$. Since the distributions are Gaussian, we choose $\psi^*(x) = \frac{x^2}{2\sigma^2}$. All parameters except for this choice of ψ^* were provided by Kandasamy et al. However, our result differ sharply from theirs, and the only way we could recreate a plot that looks somewhat like theirs was to invert the cost quotient in the computation of $\gamma^{(m)}$. We see this situation in Figure 2 which resembles the plot in [12]. The same experiment with non-inverted cost quotient is depicted in Figure 3. Figure 3c shows only the number of plays on each arm at the highest fidelity, since they are hidden in the plot that shows all fidelities. We note that MF-UCB manages to eliminate suboptimal arms at the lower fidelities, resulting in more plays at the near-optimal arms.

3.7 Analysis of MF-UCB

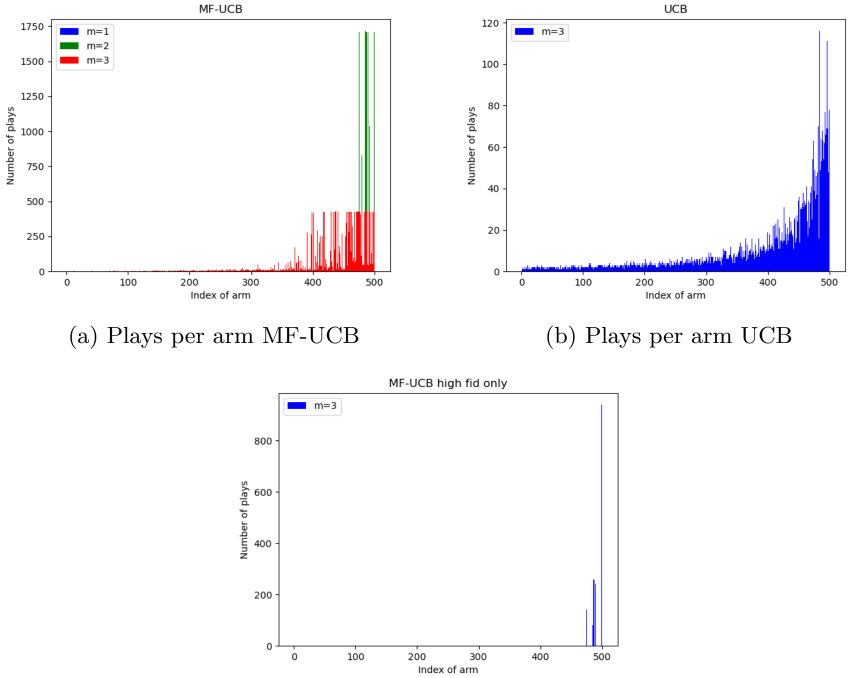
As for UCB, it is possible to give an upper bound on the expected regret of MF-UCB. The following discussion is lifted from Kandasamy et al. [12], and is included to give a flavor of the theoretical issues that the fidelities introduces.



(a) Plays per arm MF-UCB

(b) Plays per arm UCB

Figure 2: The simulation with inverted cost quotient.



(a) Plays per arm MF-UCB

(b) Plays per arm UCB

(c) Plays per arm on highest fidelity only

Figure 3: Multi-fidelity bandit problem with cost quotient not inverted.

In the following analysis, we will primarily consider the term

$$\tilde{R}(\Lambda, \mathcal{A}) = \tilde{R}(\Lambda) = \sum_{t=1}^{N} \lambda^{(m_t)} r_t$$

from (3.16). The other term, $\tilde{r}(\Lambda)$ is a residual term that reflects the fact that if we were to play the $N+1$:st play, we would exceed our capital Λ. We will use the following form for $\tilde{R}(\Lambda)$, as a sum over arms and fidelities:

$$\tilde{R}(\Lambda) = \sum_{i \in \mathcal{K}} \Delta_i^{(M)} \left(\sum_{m=1}^{M} \lambda^{(m)} T_{i,N}^{(m)} \right),$$

where, of course, $\Delta_i^{(m)} = \mu^* - \mu_i^{(m)}$ for $m = 1, \dots, M$.

Before we can move on to analyze the regret, we partition the set of arms $\mathcal{K}$ as follows: First we define the set of arms whose expectation at fidelity m, $\mu^{(m)}$ is within η of μ^* by

$$\mathcal{J}_\eta^{(m)} = \{ i \in \mathcal{K} : \Delta_i^{(m)} \leq \eta \}.$$

Next, define

$$\mathcal{K}^{(1)} = \overline{\mathcal{J}}_{\zeta^{(1)} + 2\gamma^{(1)}}^{(1)} = \{ i \in \mathcal{K} : \Delta_i^{(1)} > 2\gamma^{(1)} \}$$

to be the arms whose fidelity 1 expectation, $\mu^{(1)}$ is at least $\zeta^{(1)} + 2\gamma^{(1)}$ below the optimum μ^*. Then we recursively define

$$\mathcal{K}^{(m)} = \overline{\mathcal{J}}_{\zeta^{(m)} + 2\gamma^{(m)}}^{(m)} \cap \left(\bigcap_{\ell=1}^{m-1} \mathcal{J}_{\zeta^{(\ell)} + 2\gamma^{(\ell)}}^{(\ell)} \right), \ \forall m \leq M-1,$$

and finally

$$\mathcal{K}^{(M)} = \mathcal{K}^* \cap \left(\bigcap_{\ell=1}^{M-1} \mathcal{J}_{\zeta^{(\ell)} + 2\gamma^{(\ell)}}^{(\ell)} \right).$$

Then, for all $i \in \mathcal{K}^{(m)}$, we have $\Delta_i^{(m)} > 2\gamma^{(m)}$ and $\Delta_i^{(\ell)} > 2\gamma^{(\ell)}$ for all $i < m$. We will write $[\![i]\!]$ for the partition arm i belongs to, that is $[\![i]\!] = m$ means $i \in \mathcal{K}^{(m)}$. This partition of $\mathcal{K}$ results, as we shall see, in that $\mathcal{K}^{(m)}$ are the arms that are played at the mth fidelity but, based on the information available at fidelity m, can be excluded from plays at higher fidelities. An illustration of the partitions is given in Figure 4.

We now move on to give a regret bound for MF-UCB. In what follows we will follow Kandasamy et al. and use $\asymp, \gtrsim, \lesssim$ to denote equality and inequalities ignoring constants. The total number of plays within the capital Λ, which is a random integer, is denoted by N. Moreover, we let $n_\Lambda = \lfloor \Lambda / \lambda^{(M)} \rfloor$ be the deterministic number of plays that would have resulted if we only played at the highest fidelity. Because the costs increase with m, the total number of plays N might well be very large if cheap low-fidelities are available. We will however show that for MF-UCB, $N \lesssim n_\Lambda$ with high probability.

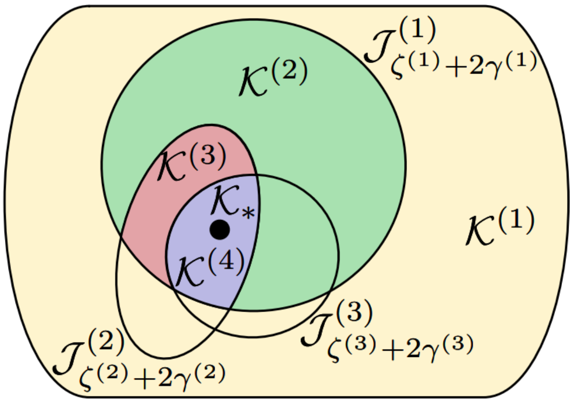

$\mathcal{J}^{(1)}_{\zeta^{(1)}+2\gamma^{(1)}}$
$\mathcal{K}^{(2)}$
$\mathcal{K}^{(3)}$
$\mathcal{K}_*$
$\mathcal{K}^{(1)}$
$\mathcal{K}^{(4)}$
$\mathcal{J}^{(3)}_{\zeta^{(3)}+2\gamma^{(3)}}$
$\mathcal{J}^{(2)}_{\zeta^{(2)}+2\gamma^{(2)}}$

Figure 4: Illustration of the partition into $\mathcal{K}^{(m)}$'s for $M = 4$. The sets $\mathcal{J}^{(m)}_{\zeta^{(m)}+2\gamma^{(m)}}$ are indicated next to their boundaries. $\mathcal{K}^{(1)}, \mathcal{K}^{(2)}, \mathcal{K}^{(3)}$ and $\mathcal{K}^{(4)}$ are shown in yellow, green, red, and purple respectively. The optimal arms K^* are shown as a black circle. (The image is lifted from Kandasamy et. al. [12].)

Theorem 3.2. *Let $\alpha > 4$. There exists Λ_0, depending on the $\lambda^{(m)}$'s such that for all $\Lambda > \Lambda_0$, the pseudo-regret of MF-UCB satisfies*

$$\frac{\mathbb{E}[R(\Lambda)]}{\ln n_\Lambda} \lesssim \sum_{i \notin \mathcal{K}^*} \Delta_i^{(M)} \frac{\lambda^{(\llbracket i \rrbracket)}}{\psi(\Delta_i^{(\llbracket i \rrbracket)})} \asymp \sum_{m=1}^{M} \sum_{k \in \mathcal{K}^{(m)}} \Delta_i^{(M)} \frac{\lambda^{(m)}}{\psi(\Delta_i^{(m)})}$$

We discuss briefly the proof of Theorem 3.2. For a full proof, see Appendix A of [12]. First, we control the number of plays at an arm at the various fidelities, depending on which $\mathcal{K}^m$ the arm belongs to.

Lemma 3.1. *Assume that $\alpha > 2$ and $\nu > 0$. Let $m \leq M$ and consider any arm $i \in \mathcal{K}^{(m)}$. After n steps of MF-UCB, we have the following bounds on $\mathbb{E}[T^{(\ell)}_{i,n}]$ for $\ell = 1, \dots, M$.*

$$T^{(\ell)}_{i,n} \leq \frac{\alpha \ln n}{\psi^*(\gamma^{(m)})} + 1, \forall \ell < m, \quad \mathbb{E}[T^{(m)}_{i,n}] \leq \frac{\alpha \ln n}{\psi^*(\Delta_i^{(m)}/2)} + \kappa_\alpha \quad \mathbb{E}[T^{(>m)}_{i,n}] \leq \kappa_\alpha$$

where $\kappa_\alpha = 1 + \frac{\nu}{2} + \frac{M\nu}{\alpha - 2}$ is a constant.

The first inequality follows by the design of MF-UCB, as it does not play any arm more than $\lfloor \frac{\alpha \ln t}{\psi^*(\gamma^{(m)})} \rfloor + 1$ times at fidelity $m < M$. The second and third are shown in a manner similar to Theorem 3.1, the classical bandit problem, with some additional details due to the different fidelities.

Denoting the regret that results from plays at arm i by

$$\tilde{R}_i(\Lambda) = \sum_{m=1}^{M} \lambda^{(m)} \Delta_i^{(M)} T_{i,N}^{(m)}$$

we can control the conditional expectation (recall that N, the total number of plays is a random variable)

$$\tilde{R}_{i,n} = \mathbb{E}[\tilde{R}_i(\Lambda)|N = n].$$

The lemma tells us that

$$\frac{\tilde{R}_{i,n}}{\Delta_i^{(M)} \ln n} \lesssim \sum_{\ell=1}^{[\![i]\!]-1} \frac{\lambda^{(\ell)}}{\psi^*(\gamma^{(m)})} + \frac{\lambda^{([\![i]\!])}}{\psi^*(\Delta_i^{([\![i]\!])})} + o(1) \tag{3.22}$$

Next we need to control N, the number of plays within the specified capital. To this end, we begin by the following high-probability bounds of $T_{i,n}^{(m)}$.

Lemma 3.2. *For any arm $i \in \mathcal{K}^{(m)}$, with $\alpha > 2, \gamma^{(m)} > 0$, the following concentration inequalities for $\ell = 1, \ldots, M$ and for any $x \geq 1$:*

$$\mathbb{P}\left(T_{i,n}^{(m)} > x\left(1 + \frac{\alpha \ln n}{\psi^*(\Delta_i^{(m)}/2)}\right)\right) \leq \frac{\nu \tilde{\kappa}_{i,\alpha}^{(m)}}{(x \ln n)^{\alpha-1}} + \frac{\nu}{n^{x\alpha-1}}$$

$$\mathbb{P}(T_{i,n}^{(>m)} > x) \leq \frac{M\nu}{\alpha - 1}\frac{1}{x^{\alpha-1}} + \frac{1}{(\alpha-2)x^{\alpha-2}}$$

where

$$\tilde{\kappa}_{i,\alpha}^{(m)} = \frac{M}{\alpha - 1}\left(\frac{\psi^*(\Delta_i^{(m)}/2)}{\rho}\right)^{\rho-1}.$$

With Lemma 3.2 we bound the number of plays at fidelities less than M to obtain that

$$\frac{n}{2} > \sum_{m=1}^{M-1} Q_n^{(m)}$$

high probability, say greater than δ, for all $n \geq n_0$. Setting $\delta = 1/\ln(\Lambda/\lambda^{(1)})$ yields $\mathbb{E}[\ln N] \lesssim \mathbb{E}[\ln n_\Lambda]$ but the argument is a bit delicate as δ depends on Λ. This gives us a bound for the regret caused by ark i of the form 3.22, but with n_Λ in place of n. Then, using the design of the sets $\mathcal{K}^{(m)}$ and Assumption 3.3, we can argue that the regret caused by plays at arm i at fidelities lower than $[\![k]\!]$ is dominated by $\lambda^{([\![k]\!])}/\psi^*(\Delta_i^{([\![k]\!])})$.

We formulate one last Lemma:

Lemma 3.3. *Let $\alpha > 4$. There is a number Λ_0 that depends on $\lambda^{(1)}$ and $\lambda^{(M)}$, such that for all $\Lambda > \Lambda_0$, we have*

$$\mathbb{E}[R(\Lambda)] \leq \mu^* \lambda^{(M)} + \sum_{i=1}^{K} \Delta_i^{(M)} \Bigg(\sum_{\ell=1}^{[\![i]\!]-1} \lambda^{(\ell)} \frac{\alpha(\ln(2n_\Lambda)+1)}{\psi^*(\gamma^{(\ell)})} + \lambda^{([\![i]\!])} \frac{\alpha(\ln(2n_\Lambda)+1)}{\psi^*(\Delta_i^{([\![i]\!])}/2)} + \mu^*(1+\tfrac{\nu(1+M)}{\alpha-2})\lambda^{(M)} \Bigg).$$

Theorem 3.2 now follows by plugging $\psi^*(\gamma^{(m)}) = \frac{\lambda^{(m)}}{\lambda^{(m+1)}} \psi^*(\zeta^{(m)})$ into Lemma 3.3 and using Assumption 3.3.

The assumption $\alpha > 4$ is here needed to ensure that $\sum_{m=1}^{M-1}$ stays sublinear when we apply the probabilities from Lemma 3.2. According to Kandasamy et al. it is possible to relax this to $\alpha > 2$ by a more careful design of the partitions $\mathcal{K}^{(m)}$. The lower bound Λ_0 on the budget arises because MF-UCB plays at low fidelities in the initial phase which for small budgets can result in N much larger than n_Λ, which we remind ourselves, is the number of plays that would have resulted from playing only at the highest fidelity from the start.

4 Monte Carlo Tree Search

Monte Carlo Tree Search (MCTS) is a family of algorithms developed mainly for game play, but in recent years other use cases such as discrete optimization have been considered. Of the method's successes, one may note that MCTS is used by alpha go to play the game Go at superhuman levels. MCTS has also been successfully used to play real time computer games such as Super Mario[1]. This section gives a brief overview of MCTS and introduces the UCT algorithm that forms the basis for all the methods we will consider for mobile network optimization. It builds on the UCB algorithm of the previous section. As a way to handle expensive black-box functions, we further suggest a new *multi-fidelity* UCT based on MF-UCB.

A game like Go can be visualized as a game tree, where we associate the starting position to the root node, and then all possible game states resulting from one legal move follow as the nodes at level 1. In the tree, the children of each node are precisely the possible game states that result from making a move from the current state. Evaluating a function of n variables can be viewed similarly as assigning a value to the first variable at the root node, and all possible values of the second variable are nodes at level 1 and so on until we reach level n. A path from the root to a leaf node then represents a full assignment.

Monte Carlo Tree Search builds a search tree by, loosely speaking, identifying the "most promising" legal move from the current position, and then simulating the rest of the game from the game state resulting from this most promising move. The outcome of the simulation is then used to update how "promising" the chosen move is in the future. Note that this is again the optimism in face of uncertainty heuristic, indicating that upper confidence bounds could be useful.

We stress that MCTS is a family of algorithms. However, they share a general approach, which we summarize in Algorithm 3 from the excellent survey of Monte Carlo Tree Search methods [25]. Each iteration consists of four steps:

1. *Selection*: Starting from the root node, the most appealing expandable node is found by recursively applying a child selection policy. A node is expandable if it is a non-terminal state that has unexplored children.
2. *Expansion:* A node is added as a child of the previously chosen node, according to the available actions.
3. *Simulation:* A simulation is run from the newly added node according to the default policy until an outcome can be observed.
4. *Backpropagation:* The observed outcome is "backed up" through the selected nodes to update their statistics.

A schematic overview of the procedure is found in Figure 5.

[1] This is highly entertaining to watch, if one has spent some playing the game oneself

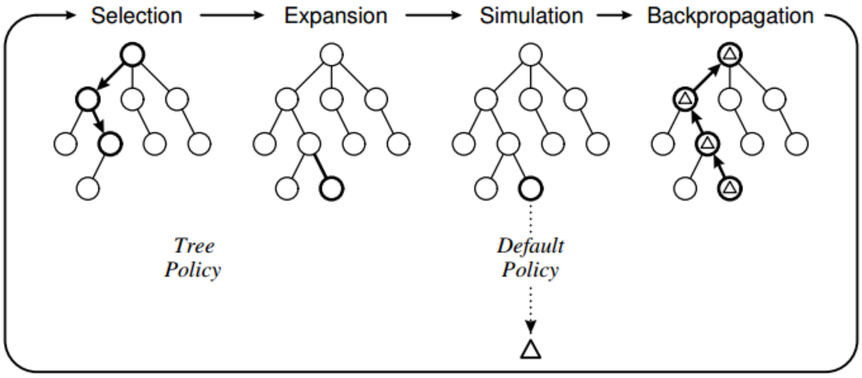


Figure 5: One iteration of the general MCTS approach from [25]

Algorithm 3 General MCTS approach

```
function MF-UCTSEARCH($s_0$)
    Create root node $v_0$ with state $s_0$
    while within computational budget do
        $v_l \leftarrow$ TREEPOLICY($v_0$)
        $\Delta \leftarrow$ DEFAULTPOLICY($s(v_l)$)
        BACKUP($v_l, \Delta$)
    return $a$(BESTCHILD($v_0$))
```

4.1 The upper confidence bound for trees (UCT) algorithm

The most basic MCTS is the upper confidence bound for trees (UCT) algorithm introduced by Kocsis, Szepesvári and Willemson in their seminal paper from 2002 [26].

In the UCT algorithm, to be defined below, we treat move selection as a separate multi-armed bandit problem for each internal node.

In this model, the arms of the bandit are the moves that are available from the current node, and the rewards are the result of games that passes through the node. The main difference from the bandit problem is that we allow the mean value of the payoffs X_i to drift with time, resulting in a *non-stationary bandit problem.* Indeed, the main assumption we need is that the expected values for the averages

$$\bar{X}_{i,n} = \frac{1}{n}\sum_{t=1}^{n} X_{i,t}$$

converge, which is used to ensure that we still have an exponential concentration inequality of the form (4.1) when we move down in the tree, and ultimately to ensure the convergence of UCT. We write $\mu_{i,n} := \mathbb{E}[\bar{X}_{i,n}]$ and

$$\mu_i = \lim_{n\to\infty} \mu_{i,n},$$

and we define $\delta_{i,n} = \mu_{i,n} - \mu_i$. As the name suggests, Kocsis et al. solve the resulting *non-stationary bandit problem* by applying UCB1 (that is 4-UCB in our previous terminology).

The UCT algorithm is summarized in Algorithm 4. This is the basic formulation from the survey [25] with the modification that it has been put in the (α, ψ) form of earlier sections.

4.2 Analysis of UCT

Here we state a series of theorems that allow us to prove the convergence of the (α, ψ)-UCT algorithm, which is a generalized version of the UCT that was introduced by Kocsis and Szepesvári. Their version uses 4-UCB to select nodes, while ours uses (α, ψ)-UCB, allowing for more general probability distributions. To the best of our knowledge, this is a new generalization of UCT. The proofs of these results are placed in Appendix B to improve the readability of this section. We start by analyzing (α, ψ)-UCB for a non-stationary bandit problem. Following the analysis of the non-stationary bandit problem, we prove the convergence and consistency of the UCT algorithm. Following Kocsic et al. we collect our assumptions on the rewards in the non-stationary bandit problem here:

Assumption 4.1. *Fix an arm $1 \leq i \leq \mathcal{K}$. We assume that $X_{i,t} \in [0,1]$ and the limit of $\mu_{i,n} = \mathbb{E}[\overline{X}_{i,n}]$ exists. Let $\{\mathcal{F}_{i,t}\}_t$ be a filtration such that $\{X_{i,t}\}_t$ is*

$\{\mathcal{F}_{i,t}\}$-adapted, and $X_{i,t}$ is conditionally independent of $\mathcal{F}_{i,t+1}, \mathcal{F}_{i,t+2}, \ldots$ given $\mathcal{F}_{i,t-1}$. Moreover, we assume that there exist a constant $C_p > 0$, and a positive integer N_p such that for $n \geq N_p$, for any $\delta > 0$, the following bounds hold:

$$\mathbb{P}\bigg(n\big|\overline{X}_{i,n} - \mathbb{E}[\overline{X}_{i,n}]\big| \geq \Delta_n(\delta)\bigg) \leq \delta. \tag{4.1}$$

That $X_{i,t}$ is an adapted process means essentially that we can not use the sequence of rewards obtained so far to gain information on the next reward. Note that letting $\delta = t^{-\alpha}$ in (4.1), we get

$$\mathbb{P}\bigg(\big|\overline{X}_{i,n} - \mu_{i,n}\big| \geq C_p(\psi^*)^{-1}\bigg(\frac{\alpha \ln t}{n}\bigg)\bigg) \leq t^{-\alpha}.$$

Thus, choosing

$$c_{t,s} = C_p(\psi^*)^{-1}\bigg(\frac{\alpha \ln t}{n}\bigg) \tag{4.2}$$

for the exploration term, we have

$$\mathbb{P}\bigg(\big|\overline{X}_{i,n} - \mu_{i,n}\big| \geq c_{t,n}\bigg) \leq t^{-\alpha}. \tag{4.3}$$

This bound is the non-stationary version of (3.7) that was used in the proof of Theorem 3.1, so Assumption 4.1 is that if we run (α, ψ)-UCB on the non-stationary bandit problem for long enough, we can choose the bias term (4.2) to ensure that we still have a concentration inequality of the same kind as (3.7).

In what follows, we give an upper bound on the expected regret in Theorem 4.1. Next, Theorem 4.2 bounds the difference of μ^* and the total payoff received so far. An exponential tail inequality for the estimated payoff Theorem is shown in 4.3. Next, we prove in Theorem 4.4 that the probability of failure vanishes with time. Based on these results, we prove the main result of this section, namely the consistency of the UCT algorithm. The proofs of all theorems are located in Appendix B. To the best of our knowledge, these generalizations to the (α, ψ) setting, of the results of Kocsis et al. in [26] have not been published before.

We let $\Delta_i = \mu^* - \mu_i$. Since $\delta_{i,t}$ converges to zero by assumption, there is, for all $\varepsilon > 0$ some index $N_0(\varepsilon)$ such that if $t \geq N_0(\varepsilon)$, we have $|\delta_{i,t}| \leq \varepsilon\Delta_i/2$ and $|\delta_{i^*,t}| \leq \varepsilon\Delta_i/2$ whenever i is a suboptimal arm, and i^* is an optimal arm. In particular, it follows that for any optimal arm i^* we have for $t \geq N_0(\varepsilon)$ that

$$|\delta_{i^*,t}| \leq \frac{\varepsilon}{2} \min_{i:\Delta_i > 0} \Delta_i.$$

Theorem 4.1. *Consider (α, ψ)-UCB applied to a non-stationary bandit problem where the payoff sequence satisfies Assumption 4.1 and the exploration term $c_{t,s}$*

Algorithm 4 UCT

```
function uctSearch($\Lambda$)
    $t \leftarrow 0$
    create root node $v_0$
    while within computational budget do
        $v_t \leftarrow$ TreePolicy($v_0$)
        $\Delta \leftarrow$ DefaultPolicy($v_t$)
        Backup($v_t, \Delta$)
        $t \leftarrow t + 1$
    return BestChild($v_0, 0$)

function TreePolicy($v$)
    while $v$ is not terminal do
        if $v$ not fully expanded then
            return Expand($v$)
        else
            $v \leftarrow$ BestChild($v, C$)
    return $v$

function Expand($v$)
    $a \leftarrow$ action from untried actions
    create child node with action $a$
    return child

function BestChild(v, C)
    $v_t \leftarrow \text{argmax}_{v' \in \{\text{children of } v\}} \overline{X}_{v',t-1} + C(\psi^*)^{-1} \left( \frac{\alpha \ln (T_{v,t-1})}{T_{v',t-1})} \right)$
    return $v_t$

function DefaultPolicy($v, m$)
    simulate until terminal state $s_T$ is reached
    $\Delta \leftarrow f(s_T)$
    return $\Delta$

function Backup($v, \Delta, m$)
    while $v$ is not None do
        $T_{v,t} \leftarrow T_{v,(t-1)} + 1$
        $X_{v,t} \leftarrow X_{v,(t-1)} + \Delta$
        $v \leftarrow$ parent of $v$
```

is given by (4.2). Fix $\varepsilon > 0$, and let $T_i(n)$ denote the number of plays at arm i after n time steps. Then, for $\alpha > 2$, if i is any suboptimal arm we have

$$\mathbb{E}[T_i(n)] \leq \left\lceil \frac{\alpha \ln n}{\psi^*\big(\frac{(1-\varepsilon)\Delta_i}{2C_p}\big)} \right\rceil + N_0(\varepsilon) + N_p + \frac{\alpha}{\alpha - 2}.$$

Theorem 4.2. *Let*

$$\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} T_i(n)\overline{X}_{i,T_i(n)}.$$

Then, under Assumption 4.1 we have

$$|\mathbb{E}[\overline{X}_n] - \mu^*| \leq |\delta_n^*| + \frac{2}{n}\sum_{i \neq i^*}\left(\left\lceil \frac{\alpha \ln n}{\psi^*\big(\frac{(1-\varepsilon)\Delta_i}{2C_p}\big)} \right\rceil + N_0(\varepsilon) + N_p + \frac{\alpha}{\alpha - 2}\right).$$

Theorem 4.3. *For $\delta > 0$, let $\Delta_n(\delta) = 9\sqrt{2n \ln(2/\delta)}$, and let*

$$\sqrt{n_0} \geq \frac{1}{\sqrt{2\ln 2}}\sum_{i=1}^{K}\left(\left\lceil \frac{\alpha \ln n_0}{\psi^*\big(\frac{(1-\varepsilon)\Delta_i}{2C_p}\big)} \right\rceil + N_0(\varepsilon) + N_p + \frac{\alpha}{\alpha - 2}\right).$$

Then for any $n \geq n_0$, we have the following bounds

$$\mathbb{P}(n\big|\overline{X}_n \geq -\mathbb{E}[\overline{X}_n]\big| \geq \Delta_n(\delta)) \leq \delta.$$

Theorem 4.4. *It holds that*

$$\lim_{t\to\infty} \mathbb{P}(I_t \neq i^*) = 0$$

Theorem 4.5. *Consider UCT running on a game tree of depth D and branching factor K, with stochastic payoffs at the leaves. Assume that the payoffs lie in $[0, 1]$, and satisfy Assumption 3.1. Then the bias, $|\mathbb{E}[\overline{X}_n] - \mu^*|$ of the estimated expected payoff $\overline{X}_n$ is $O((KD \ln n + K^D)/n)$. Moreover, the failure probability at the root converges to zero as the number of samples grow to infinity.*

Proof. The proof is done by induction on the depth D. When $D = 1$, (α, ψ)-UCT is just (α, ψ)-UCB. By Assumption 3.1, the payoffs satisfy (4.3) with $C_p = 1$. Hence, by Theorem 4.2 the bias is

$$|\mathbb{E}[\overline{X}_n] - \mu^*| = O\bigg(\frac{K(\ln n + N_0(\varepsilon))}{n}\bigg) = O\bigg(\frac{K \ln n + K}{n}\bigg),$$

and the failure probability converges to zero by Theorem 4.5.

Now assume that the result holds for all trees of depth up to $D-1$, and consider a tree of depth D. Then, from the root node, running (α, ψ)-UCT is equivalent to running (α, ψ)-UCB on a non-stationary bandit problem.

Fix an arm i. The payoff for move i from the root at time t will depend on all the previous entries into the subtree that has the successor node of arm i as root. Let us refer to this node as i as well for convenience. From the perspective of node i, $\overline{X}_n$ is the mean reward of all the plays from node i. From the perspective of the root, this is $\overline{X}_{i,n}$, the mean reward of all plays at arm i. The tree originating at node i has depth $D-1$ so the induction hypothesis hold, and we may apply Theorem 4.2. The exponential concentration of the payoffs hold by theorem 4.3, and this holds for any arm i, so it follows from Theorem 4.2 that the bias at the root satisfies

$$|\mathbb{E}[\overline{X}_n] - \mu^*| \leq |\delta_n^*| + O\bigg(\frac{K(\ln n + N_0(\varepsilon))}{n}\bigg). \tag{4.4}$$

where δ_n^* is the rate of convergence of the bias for the best move, and

$$N_0(\varepsilon) = \min\{n : |\delta_{i,n}| \leq \frac{\varepsilon\Delta_i}{2}, i \neq i^*\}.$$

Now, by the induction hypothesis,

$$|\delta_{i,n}| = O\bigg(\frac{K(D-1)\ln n + K^{D-1}}{n}\bigg),$$

for $i = 1, \dots, K$. Hence,

$$|\delta_{i,N_0(\varepsilon)}| = O\bigg(\frac{K(D-1)\ln N_0(\varepsilon) + K^{D-1}}{N_0(\varepsilon)}\bigg) \leq \frac{\varepsilon}{2}\Delta_i,$$

for $i = 1, \dots, K$, and we conclude that $N_0(\varepsilon) = O(K^{D-1})$, which together with (4.4) yields the desired result for the bias at the root. Note that the failure probability converges to zero by Theorem 4.4, completing the proof. □

4.3 The multi-fidelity UCT

The UCT algorithm is an MCTS algorithm that models child-node selection as a separate multi-armed bandit problem for each internal node. If computing the reward after each rollout is expensive, and we have access to cheaper approximations, it makes sense to model child-node selection as a multi-fidelity bandit problem instead. The resulting Multi-fidelity UCT (MF-UCT) is summarized in Algorithm 5. The main difference from UCT is that the function *BestChild* now uses MF-UCB to choose a node-fidelity pair. The functions that are left out of Algorithm 5 are identical to those of UCT, except that they take, and return the one additional argument; the fidelity that was chosen by *BestChild*. To the best of our knowledge, this is a new algorithm. Note that this version of MF-UCT that has not been modified for black-box optimization. We will discuss how to do this later in this section. As in UCT we have an exploration parameter C which modifies the upper confidence bound formula as follows:

$$\mathcal{B}_{i,t}^{(m)}(s,C) = \overline{X}_{i,s}^{(m)} + C(\psi^*)^{-1}\big(\frac{\alpha \ln t}{s}\big) + \zeta^{(m)}, \quad \text{for all } m \in \{1,\dots,M\}, i \in \mathcal{K} \tag{4.5}$$

$$\mathcal{B}_{i,t}(C) = \min_{m\in\{1,\dots,M\}} \mathcal{B}_{i,t}^{(m)}(T_{i,t-1}^{(m)}, C). \tag{4.6}$$

Algorithm 5 MF-UCT

```
function MF-UCTSEARCH($\Lambda$)
    $t \leftarrow 0$
    create root node $v_0$
    while used budget $< \Lambda$ do
        $v_t, m_t \leftarrow$ TREEPOLICY($v_0, 1$)
        $\Delta \leftarrow$ DEFAULTPOLICY($v_t, m_t$)
        BACKUP($v_t, \Delta, m_t$)
        $t \leftarrow t + 1$
    return BESTCHILD($v_0, 0$)

function BESTCHILD(v, C)
    $v_t \leftarrow \text{argmax}_{v\in\{\text{children of } v\}}\mathcal{B}_{v,t}(C)$                ▷ See eq. (4.6).
    $m_t \leftarrow \min_m\{m : (\psi^*)^{-1}(\frac{\alpha \log t}{T^{(m)}_{v_t,(t-1)}}) \geq \gamma^{(m)} \vee\ m = M\}$   ▷ See eq. (3.21)
    return $v_t, m_t$
```

One thing to note is that a typical node has a parent, and at some point this parent will request a higher fidelity rollout. In this situation, it makes no sense not to update the child node with this better information. Hence, as soon as all nodes at level 1 of the tree has switched to a higher fidelity, there will be no more rollouts at the first fidelity. In a setting where we have two fidelity levels, this means that when all level 1 nodes have requested high-fidelity rollouts, the rest of the search is essentially identical to UCT. The idea is then that most bad opening moves will have been eliminated by the low-fidelity search, which means that the high-fidelity search has a much smaller tree to traverse.

We do not prove the convergence of MF-UCT in the present work, tough we will see some experimental results indicating that it does converge. The main problem that arises in proving MF-UCT is similar to that of UCT, namely the non-stationary multi-fidelity bandit problem, in which the means are allowed to drift with time. A rigorous proof would have to establish regret bounds for MF-UCB, applied to the non-stationary multi-fidelity bandit problem, as well as concentration inequalities like those of Theorem 4.2, and convergence of the probability of failure.

4.4 Adapting Monte Carlo Tree Search for discrete black-box optimization

We now describe the adaptations that allow us to use UCT and MF-UCT for antenna tilt optimization, and indeed any discrete black-box optimization problem. As far as we are aware, these adaptations are new.

Let F be a real valued black-box function defined on some discrete subset X of $\mathbb{R}^n$. We attempt to maximize F over some finite subset G of X. Each variable corresponds to a level in the search tree, so we must impose an arbitrary order on the variables. A discussion on the impact of the antenna ordering in the case of antenna tilt optimization can be found in Section 5. Game states correspond to partial variable assignment, and setting a value for the next variable corresponds to a move. The rollout phase consists of uniformly at random choosing a feasible value until a full assignment x has been reached. This assignment is then evaluated on the black-box function, and the associated reward that is backed up through the tree is $F(x)$.

In its original formulation, UCT simulates games against an opponent, and returns the move that is currently believed to be the best next move from the current position. In our case there is no opponent, and rather than returning a single move (which would be the down tilt angle believed to be best for the first antenna to be configured in the case of antenna tilt optimization) we would like the algorithm to output the best configuration of all the antennas at once.

We adapt UCT for black-box optimization in two ways. First, in the rollout phase, we store each full assignment along with its associated reward. Once UCT has exhausted its computational budget, it returns the stored configuration with the highest reward. This has no effect on the preceding analysis.

Secondly, once we reach a terminal node (a full assignment) in the tree, we close that node as further visits will yield no additional information. We also recursively close nodes whose children are all closed. See Figure 6. This adaptation means that Theorem 4.4 is not necessarily true anymore, as the optimal arm from a given position may be closed. These adaptations are done in all the Monte Carlo Tree Search methods that we consider. In the case of MF-UCT, we also need to evaluate the assignment that results from reaching the terminal node at the highest fidelity before closing.

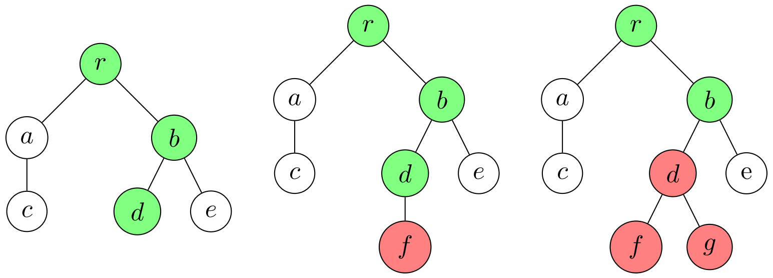


Figure 6: Small example to indicate when nodes are closed. Here we have three binary variables to configure, so each node has at most two children, and the tree has depth 4. We show only expanded nodes. In the left tree, node d at level 2 has been expanded, but it is not at the maximal depth. Hence, it is kept open. Node f at level 3 in the middle tree is terminal, as it is at the maximal depth. This means that (b, d, e) is a full assignment, i.e. can be evaluated on the black-box function under consideration. When it gets expanded, it is closed. However, node d has another (unexpanded) child, so it is kept open. In the right tree, node g is expanded and closed because it is at the maximal depth. Now all children of node d are closed, so node d is closed.

Any MCTS method attempts to identify the most promising moves and focus the search around them. In UCT, applying UCB to the bandit problem built into each node ensures that, in the long run, most rollouts will come from children with near-optimal expected values. However, we also know that no arm of the bandit is completely excluded, so eventually we will see rollouts even from arms that are initially deemed very bad. The resulting tree will therefore tend to be asymmetrical, with some branches going deep, and others very shallow. In applying UCT to black-box optimization, we therefore expect to see an asymmetric search tree. In fact, if we were to observe too much symmetry in the tree, we should have to conclude that our method needs modification.

5 Application to mobile network optimization

We want to optimize the performance of a mobile network that is made of several base stations, each one having multiple antennas. To each antenna we associate a cell as illustrated in Figure 7. If a user receives a strong enough signal from an antenna, it is considered to belong to the cell associated with the antenna. The signal strength depends on tunable parameters such as tilt angle and the power of each antenna, but also on the surrounding environment. Importantly, interference from other antennas affect the signal quality negatively. In the end, the quality of the user's experience depends on the signal strength.

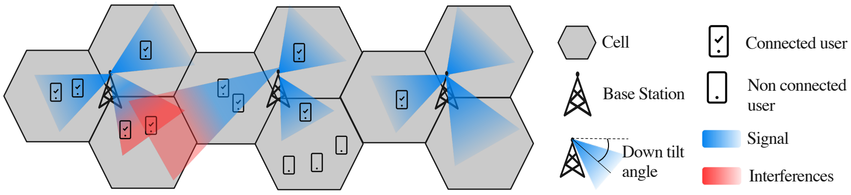


Figure 7: Illustration of a network configuration problem from Bouton et al. [9]. Each base station has three antennas. Two bad situations occur due to suboptimal tilt configuration: To the left, two antennas interfere with each other because the tilt angle is too small (signal beams upwards). The bottom middle cell has the opposite problem. A too large tilt angle causes users to lose coverage.

Figure 7 illustrates one possible configuration. The red color indicates interference where two antennas are configured with too low tilt angle, while the bottom middle cell has too high tilt angle, which causes users near the far edges of the cell to lose coverage.

In the present work, we attempt to optimize antenna tilt configurations. Each base station consists of three antennas. However, our approach is applicable to any network parameter optimization, and any number of antennas per base station. Even different number of antennas for different base stations is possible.

The measure of performance that we attempt to maximize is the average *signal to interference and noise ratio* (SINR) per user, that is we observe the SINR of each user in the network, and calculate the (arithmetic) mean value. The SINR is directly influenced by a change in antenna tilt and also captures the interference from other antennas. Note that the choice of averaging the SINR value of all users means that a configuration that provides excellent coverage to many users, but very poor coverage for a few, would still be considered as a fairly good one, even tough some users suffer from bad coverage. Other choices, such as the geometric mean value, could avoid this situation. Our methods can handle any choice of performance metric, but different choices could lead to very different results.

The antenna of each cell can set its electrical down tilt to one of sixteen values in the set $\{0°, \ldots, 15°\}$. The action space of each antenna is thus discrete, with sixteen actions. The size of the search space is 16^N where N is the number of antennas. In our experiments, we have seven base stations with three antennas each, resulting in a search space of $16^{21} \approx 1.9 \times 10^{25}$ tilt configurations.

We empirically test a number of variations of the basic UCT and the MF-UCT algorithms, with the adaptations for black-box optimization as described above, as well as a simple cross entropy method for comparison, using a high fidelity proprietary network simulator that handles multiple cells and antennas. The simulator performs all the path gain calculations using advanced radio propagation models, as well as traffic calculations for each user [27]. From that simulator we can get different key performance indicators for each user such as SINR and throughput. The mobile network in question consists of 7 base stations placed in a hexagonal configuration as shown in Figure 8. Each base station has three antennas for a total of 21 antenna tilt angles to configure. There are 1000 uniformly distributed users, and the environment is generated randomly so that half the area corresponds to an outdoor environment, and the rest is indoor.

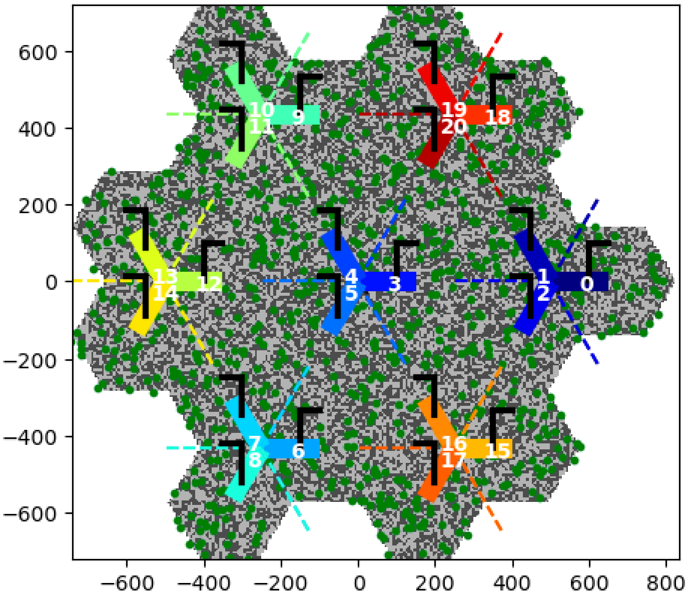


Figure 8: The simulation environment consists of 7 base stations with 3 antennas each, all set to down-tilt angle 0 degrees in the image. The cell of each antenna is indicated by the dotted lines. Half the area is randomly selected to be outdoors (dark gray), and the rest indoors (light gray). There are 1000 users uniformly distributed over the map, and these are represented by green dots.

5.1 Algorithms

UCT (uct): The basic UCT algorithm.

UCT-adaptive (uct_ad): As the search proceeds, it makes sense to explore less and less in favor of exploiting the fact that our estimates should become better with time. Moreover, given a good antenna ordering, the antennas that most influence the network performance should come early in the order. Therefore, it makes sense to explore these more. We capture this intuition by shrinking the exploration parameter as we go along. We choose to do this linearly so that the exploration parameter is $C(1 - \frac{b}{B})$, where B is the computational budget, and b is the budget used so far. Other shrinking rates are of course possible, and might be the subject of further study.

UCT-heuristic (uct_h): An alternative to random rollouts is to use some problem specific heuristic to simulate the rollouts. We use the heuristic "set every antenna to a fixed angle". The fixed angle in this case is 11°, and was deduced by keeping all antennas at the same angle, and sweeping through all possible angles. The angle resulting in the highest average SINR per user (see Figure 9) was chosen.

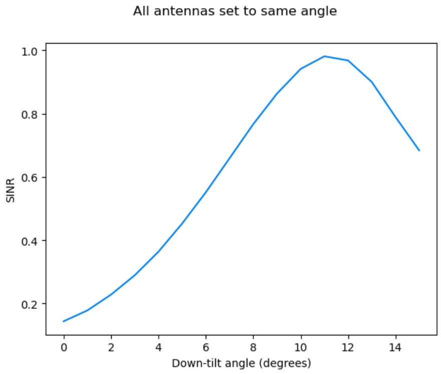


Figure 9: Plot of average SINR per user with all antennas set to the same angle.

Random tree (rnd): We also include as a baseline the variation that node choice is done uniformly at random instead of using the UCB approach.

MF-UCT Gaussian (mf_gp): Calculating the average SINR per user is computationally expensive. Each call to the simulator takes about two seconds. This makes our problem a candidate for our new MF-UCT approach. For low-fidelity approximation, we tried a few different machine learning methods, but Gaussian process seems to work best. We use the heuristic described in section 6 of Kandasamy et al. (2019) [24] for $\gamma^{(1)}$ as it is generally difficult to accurately estimate $\zeta^{(1)}$. Instead of only doubling, we allow for different constants, so that our heuristic is "if no query has been made to the high-fidelity for more than

$\lambda^{(2)}/\lambda^{(1)}$ iterations, we multiply $\gamma^{(1)}$ by a constant". The initial value of $\gamma^{(1)}$ is calculated from the training data for the Gaussian process. We tried other learning algorithms to approximate the simulator, primarily decision trees, but the result was significantly worse.

MF-UCT Env (mf_env): We have also tried a version of MF-UCT that uses for low-fidelity approximation a version of the simulation environment with fewer users in the network, which we denote MF-UCT-env. The computation time is roughly linear in the number of users. We decrease the number of users from 1000 to 100 which gives $\lambda^{(1)} \approx 0.5$ and $\gamma^{(1)} \approx 0.02$. These approximations were computed by evaluating 50000 random assignments on both the high- and low-fidelity environment. Here we did not use the heuristic update of $\gamma^{(1)}$.

Cross entropy method (ce): We compare the MCTS methods to another black-box optimization approach, cross entropy. Our implementation is very basic, and there are many improvements possible. We have however taken into account that the cross entropy method has the potential to get stuck on local maxima, so we attempt to detect when this happens, and restart from another randomly chosen point. A number of iterations, and a step size, β is chosen. If the parameters of the distributions do not change by more than some fixed value ε between consecutive iterations, we interpret this as being stuck on a local maximum, and shift to a new, random position.

5.2 Antenna order

In all the MCTS methods, we must impose an arbitrary ordering of the antennas in which they are configured. This ordering could impact the rate of convergence since some antennas, or clusters of antennas, might have a greater impact on the network performance. Consider for example antenna 0 in Figure 8. It points away from antennas $10, 11, 13$ and 14, and will therefore not cause or be affected by interference with these. On the other hand, antennas 1 and 3 in the figure have to be configured together so that they optimally cover their cell without interference. Consider the extreme case when these antennas are first and last respectively in the antenna ordering. In order to evaluate the effect of their interference, an MCTS method would have to come up with a configuration of the entire network. On the other hand, if they were adjacent in the ordering, their interference would be apparent immediately. We wish to test the effect of antenna ordering on the performance of our algorithms. Following Bouton et al. [9] we represent our network as a graph based on the radiation pattern of the antennas, that is, antennas that can interfere with each other are connected by an edge. The resulting graph is shown in Figure 10. Using this information, we try to come up with good and bad antenna orderings to examine their effect on performance. The different orders are summarized in Table 1.

For a good ordering, we want antennas that interfere with each other to be clustered together. We suggest the following procedure to ensure that. Pick a

“g”:	12, 16, 20, 3, 1, 9, 4, 5, 6, 7, 11, 10, 2, 15, 17, 18, 19, 13, 14, 0, 8
“o”:	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
“s”:	1, 16, 12, 20, 4, 7, 9, 2, 6, 3, 5, 10, 11, 8, 19, 17, 18, 13, 15, 14, 0
“b”:	12, 0, 10, 20, 8, 17, 5, 13, 15, 19, 2, 9, 6, 3, 18, 11, 16, 4, 7, 1, 14

Table 1: The order in which the antennas are configured in the MCTS methods. The placement of the antennas can be found in Figure 10.

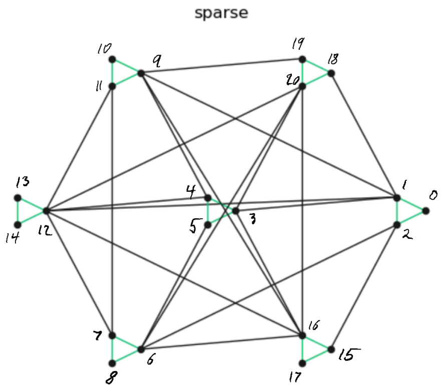


Figure 10: Our network graph representation. Each vertex corresponds to an antenna. An edge between antennas indicate that the antennas can interfere with each other.

vertex with maximum degree (in this case, the unique such vertex is 12 with degree 7). Next, choose a vertex of maximum degree among the vertices that are connected to the first. If there is no unique such vertex, choose one that has the most connections to the previously chosen vertices. Continue in this way until no new vertex can be chosen. Then restart with the highest degree among the remaining vertices. We label the resulting ordering with “g”. We can also use the graph to produce a (presumably) bad ordering. The idea is then to pick a starting vertex and then wait as long as possible to insert a vertex that is connected to the first. The order produced in this way is labelled “b”.

An alternative approach that does not rely on producing a graph is to fix all antennas at the same angle and then for each antenna sweep through all possible positions. For each antenna, we save the reward for each position it swept through. Then we compute the sample standard deviation for the scores of each antenna, and order the antennas according to standard deviation in descending order. We call this ordering “s”.

Finally, we also include the canonical order, where we just configure the antennas according to their numerical label. This order is called “o”.

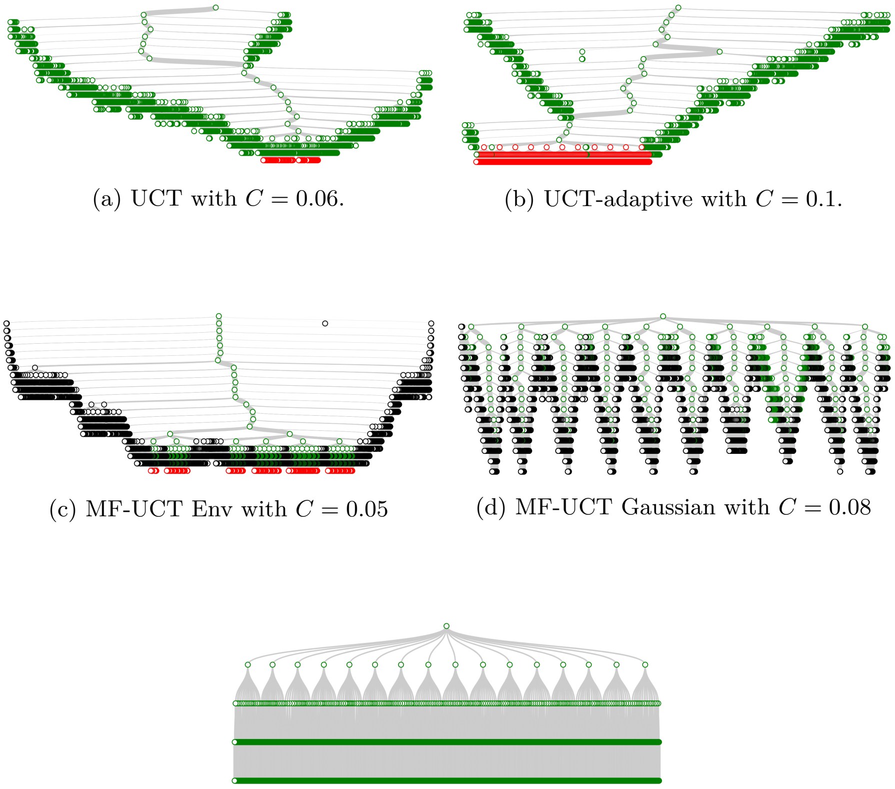
We now test the relative performance of each algorithm for the four different antenna orderings. Each algorithm keeps track of the best SINR value found so far after n simulator calls. All algorithms are given the same computational capital, which is simply a fixed number of N simulator calls. In the MF methods, the cost $\lambda^{(M)}$ of using the high-fidelity is 1, and the capital Λ corresponds to N simulator calls. This includes the points used to train the Gaussian process. All low-fidelity approximations have costs that are proportional to the time it takes to compute them. Thus, MF-UCT runs for approximately the same time as UCT.

Relative performance is determined by the highest SINR value, but also by the convergence rate. All MCTS versions are anytime algorithms, so we could stop them at any time and compare their relative performance at this point. Thus, we plot the running best SINR of each algorithm and look for the one that finds the best SINR value with the smallest capital spent. We expect that, given enough capital, UCT will converge to an optimal solution. Without our adaptations, this is theoretically guaranteed, and our closing of nodes is designed to make this happen faster. With this in mind, MF-UCT can at best be expected to converge to the same SINR as UCT, but we expect faster convergence.

We also expect that the different antenna orderings has an impact on the convergence time for the MCTS methods, which in some cases might lead to a lower SINR value found in the computational capital. However, given a large enough capital, we should see convergence to the same value.

5.3 Results

We begin by displaying some search trees to get an idea of what happens during the search. In Figure 11, we see the best performing parameters of the respective algorithms. In the plots, green nodes are open, while red ones are closed. For MF-UCT, a black node indicates that it has not been visited at the high-fidelity. Hence, a black node has been excluded by the low-fidelity search. We see that RND 11e is entirely asymmetric, as there is no selection. We see that UCT 11a, UCT-adaptive 11b, and MF-UCT Env 11c are similar in that they are asymmetric, with one major branch going deep enough to start closing nodes. The effect of shrinking the exploration parameter with time is evident, as UCT-adaptive starts with a high value, but still mange to close more nodes than UCT. The resulting search is probably too greedy, as UCT outperforms UCT-adaptive, as we shall see later. Finally, MF-UCT Gaussian also builds an asymmetric tree, but here we see many branches going deeper, but not as deep as the others. If we consider only the green, high-fidelity nodes, there seems to be an emphasis on the far right branch. From the two lower plots, we see that the number of black nodes is rather large, meaning that we succeed in reducing the search space significantly for the high-fidelity search.



(a) UCT with $C = 0.06$.

(b) UCT-adaptive with $C = 0.1$.

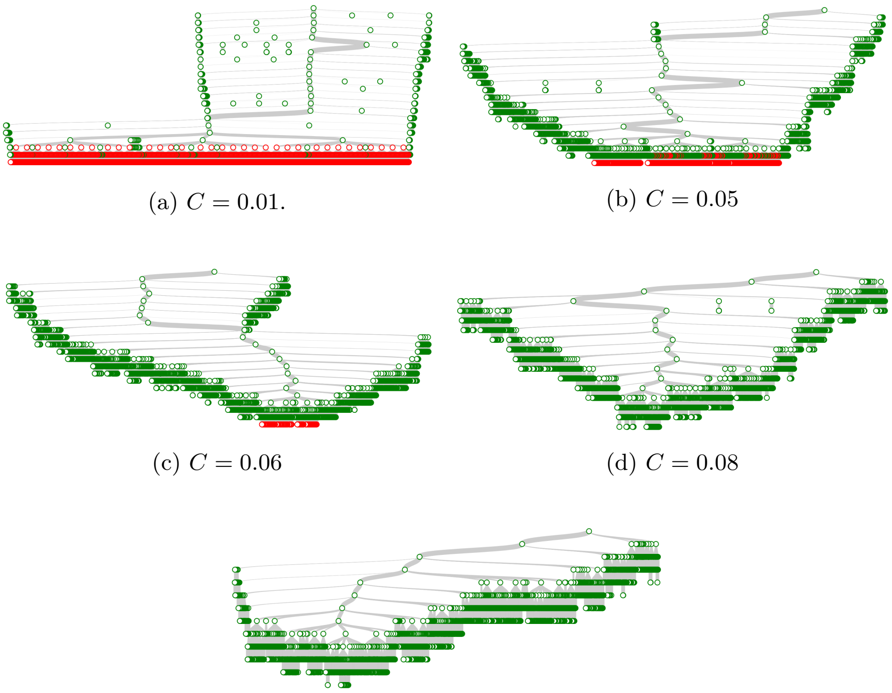
(c) MF-UCT Env with $C = 0.05$

(d) MF-UCT Gaussian with $C = 0.08$

(e) Random tree

Figure 11: Search trees for the MCTS methods. Green nodes are open, and red ones are closed. A black node in MF-UCT indicates that it has not been visited at the high fidelity. The exploration constants on display are those that turned out best in each case.

Figure 12 illustrates how different exploration parameters C builds different trees in UCT. In 12a, we see a very deep branch, but almost no exploration of the rest of the tree. We also see the effect of closing nodes once we reach the end. What happens is that the search progresses down the deep branch until it reaches the end, and then it spreads out in a kind of river delta pattern, as the "greedy" choice is no longer available. Given enough time, we would expect to see red nodes building towards the top of the tree almost symmetrically as they are recursively closed. Typically, for too low values of C, UCT does fairly well early in the search, but performance gain levels out quickly. Probably this is because almost all the search consists of trying different values for the last variables to be configured. Note that in a good antenna ordering, these last variables should have the least influence on the total network performance. The other trees are, relatively, more symmetric, but still very asymmetric. The value $C = 0.06$ performed best, and we see that the tree in 12c just reaches the final level, while the ones in 12d and 12e does not reach that far down. The latter is also, predictably, more symmetric, but still performs reasonably well.



(a) $C = 0.01$.

(b) $C = 0.05$

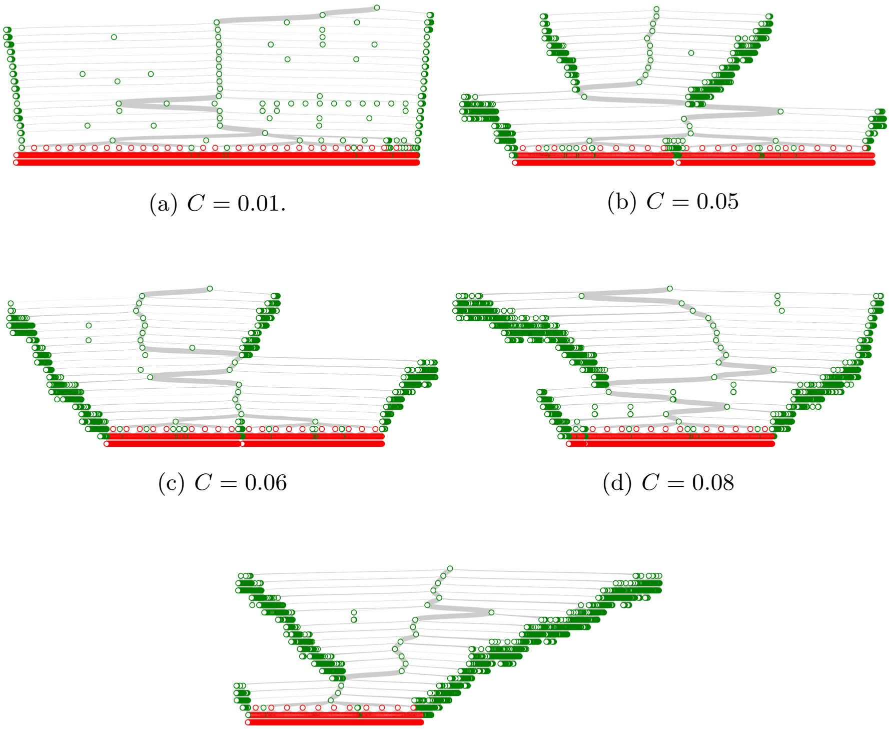
(c) $C = 0.06$

(d) $C = 0.08$

(e) $C = 0.1$.

Figure 12: The effect of varying the exploration parameter C in UCT.

Figure 13 shows some trees built by UCT-adaptive. The best results came from $C = 0.1$ in 13e. Our experiments did not consider any larger values for the exploration constant. We see that all trees get to the point of closing nodes recursively. Led by the shape of the best performing tree of UCT, 12c, where we just reach the terminal nodes, we might expect better performance of UCT-adaptive either by starting with a larger exploration parameter, or changing the rate at which it shrinks.



(a) $C = 0.01$.

(b) $C = 0.05$

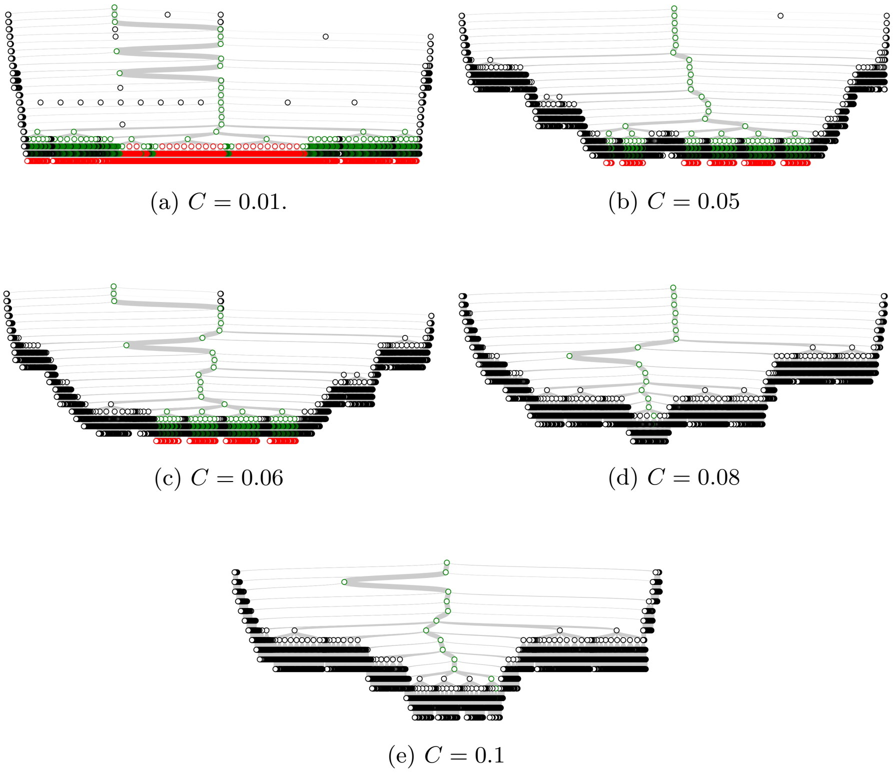
(c) $C = 0.06$

(d) $C = 0.08$

(e) $C = 0.1$.

Figure 13: The effect of varying the exploration parameter C in UCT-adaptive.

Some tree built by MF-UCT Env are shown in Figure 14. We see that here, varying the exploration parameter does not only change the shape of the tree, but also the length of the low-fidelity search. Recall that a black node indicates that it has not been visited at the highest fidelity. Recall also that MF-UCT Env does not use the heuristic to update $\gamma^{(1)}$. In the implementation, to avoid crashes, it does however impose that the last simulator call is at the high fidelity. In fact, both for $C = 0.08$ and $C = 0.1$, the search is done entirely at low-fidelity. This means that the exploration parameter is too high.



(a) $C = 0.01$.

(b) $C = 0.05$

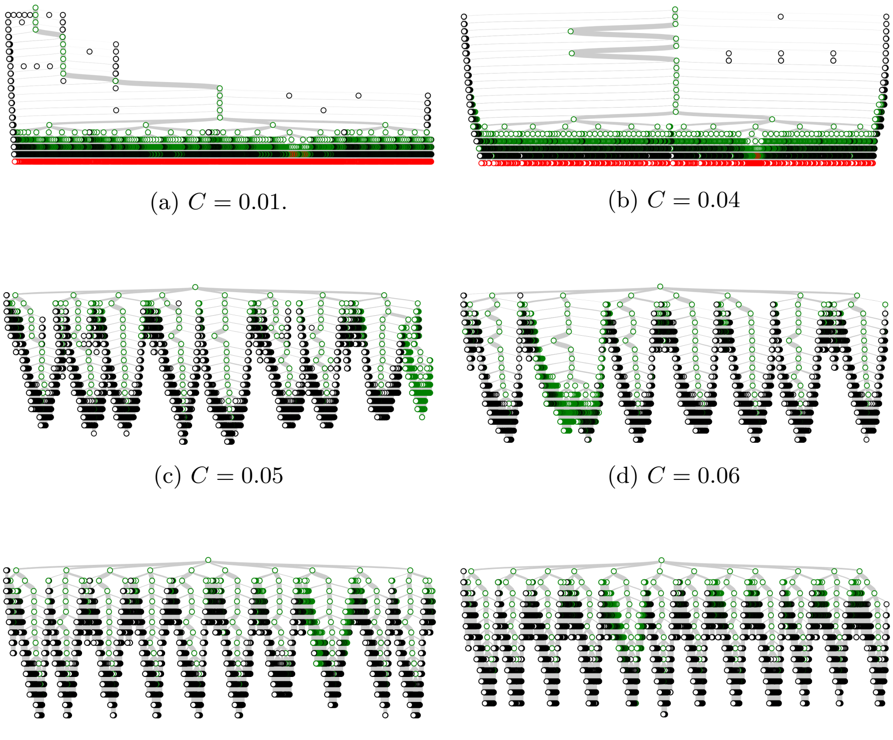
(c) $C = 0.06$

(d) $C = 0.08$

(e) $C = 0.1$

Figure 14: The effect of varying the exploration parameter C in MF-UCT Env.

The last batch of trees, built by MF-UCT Gaussian, is show in Figure 15. The best performance came from 15e. Interestingly, MF-UCT Gaussian builds a very different looking tree. It is as if the Gaussian process that is used for low-fidelity approximation struggles with deciding on which of the branches is the most promising. It is however able to exclude many, presumably, suboptimal branches early on. It is worth pointing out that the cost $\lambda^{(1)}$ of using the Gaussian process is very much cheaper than that of using the simulator with fewer users, as we do in MF-UCT Env. The low-fidelity search can therefore run for that much longer. For $C = 0.1$, the resulting tree contains 180384 nodes, most of which are excluded in the low-fidelity search. Interestingly, there is a sharp difference between the trees in 15b and 15c despite the exploration parameters being close to each other.



(a) $C = 0.01$.

(b) $C = 0.04$

(c) $C = 0.05$

(d) $C = 0.06$

(e) $C = 0.08$

(f) $C = 0.1$

Figure 15: The effect of varying the exploration parameter C in MF-UCT Gaussian.

Figure 16 shows the performance of the best performing algorithms on our network of 7 base stations. The plot shows the mean of six experiments using different random seeds, and a 95% confidence interval. Each experiment was assigned a computational budget of 10000 calls to the simulator. This budget was chosen to be large enough so that the algorithms have time to converge, but still within a reasonable running time. The plots indicate that the relative performance is somewhat stable after 8000 simulator calls. All MCTS methods ran with a single rollout per iteration. We tried multiple rollouts as well, but only small numbers. Multiple rollouts mean that the estimate of the mean reward of a node is better at each iteration, but it comes in this setting at the cost of fewer iterations. If one were to run multiple rollouts, say in parallel, while maintaining the number of iterations, we would expect faster convergence. The MCTS methods were tested with exploration parameters $C = 0.01, 0.02, \ldots, 0.1$.

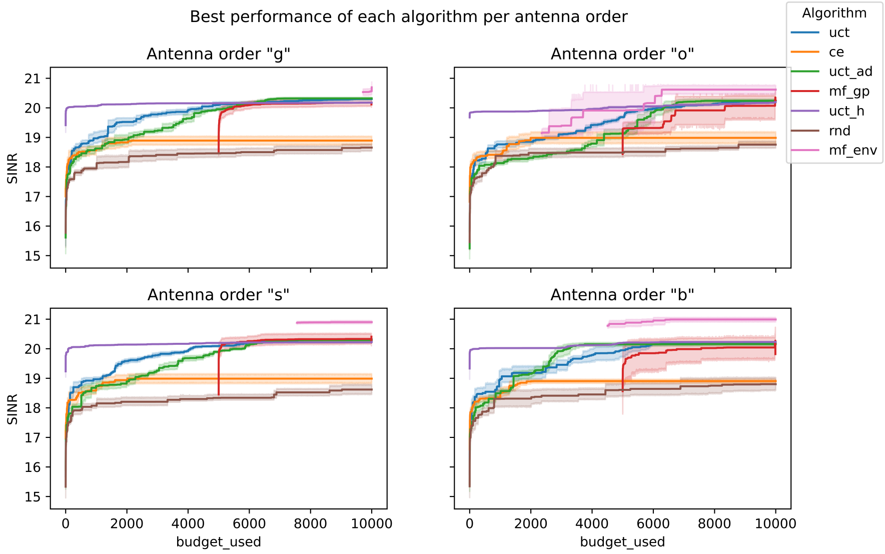


Figure 16: The best performing version of each algorithm per antenna ordering.

An initial observation from Figure 16 is that most UCT-like algorithms seem to converge to roughly the same average SINR per user, with the main difference being convergence time. The exception is MF-UCT Env, which in all cases reach a higher SINR. From Figure 17, we see that antenna order is important to obtain faster convergence. Interestingly, it also seems to have a big impact on the number of high-fidelity calls that MF-UCT Env makes.

We note that UCT-adaptive allows for more exploration in the beginning, leading to slower convergence than UCT, but is in all cases able to catch up to, and even slightly surpass UCT. Recall that we chose to linearly shrink the exploration constant with time. Other rates of adaptive exploration might perform better. It is also possible that a larger initial exploration parameter leads to better results. The heuristic approach, as expected, outclasses all other algorithms early on, but is eventually outperformed by all the other MCTS algorithms with the exception of Random tree.

MF-UCT Gaussian, in all cases, uses the first 5000 simulator calls to train a Gaussian process as the low-fidelity approximation. We store these SINR values in memory, so when MF-UCT Gaussian shows up in Figure 16, the first value is the best value that showed up in the random search that makes up the training data. After this point, MF-UCT Gaussian performs a low-fidelity search to exclude suboptimal branches from the tree, and the next simulator call we see in the plots is the first one that follows this low-fidelity search. The vertical line in the plots is thus the effect on SINR of the low-fidelity search. Interestingly, the relative performance of MF-UCT Gaussian seems to depend on the antenna order. Apparently, it struggles in bad orderings, which manifests in both lower average SINR and higher variance. It should however be noted that the performance depends heavily on the choice of low-fidelity approximation. Initial attempts using decision trees perform no better than random search. An interesting area for future research could be to figure out how the performance is affected by choice of low-fidelity approximation. Note also that our choice of $\gamma^{(1)}$ is heuristic in nature, and there may be much better choices available. Note also that if we were to have access to a pre-trained Gaussian process, the same result would heve been reached with half the computational budget, meaning that we can either save the time, or keep searching longer.

MF-UCT Env does not use the heuristic to adapt $\gamma^{(1)}$, as we are more confident in our estimation. The simulation environment with fewer users should be a good approximation of the full environment. The cost $\lambda^{(1)}$ is approximately half that of a high-fidelity simulator call, which means that MF-UCT Env can perform roughly twice the number of iterations that UCT does. As with MF-UCT, the plot in 16 shows only the high-fidelity search, so the first value that shows up is the first time that a node requests a high-fidelity call. In most antenna orderings, with the exception of “o”, this is already a higher value than the other algorithms reach. The length of the low-fidelity search depends partly on the exploration constant C. A smaller C value means that initially promising nodes get visited more often, leading to an earlier switch to high-fidelity and vice versa. All antenna orderings performed best with a C value around 0.05, with the exception of “o” which did best with $C = 0.01$. This accounts for the much earlier appearance in 16. Interestingly, the antenna order seems to influence the length of the low-fidelity search.

The Cross entropy method does relatively poorly, and seem to get stuck on a local maximum. Even with the restart property, it does not come close to the MCTS methods.

Finally, the Random tree has the poorest performance, which is as expected.

In Figure 17, we see the best performing experiment for each algorithm per antenna order. We see that the order in which we put the antennas does indeed matter for the overall performance. Interestingly, the order “b” that was chosen deliberately to be bad, actually performs better than the numerical ordering “o”. In most cases a bad ordering seems to result in slower convergence to about the same result, but in the case of MF-UCT with Gaussian process as low-fidelity approximation, the worse antenna ordering do not seem to catch up, but rather stabilize at lower values. This could be a consequence of the low-fidelity search, dismissing good paths in the tree early on. The early high performance of the “bad” antenna order “b” with UCT-adaptive in Figure 17 is due to the exploration parameter C of the best performing experiment being significantly lower than in the other orderings (0.04 and 0.1 respectively), resulting in a greedier search. It is eventually outperformed by the others. In general, both the “good” orderings “g” and “s” perform similarly, with “s” winning out when there is a notable difference. This is good news, as the method of choosing antenna ordering based on standard deviation is quite simple and requires no prior knowledge of the network. To be completely fair, the calculation of standard deviation requires $16 \cdot 21 = 336$ simulator calls that should be subtracted from the budget at the start. We have however not done this in our experiments.

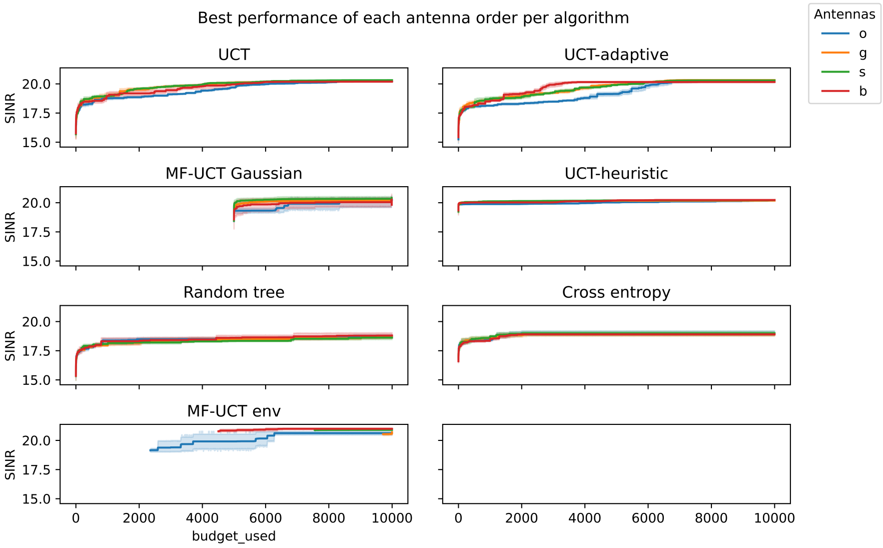


Figure 17: Each plot shows the best performing version of the respective algorithm for the different antenna orderings, which reveals the impact of antenna ordering on performance.

6 Conclusions and future directions

This thesis has presented a version of the UCT algorithm, adapted for black-box optimization. We also introduced a novel MF-UCT algorithm. These were then applied to the problem of antenna tilt optimization of a mobile network. Their performance was tested experimentally on mobile network optimization. Empirically, we established that MF-UCT converge to the same average SINR per user as UCT. Since the convergence of the latter is guaranteed asymptotically, the main issue to resolve is convergence rate. The MF-UCT algorithm can speed up the convergence significantly, given an existing low-fidelity approximation. We note however that when training a regression model, the result seems to be sensitive to the kind of model that is used. In fact, MF-UCT with a decision tree as low-fidelity did not perform any better than just random search. We believe that good performance is attainable using many kinds of regression models, but it may require lots of trial and error to set the correct parameters of the nodes. We also note that $\zeta^{(m)}$, the deviation in mean of the different fidelities is assumed to be a known quantity, which it typically is not in practice. Estimating this value so that the bound is tight could help in using the computational budget more efficiently. It is worth mentioning that we have so far only considered the MF-UCT in the case of one low-fidelity approximation. The algorithm allows for an arbitrary number. In fact, since even the simulation environment is an approximation of the real performance of a physical mobile network, MF-UCT would allow us to use real-world data as high-fidelity score. In that case, one would have to think carefully how to assign a reasonable cost of trying out a configuration in the real network.[2]

In the present work, we have not given a proof of the convergence of MF-UCT. This is left for future investigation.

One flaw of MF-UCT is that it does not allow for retraining the approximation as the search progresses. With this in mind, we have suggested two other approaches based on low-fidelity modelling. First we have a version of UCT, but instead of random rollouts, we use a separate low-fidelity UCT search to determine a promising configuration. We then score this, using the high-fidelity score, and use this to retrain the low-fidelity approximation. This is a kind of so-called heavy rollout policy. In their 2017 paper [28], James, Konidaris and Rosman show that this does not necessarily improve the performance of UCT even if the policy is good. Secondly, we propose something more akin to classical black-box algorithms. Run UCT only with low-fidelity approximation in generations: After each low-fidelity search, score top N assignments on the high-fidelity score. This is then used as training data to update the low-fidelity approximation.

As noted previously, the antenna ordering should impact the convergence rate. We have seen that this is indeed the case, and we have given two simple methods for coming up with supposedly good antenna orderings. Looking at the UCT

[2] This is probably not a reasonable use case. It is however possible, from the perspective of MF-UCT.

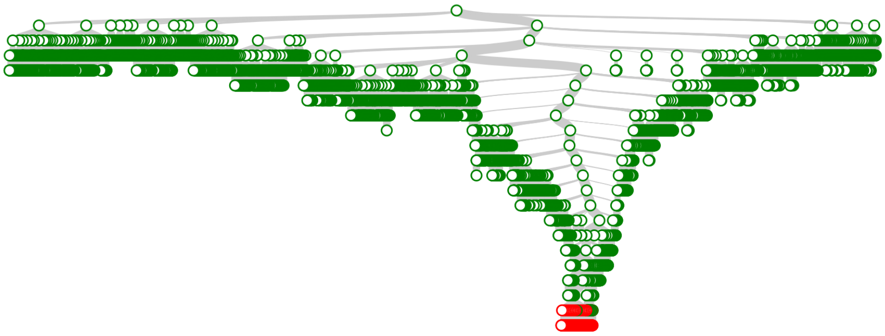


Figure 18: Trying a higher initial value $C = 0.3$ for UCT-adaptive.

plot in Figure 17, the worst ordering takes about 6000 simulators calls to reach the SINR that the best order got to in about 2000 calls. Future research into ways of picking good orderings could possibly result in much faster convergence.

Figure 18 shows a tree from UCT-adaptive with initial exploration parameter $C = 0.3$, which is much larger than anything our large scale experiments considered. The resulting tree displays lots of exploration in the higher levels, which is desirable, given a good antenna ordering, as these antennas are expected to have the greatest influence on the network performance. However, the tree still reaches terminal nodes, meaning that the main branch has been thoroughly explored. Further investigation into UCT-adaptive might include large scale experiments with more varied initial exploration parameters, as well as different rates of shrinking. It is also worth mentioning that MF-UCT could benefit from an adaptive exploration parameter. The situation there, however, is more complicated, as the exploration parameter effects the length of the low-fidelity search as well.

It is interesting to contrast our approach to existing reinforcement learning (RL) methods. Our methods do not require any data or any online learning, but can still produce good solutions by relying on a black-box model of the system. However, we only get one single configuration in the end. If the environment changes, we have to repeat the whole process from the start. In RL, by contrast, a control strategy is learned so that after training it can adapt to a changing environment. To find this control strategy in RL, we have to explore the search space, using for example ε-greedy strategies or UCB. Our methods may be superior to these exploration strategies in identifying more relevant configurations that could be used to generate more informative samples to train an RL agent. Notably, AlphaZero [29], the successor of AlphaGo uses a combination of RL and MCTS to learn the game of Go without any prior domain knowledge. It uses RL to guide the tree search, and it uses tree search to find good moves for the RL agent to learn.

Appendices

A Lower bound on regret for any strategy

Here we state and prove Theorem 2 of Lai and Robbins' classical paper. Let $\Pi_i, i = 1, \ldots, K$, with $K \geq 2$, denote statistical populations specified respectively by univariate density functions $f(x; \theta_i)$ with respect to some measure ν, where $f(\cdot; \cdot)$ is known and the θ_i are unknown parameters belonging to some set Θ. The multi-armed bandit problem consists of finding a strategy $\mathcal{A}$ that samples $x_1, x_2, \ldots$ from the K populations with the aim to maximise the expected value of the sum $S_n = x_1 + \cdots + x_n$ as $n \to \infty$. We will assume that $\int_{-\infty}^{\infty} |x| f(x; \theta_i) d\nu(x) < \infty$ for all $\theta_i \in \Theta$. Let $\mu(\theta) = \int_{-\infty}^{\infty} x f(x; \theta) d\nu(x)$. Then

$$\mathbb{E}[S_n] = \sum_{i=1}^{K} \mu(\theta_i) \mathbb{E}[T_i(n)]$$

where $T_i(n)$ as before is the number of times that $\mathcal{A}$ samples from Π_i up to time n. Maximizing this sum is equivalent to minimizing the pseudo-regret

$$R_n = \sum_{i : \mu(\theta_i) < \mu^*} \Delta_i \mathbb{E}[T_i(n)]$$

where, as usual

$$\mu^* = \max\{\mu(\theta_1), \ldots, \mu(\theta_k)\} = \mu(\theta^*) \text{ for some } \theta^* \in \{\theta_1, \ldots, \theta_k\}, \tag{A.1}$$

and $\Delta_i = \mu^* - \mu(\theta_i)$. Let $I(\theta, \lambda)$ denote the Kullback-Leibler number,

$$I(\theta, \lambda) = \int_{-\infty}^{\infty} \log\left(\frac{f(x; \theta)}{f(x; \lambda)}\right) f(x; \theta) d\nu(x). \tag{A.2}$$

Then $0 \leq I(\theta, \lambda) \leq \infty$. We shall assume that f is such that

$$0 < I(\theta, \lambda) < \infty \quad \text{whenever } \mu(\lambda) > \mu(\theta) \tag{A.3}$$

and that for every $\varepsilon > 0$ and all θ, λ such that $\mu(\lambda) > \mu(\theta)$, there is some $\delta = \delta(\varepsilon, \theta, \lambda) > 0$ such that

$$|I(\theta, \lambda) - I(\theta, \lambda')| < \varepsilon \text{ whenever } \mu(\lambda) \leq \mu(\lambda') \leq \mu(\lambda) + \delta. \tag{A.4}$$

Define $\theta = (\theta_1, \ldots, \theta_k)$ and let $\mathbb{P}_\theta$ denote the probability measure under which θ_i is the parameter corresponding to population Π_i. Define for $i = 1, \ldots, K$ the parameter sets

$$\begin{aligned} \Theta_i &= \left\{\theta : \mu(\theta_i) < \max\nolimits_{j \neq i} \mu(\theta_j)\right\} \quad (\text{"}\theta_i \text{ is not best"}) \\ \Theta_i^* &= \left\{\theta : \mu(\theta_i) > \max\nolimits_{j \neq i} \mu(\theta_j)\right\} \quad (\text{"}\theta_i \text{ is best"}). \end{aligned} \tag{A.5}$$

Now we are ready to state and prove the following theorem.

Theorem A.1. *Assume that $I(\theta, \lambda)$ satisfies* (A.3) *and* (A.4)*, and that Θ is such that for every $\lambda \in \Theta$ and for every $\delta > 0$, there is some $\lambda' \in \Theta$ such that*

$$\mu(\lambda) < \mu(\lambda') < \mu(\lambda) + \delta. \tag{A.6}$$

Fix $i \in \mathcal{K}$ and define Θ_i and Θ_i^* by (A.5)*. Let $\mathcal{A}$ be any allocation strategy such that for every $\theta \in \Theta_i^*$, as $n \to \infty$*

$$\sum_{j \neq i} \mathbb{E}_\theta[T_j(n)] = o(n^a) \textit{ for every } a > 0. \tag{A.7}$$

Then for every $\theta \in \Theta_i$ and every $\varepsilon > 0$,

$$\lim_{n\to\infty} \mathbb{P}_\theta\left(T_i(n) \geq \frac{(1-\varepsilon)\ln n}{I(\theta_i, \theta^*)}\right) = 1 \tag{A.8}$$

where θ^* is defined in (A.1)*. Hence,*

$$\liminf_{n\to\infty} \mathbb{E}_\theta[T_i(n)] \geq \frac{\ln n}{I(\theta_i, \theta^*)}.$$

Proof. We fix $i = 1, \theta \in \Theta_1$, and $\theta^* = \theta_2$. Then $\mu(\theta_2) > \mu(\theta_1)$, and $\mu(\theta_2) > \mu_j)$ for $j = 3, \ldots, K$. Fix any $0 < \delta < 1$. Then by (A.3), (A.4) and (A.6), we may choose $\lambda \in \Theta$ such that

$$\mu(\lambda) > \mu(\theta_2) \quad \text{and} \quad |I(\theta_1, \lambda) - I(\theta_1, \theta_2)| \leq \delta I(\theta_1, \theta_2). \tag{A.9}$$

We define the new parameter vector $\gamma = (\lambda, \theta_2, \ldots, \theta_K)$. Then $\gamma \in \Theta_1^*$, so by (A.7) we have

$$\mathbb{E}_\gamma[n - T_1(n)] = \sum_{h\neq 1} \mathbb{E}_\gamma[T_n(n)] = o(n^a)$$

with $0 < a < \delta$, and therefore

$$(n - O(\ln n))\mathbb{P}_\gamma\left(T_1(n) < \frac{(1-\delta)\ln n}{I(\theta_1, \lambda)}\right) \leq \mathbb{E}_\gamma[n - T_1(n)] = o(n^a).$$

Let $Y_1, Y_2, \ldots$ denote successive observations from Π_1, and define

$$L_m = \sum_{j=1}^m \ln\left(\frac{f(Y_j;\theta_1)}{f(Y_j, \lambda)}\right).$$

Then it follows that

$$P_\gamma(C_n) = o(n^{a-1}), \tag{A.10}$$

where

$$C_n = \left\{T_n(1) < \frac{(1-\delta)\ln n}{I(\theta_1, \lambda)} \text{ and } L_{T_1(n)} \leq (1-a)\ln n\right\}.$$

Note that

$$\begin{aligned}
&\mathbb{P}_\gamma(\{T_1(n) = n_1, \dots, T_k(n) = n_k, L_{n_1} \leq (1-a)\ln n\}) \\
&\quad = \int_{\{T_1(n)=n_1,\dots,T_k(n)=n_k, L_{n_1}\leq(1-a)\ln n)\}} \prod j = 1^{n_1} \frac{f(Y_j;\lambda)}{f(Y_j;\theta_1)} d\mathbb{P}_\theta \\
&\quad \geq e^{-(1-a)\ln n}(\mathbb{P}_\theta(\{T_1(n) = n_1, \dots, T_k(n) = n_k, L_{n_1} \leq (1-a)\ln n)\}).
\end{aligned} \tag{A.11}$$

Since C_n is a disjoint union of events of the form $\{T_1(n) = n_1, \dots, T_k(n) = n_k, L_{n_1} \leq (1-a)\ln n)\}$ with $n_1 + \dots + n_K = n$, and $n_1 \leq ((1-\delta)\ln n)/I(\theta_1, \lambda)$, it follows from (A.10) and (A.11) that, as $n \to \infty$

$$\mathbb{P}_\theta(C_n) \leq n^{1-a}\mathbb{P}_\lambda(C_n) \to 0. \tag{A.12}$$

By the strong law of large numbers, $L_m/m \to I(\theta_1, \lambda) > 0$, and therefore $\max_{j\leq m} L_j/m \to I(\theta_1, \lambda)$, a.s (i.e. with probability 1) $[\mathbb{P}_\theta]$. Since $1 - a > 1 - \delta$, it then follows that

$$\mathbb{P}_\theta\left(\left\{L_j > (1-a)\ln n, \text{ for some } j < \frac{(1-\delta)\ln n}{I(\theta_1,\lambda)}\right\}\right) \to 0 \tag{A.13}$$

as $n \to \infty$. From (A.12) and (A.13) we see that

$$\lim_{n\to\infty} \mathbb{P}_\theta\left(\left\{T_1(n) < \frac{(1-\delta)\ln n}{I(\theta-1,\lambda)}\right\}\right) = 0.$$

In view of (A.9) this implies that

$$\lim_{n\to\infty} \mathbb{P}_\theta\left(\left\{T_1(n) < \frac{(1-\delta)\ln n}{(1+\delta)I(\theta_1,\theta_2)}\right\}\right) = 0$$

from which (A.8) with $i = 1$ follows. □

B Proofs: Non-stationary bandit problem

This appendix contains the proofs of the theorems on the non-stationary bandit problem. The theorems are restated here for ease of reference.

Recall that by Assumption 4.1 we have that $\mu_i = \lim_{n\to\infty} \mu_{i,n}$ where $\mu_{i,n} = \mathbb{E}[\overline{X}_{i,n}]$. Moreover $\Delta_i = \mu^* - \mu_i$, and we define $\delta_{i,t}$ by

$$\mu_{i,n} = \mu_i + \delta_{i,n}.$$

Recall also the convention to mark optimals with "$*$", so that $\mu_{i^*,n} = \mu_n^*$. Since $\delta_{i,n}$ converges to zero, there exists for all $\varepsilon > 0$ a positive integer $N_0(\varepsilon)$ such that if $t \geq N_0(\varepsilon)$, then $|\delta_{i,t}| \leq \varepsilon\Delta_i/2$ and $|\delta_{i^*,t}| \leq \varepsilon\Delta_i/2$ whenever i is a suboptimal arm and i^* is an optimal arm. It follows in particular that for any optimal arm i^* and $t \geq N_0(\varepsilon)$ we have

$$|\delta_{i^*,t}| \leq \frac{\varepsilon}{2} \min_{i:\Delta_i>0} \Delta_i.$$

Theorem 4.1. *Consider (α,ψ)-UCB applied to a non-stationary bandit problem where the payoff sequence satisfies Assumption 4.1 and the exploration term $c_{t,s}$ is given by* (4.2)*. Fix $\varepsilon > 0$, and let $T_i(n)$ denote the number of plays at arm i after n time steps. Then, for $\alpha > 2$, if i is any suboptimal arm we have*

$$\mathbb{E}[T_i(n)] \leq \left\lceil \frac{\alpha \ln n}{\psi^*\left(\frac{(1-\varepsilon)\Delta_i}{2C_p}\right)} \right\rceil + N_0(\varepsilon) + N_p + \frac{\alpha}{\alpha-2}.$$

Proof. Let i be a suboptimal arm. What follows is similar to the Proof of Theorem 3.1. Recall that we denote the exploration term

$$c_{t,s} = C_p(\psi^*)^{-1}\left(\frac{\alpha \ln t}{s}\right),$$

and that $(\psi^*)^{-1}$ is a convex function with $(\psi^*)^{-1}(0) = 0$ (see eq. (3.6)). In particular, this means that $c_{t,s}$ increases with t, and decreases with s.

Let

$$A_0(n,\varepsilon) = \min\{n : c_{t,n} < (1-\varepsilon)\Delta_i/2\}.$$

Then, by definition of the exploration term,

$$A_0(n,\varepsilon) = \left\lceil \frac{\alpha \ln t}{\psi^*\left(\frac{(1-\varepsilon)\Delta_i}{2C_p}\right)} \right\rceil.$$

We define

$$A(n,\varepsilon) = \max\{A_0(n,\varepsilon), N_0(\varepsilon), N_p\}.$$

Similar to the Proof of Theorem 3.1 we have that for $t \geq N_0(\varepsilon)$, $I_t = i \neq i^*$ implies that at least one of the following must hold:

$$\overline{X}_{i^*,T_{i^*}(t-1)} + C_p(\psi^*)^{-1}\left(\frac{\alpha \ln t}{T_{i^*}(t-1)}\right) \leq \mu_t^*, \tag{B.1}$$

$$\overline{X}_{i,T_i(t-1)} \geq \mu_{i,t} + C_p(\psi^*)^{-1}\left(\frac{\alpha \ln t}{T_i(t-1)}\right), \tag{B.2}$$

$$T_i(t-1) < \frac{\alpha \ln t}{\psi^*\left(\frac{(1-\varepsilon)\Delta_i}{2C_p}\right)}. \tag{B.3}$$

For if none of (B.1)-(B.3) is true we find that

$$\begin{aligned}
&\overline{X}_{i^*,T_{i^*}(t-1)} + C_p(\psi^*)^{-1}\left(\frac{\alpha \ln t}{T_{i^*}(t-1)}\right) \\
&> \mu_t^* \\
&= \mu^* + \delta_{i^*,t} \\
&= \mu_i + \Delta_i + \delta_{i^*,t} \\
&= \mu_{i,t} + \Delta_i - (\delta_{i,t} - \delta_{i^*,t}) \\
&\geq \mu_{i,t} + (1-\varepsilon)\Delta_i \\
&\geq \mu_{i,t} + 2C_p(\psi^*)^{-1}\left(\frac{\alpha \ln t}{T_i(t-1)}\right) \\
&> \overline{X}_{i,T_i(t-1)} + C_p(\psi^*)^{-1}\left(\frac{\alpha \ln t}{T_i(t-1)}\right)
\end{aligned}$$

which again in particular means that $I_t \neq i$.

For $T_i(t-1) \geq A(n, \varepsilon)$, equation (B.3) is false by the definition of $A(n, \varepsilon)$. Note that this also implies that $t \geq T_i(t-1)$ (arm i cannot have more plays than time steps) which in particular, by Assumption 4.1, means that (4.3) holds. We now take expectation as in the Proof of Theorem 3.1:

$$\begin{aligned}
\mathbb{E}[T_i(n)] &\leq A(n,\varepsilon) + \mathbb{E}\left[\sum_{t=A(n,\varepsilon)+1}^{n} \{I_t = i \text{ and (B.3) is false}\}\right] \\
&\leq A(n,\varepsilon) + \mathbb{E}\left[\sum_{t=A(n,\varepsilon)+1}^{n} \{\text{(B.1) or (B.2) is true}\}\right] \\
&= A(n,\varepsilon) + \sum_{t=A(n,\varepsilon)+1}^{n} \Big(\mathbb{P}(\text{(B.1) or (B.2) is true})\Big) \\
&\leq A(n,\varepsilon) + \sum_{t=A(n,\varepsilon)+1}^{n} \Big(\mathbb{P}(\text{(B.1) is true}) + \mathbb{P}(\text{(B.2) is true})\Big).
\end{aligned}$$

Similar to the Proof of Theorem 3.1, but now using (4.3), we bound the probability of the events (B.1) and (B.2). We have

$$\begin{aligned}\mathbb{P}((\text{B.1) is true}) &\leq \mathbb{P}\Big(\exists s \in \{1,\dots,t\} : \overline{X}_{i^*,s} + C_p(\psi^*)^{-1}\Big(\frac{\alpha \ln t}{s}\Big) \leq \mu_t^*\Big) \\ &\leq \sum_{s=1}^{t} \mathbb{P}\Big(\mu_t^* - \overline{X}_{i^*,s} \geq C_p(\psi^*)^{-1}\Big(\frac{\alpha \ln t}{s}\Big)\Big) \\ &= \sum_{s=1}^{t} \frac{1}{t^\alpha} = \frac{1}{t^{\alpha-1}},\end{aligned}$$

and the same bound holds for (B.2) so repeating the calculations we see that

$$\mathbb{E}[T_i(n)] \leq A(n,\varepsilon) + \frac{\alpha}{\alpha-2} \leq \left\lceil \frac{\alpha \ln n}{\psi^*\big(\frac{(1-\varepsilon)\Delta_i}{2C_p}\big)} \right\rceil + N_0(\varepsilon) + N_p + \frac{\alpha}{\alpha-2}$$

which proves the theorem. □

Theorem 4.2. *Let*

$$\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} T_i(n)\overline{X}_{i,T_i(n)}.$$

Then, under Assumption 4.1 we have

$$|\mathbb{E}[\overline{X}_n] - \mu^*| \leq |\Delta_n(\delta)^*| + \frac{2}{n}\sum_{i\neq i^*}\left(\left\lceil \frac{\alpha \ln n}{\psi^*\big(\frac{(1-\varepsilon)\Delta_i}{2C_p}\big)} \right\rceil + N_0(\varepsilon) + N_p + \frac{\alpha}{\alpha-2}\right).$$

Proof. Without loss of generality, we assume that there exists a unique optimal arm i^*. By the triangle inequality,

$$|\mu^* - \mathbb{E}[\overline{X}_n]| \leq |\mu^* - \mu_n^*| + |\mu_n^* - \mathbb{E}[\overline{X}_n]| = |\delta_n^*| + |\mu_n^* - \mathbb{E}[\overline{X}_n]|.$$

We bound the last term as follows:

$$\begin{aligned} n|\mu^* - \mathbb{E}[\overline{X}_n]| &= \Big|\sum_{t=1}^{n} \mathbb{E}[X_{i^*,t}] - \mathbb{E}[\sum_{i=1}^{K} T_i(n)\overline{X}_{i,T_i(n)}]\Big| \\ &\leq \Big|\sum_{t=1}^{n} \mathbb{E}[X_{i^*,t}] - \mathbb{E}[T_{i^*}(n)\overline{X}_{i^*,T_{i^*}(n)}]\Big| + \mathbb{E}[\sum_{i\neq i^*}^{K} T_i(n)\overline{X}_{i,T_i(n)}].\end{aligned}$$

Since $0 \leq \overline{X}_{i,T_i(n)} \leq 1$, the last term is bounded by the total number of plays on all the suboptimal arms until time step n. By Theorem 4.1 this in turn is bounded by

$$\sum_{i\neq i^*}\left(\left\lceil \frac{\alpha \ln n}{\psi^*\big(\frac{(1-\varepsilon)\Delta_i}{2C_p}\big)} \right\rceil + N_0(\varepsilon) + N_p + \frac{\alpha}{\alpha-2}\right). \qquad \text{(B.4)}$$

To bound the first term, note that $T_{i^*}(n)\overline{X}_{i^*,T_{i^*}(n)} = \sum_{t=1}^{T_{i^*}(n)} X_{i^*,t}$, and

$$\begin{aligned}D_n &:= \sum_{t=1}^{n} \mathbb{E}[X_{i^*,t}] - \mathbb{E}\Bigg[\sum_{t=1}^{T_{i^*}(n)} X_{i^*,t}\Bigg] = \mathbb{E}\Bigg[\sum_{t=1}^{n} X_{i^*,t} - \sum_{t=1}^{T_{i^*}(n)} X_{i^*,t}\Bigg] \\ &= \mathbb{E}\Bigg[\sum_{t=T_{i^*}}^{n} X_{i^*,t}\Bigg] \geq 0.\end{aligned}$$

Since also $X_{i^*,t} \leq 1$, D_n is bounded by $\mathbb{E}[n - T_{i^*}(n)] = \sum_{i \neq i^*} \mathbb{E}[T_i(n)]$, which again by Theorem 4.1 is bounded by (B.4). Collecting the term yields

$$\begin{aligned}|\mu^* - \mathbb{E}[\overline{X}_n]| &\leq |\delta_n^*| + |\mu_n^* - \mathbb{E}[\overline{X}_n]| \\ &\leq |\delta_n^*| + \frac{2}{n} \sum_{i \neq i^*} \left(\left\lceil \frac{\alpha \ln n}{\psi^*\big(\frac{(1-\varepsilon)\Delta_i}{2C_p}\big)} \right\rceil + N_0(\varepsilon) + N_p + \frac{\alpha}{\alpha - 2} \right)\end{aligned}$$

which is what we wanted to show. □

Theorem 4.3. *For $\delta > 0$, let $\Delta_n(\delta) = 9\sqrt{2n \ln(2/\delta)}$, and let*

$$\sqrt{n_0} \geq \frac{1}{\sqrt{2 \ln 2}} \sum_{i=1}^{K} \left(\left\lceil \frac{\alpha \ln n_0}{\psi^*\big(\frac{(1-\varepsilon)\Delta_i}{2C_p}\big)} \right\rceil + N_0(\varepsilon) + N_p + \frac{\alpha}{\alpha - 2} \right).$$

Then for any $n \geq n_0$, we have the following bounds

$$\mathbb{P}(n\big|\overline{X}_n \geq -\mathbb{E}[\overline{X}_n]\big| \geq \Delta_n(\delta)) \leq \delta.$$

Proof. The proof relies on Lemma 2.6. For simplicity, we assume that the payoffs of the optimal arm are i.i.d. The general case can be treated similarly, as indicated in the proof of Lemma 2.6. Note also that Δ_i refers to the deviation in expectation as defined in Section 3, while $\Delta_n(\delta)$ is defined in the theorem. Let Z_t be the indicator function of the event that a suboptimal arm is chosen at time step t. Then by Theorem 4.1,

$$\begin{aligned}\mathbb{E}\Bigg[\sum_{t=1}^{n} Z_t\Bigg] &= \mathbb{E}\Bigg[\sum_{t=1}^{n}\sum_{i=1}^{K} \{\text{arm } i \text{ played and arm } i \text{ is suboptimal}\}\Bigg] \\ &\leq \mathbb{E}\Bigg[\sum_{t=1}^{n}\sum_{i=1}^{K} \{\text{arm } i \text{ played}\}\Bigg] \\ &= \mathbb{E}\Bigg[\sum_{i=1}^{K} T_i(n)\Bigg] \\ &\leq \sum_{i=1}^{K} \left(\left\lceil \frac{\alpha \ln n}{\psi^*\big(\frac{(1-\varepsilon)\Delta_i}{2C_p}\big)} \right\rceil + N_0(\varepsilon) + N_p + \frac{\alpha}{\alpha - 2} \right).\end{aligned}$$

Hence a_n of Lemma 2.6 may be chosen as

$$a_n = \sum_{i=1}^{K} \left(\left\lceil \frac{\alpha \ln n}{\psi^*\left(\frac{(1-\varepsilon)\Delta_i}{2C_p}\right)} \right\rceil + N_0(\varepsilon) + N_p + \frac{\alpha}{\alpha - 2} \right).$$

Further, we identify the payoff sequence of the optimal arm with X_t of Lemma 2.6, and we let Y_t denote the payoff received at time t. Then, by assumption $X_t, Y_t \in [0,1]$, and $n\overline{X}_n = \sum_{t=1}^{n}((1-Z_t)X_t + Z_t Y_t)$. Note that R_n of Lemma 2.6 corresponds to the expected total regret at time n. Hence, by Theorem 4.1,

$$R_n \leq \sum_{i=1}^{K} \Delta_i \left(\left\lceil \frac{\alpha \ln n}{\psi^*\left(\frac{(1-\varepsilon)\Delta_i}{2C_p}\right)} \right\rceil + N_0(\varepsilon) + N_p + \frac{\alpha}{\alpha - 2} \right).$$

Now let n_0 be an index such that for all $n \geq n_0$, $a_n \leq \Delta_n(\delta)/9$ and $R_n \leq 2\Delta_n(\delta)/9$. Since $\Delta_n(\delta) = O(\sqrt{n})$ and $a_n, R_n = O(\ln n)$ such an index exists. Then for $n \geq n_0$ the conditions of Lemma 2.6 are satisfied and hence the tail-inequalities hold for $\overline{X}_n$. Note moreover that since for $\delta \leq 1$, $\Delta_n(\delta) = 9\sqrt{2n \ln 2/\delta} \geq 9\sqrt{2n \ln 2}$, we may choose n_0 independently of δ. In fact, since by assumption, $\Delta_i \in [0,1]$ we may choose n_0 as the first integer so that

$$\sqrt{n} \geq \frac{1}{\sqrt{2\ln 2}} \sum_{i=1}^{K} \left(\left\lceil \frac{\alpha \ln n}{\psi^*\left(\frac{(1-\varepsilon)\Delta_i}{2C_p}\right)} \right\rceil + N_0(\varepsilon) + N_p + \frac{\alpha}{\alpha - 2} \right).$$

The theorem now follows. □

Theorem 4.4. *It holds that*

$$\lim_{t \to \infty} \mathbb{P}(I_t \neq i^*) = 0$$

Proof. Fix $\varepsilon > 0$. We will show that if t is sufficiently large, then $\mathbb{P}(I_t \neq i^*) \leq \varepsilon$. Let i be a suboptimal arm and let $p_{i,t} = \mathbb{P}(\overline{X}_{i,T_i(t)} \geq \overline{X}_{i,T_i(t)})$. Then

$$\mathbb{P}(I_t \neq i^*) \leq \sum_{i \neq i^*} p_{i,j},$$

and it suffices to show that $p_{i,t} \leq \varepsilon/K$ for all suboptimal arms and sufficiently large t. Recall that by definition, $\Delta_i = \mu^* - \mu_i$, so clearly if $\overline{X}_{i,T_i(t)} < \mu_i + \Delta_i/2$ and $\overline{X}_{i^*,T_{i^*}(t)} < \mu_{i^*} + \Delta_i/2$ then $\overline{X}_{i,T_i(t)} < \overline{X}_{i,T_i(t)}$. Hence,

$$p_{i,t} \leq \mathbb{P}(\overline{X}_{i,T_i(t)} \geq \mu_i + \Delta_i/2) + \mathbb{P}(\overline{X}_{i^*,T_{i^*}(t)} \leq \mu_{i^*} + \Delta_i/2). \tag{B.5}$$

We bound the first term of (B.5) as follows. First note that

$$\begin{aligned} \mathbb{P}(\overline{X}_{i,T_i(t)} \geq \mu_i + \Delta_i/2) &= \mathbb{P}(\overline{X}_{i,T_i(t)} \geq \mu_{i,T_i(t)} - \delta_{i,T_i(t)} + \Delta_i/2) \\ &\leq \mathbb{P}(\overline{X}_{i,T_i(t)} \geq \mu_{i,T_i(t)} - |\delta_{i,T_i(t)}| + \Delta_i/2). \end{aligned}$$

When $T_i(t) \geq N_0(\varepsilon)$, $|\delta_i, T_i(t)| \leq \varepsilon\Delta_i/2$, so we have

$$\mathbb{P}(\overline{X}_{i,T_i(t)} \geq \mu_{i,T_i(t)} - |\delta_{i,T_i(t)}| + \Delta_i/2) \leq \mathbb{P}\bigg(\overline{X}_{i,T_i(t)} \geq \mu_{i,T_i(t)} - \frac{(1-\varepsilon)\Delta_i}{2}\bigg).$$

By Assumption 3.1, with $\delta = \frac{\varepsilon}{2K}$ we have

$$\Delta_n(\delta)(\frac{\varepsilon}{2K}) = \frac{C_p}{n}(\psi^*)^{-1}\bigg(\frac{1}{n}\ln\frac{2K}{\varepsilon}\bigg),$$

and hence for all $n \geq N_p$,

$$\begin{aligned}&\mathbb{P}\bigg(\overline{X}_{i,T_i(t)} \geq \mu_{i,T_i(t)} - \frac{1}{n}\Delta_n(\delta)\bigg(\frac{\varepsilon}{2K}\bigg)\bigg)\\ &= \mathbb{P}\bigg(\overline{X}_{i,T_i(t)} \geq \mu_{i,T_i(t)} - \frac{C_p}{n^2}(\psi^*)^{-1}\bigg(\frac{1}{n}\ln\frac{2K}{\varepsilon}\bigg)\bigg)\\ &\leq \frac{\varepsilon}{2K}.\end{aligned}$$

Now, $\mathbb{P}(\overline{X}_{i,T_i(t)} \geq \mu_{i,T_i(t)} - \frac{(1-\varepsilon)\Delta_i}{2}) \leq \mathbb{P}(\overline{X}_{i,T_i(t)} \geq \mu_{i,T_i(t)} - x)$ if $x \leq \frac{(1-\varepsilon)\Delta_i}{2}$ for any real x. Hence, we want

$$\frac{C_p}{n}(\psi^*)^{-1}\bigg(\frac{1}{n}\ln\frac{2K}{\varepsilon}\bigg) \leq \frac{(1-\varepsilon)\Delta_i}{2}$$

which holds if

$$\ln\frac{2K}{\varepsilon} \leq n\psi^*\bigg(\frac{(1-\varepsilon)\Delta_i}{2C_p}n^2\bigg). \tag{B.6}$$

The right-hand side of (B.6) tends to ∞ as $n \to \infty$ since ψ^* is convex, so there is some $A(\varepsilon)$ such that if $n \geq a$ (B.6) holds. Let $m = \max\{A(\varepsilon), N_0(\varepsilon), N_p\}$. Then, for $T_i(t) \geq m$ we have

$$\mathbb{P}\bigg(\overline{X}_{i,T_i(t)} \geq \mu_{i,T_i(t)} - \frac{(1-\varepsilon)\Delta_i}{2}\bigg) \leq \frac{\varepsilon}{2K}.$$

By Theorem A.1 there is a constant $\rho > 0$ such that

$$\lim_{t\to\infty}\mathbb{P}(T_i(t) \geq \rho\ln t) = 1$$

so we conclude that there is some time t for which $T_i(t) \geq m$. Hence, we have, for sufficiently large t that the first term of (B.5),

$$\mathbb{P}(\overline{X}_{i,T_i(t)} \geq \mu_i + \Delta_i/2) \leq \frac{\varepsilon}{2K}.$$

The second term is bounded similarly. Collecting the bounds yield that $p_{i,t} \leq \varepsilon/K$ for sufficiently large t which concludes the proof. □

C The Legendre-Fenchel transform

First some preliminary definitions.

Definition C.1. *A function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is called lower semicontinuous at $x_0 \in \mathbb{R}^n$ if for every real $y < f(x_0)$ there exists a neighbourhood U of x_0 such that $f(x) > y$ for all $x \in U$. If f is lower semicontinuous at every point in its domain, we say that f is lower semicontinuous. Equivalently, f is lower semicontinuous if its epigraph*

$$\mathrm{epi(f)} = \{(x, t) \in \mathbb{R}^n \times \overline{\mathbb{R}} : f(x) \leq t\}$$

is a closed set.

Note that any continuous function is in particular lower semicontinuous. In the interest of brevity we do not prove the equivalence of the two definitions. We will use them both.

Lemma C.1. *Let $\{f_i\}_{i \in I}$ be any collection of lower semicontinuous functions on $\mathbb{R}^n$, and define the function g by*

$$g(x) = \sup_{i \in I}\{f_i(x)\}$$

for all $x \in \mathbb{R}^n$. Then g is lower semicontinuous.

Proof. Fix $x_0 \in \mathbb{R}^n$ and some real $y < g(x_0)$. Then there is some $i \in I$ such that $y < f_i(x_0)$. But f_i is lower semicontinuous, so there is some neighbourhood U of x_0 such that $y < f_i(x)$ for all $x \in U$. Since, by definition $f_i(x) < g(x)$, we also have $y < g(x)$ for all $x \in U$. In other words, g is lower semicontinuous at x_0. Since x_0 was arbitrary, g is lower semicontinuous. □

We now extend the familiar concept of convexity to the extended reals.

Definition C.2. *A function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is called convex if its epigraph*

$$\mathrm{epi(f)} = \{(x, t) \in \mathbb{R}^n \times \overline{\mathbb{R}} : f(x) \leq t\}$$

is a convex set.

If a function f takes on both infinite values, say $f(x_1) = -\infty$ and $f(x_2) = \infty$. Then $\lambda f(x_1) + (1 - \lambda) f(x_2)$ is undefined, but epi(f) still makes sense. One possibility is to take as a definition that functions taking the value $-\infty$ are not convex. We will however follow Rockafellar [30] and regard such functions as convex, but call them *improper*. Finally, the function that is identically ∞ is in this sense convex but improper. For clarity we make a formal definition.

Definition C.3. *A function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is called proper if it does not attain the value $-\infty$, and its domain $domf = \{x \in \mathbb{R}^n : f(x) \in \mathbb{R}\}$ is non-empty. In particular, there exists some $x_0 \in \mathbb{R}^n$ so that $f(x_0) < \infty$.*

Lemma C.2. *Let $\{f_i\}_{i\in I}$ be any collection of convex functions $\mathbb{R}^n \to \overline{\mathbb{R}}$. The function g, defined by*

$$g(x) = \sup_{i\in I}\{f_i(x)\}$$

is then a convex function

C.1 The LF transform

Definition C.4. *Let $\overline{\mathbb{R}}$ denote the extended real numbers, $\mathbb{R}\cup\{-\infty,\infty\}$. For a proper function $f:\mathbb{R}^n\to\overline{\mathbb{R}}$, we define the Legendre-Fenchel transform of f at $k\in X$ as*

$$f^*(k) = \sup_{x\in\mathbb{R}^n}\{\langle x,k\rangle - f(x)\}. \tag{C.1}$$

where $\langle\cdot,\cdot\rangle$ is the usual inner product on $\mathbb{R}^n$.

It is worth noting that the name LF transform is not entirely universal. In the literature, (C.1) is also known as the convex conjugate and sometimes the Fenchel conjugate. Perhaps the latter is preferable, as Fenchel was the one to study the variational formula (C.1), which in the case of f being convex and differentiable reduces to the Legendre transform commonly used in classical mechanics. We will however refer to (C.1) as the LF transform.

Immediate from the definition is the *Fenchel-Young inequality,*

$$f(x) + f^*(k) \geq \langle x,k\rangle \tag{C.2}$$

which holds for all $x,k\in\mathbb{R}^n$.

Theorem C.1. *Let $f:\mathbb{R}^n\to\overline{\mathbb{R}}$ be a proper function. Then $f^*:\mathbb{R}^n\to\overline{\mathbb{R}}$ is convex and lower semi-continuous.*

Proof. Fix $\bar{x}\in\operatorname{dom}f$. Then $f(\bar{x})\in\mathbb{R}$, and for any $k\in\mathbb{R}^n$ we have, by definition,

$$f^*(k) = \sup_{x\in\mathbb{R}^n}\{\langle x,k\rangle - f(x)\} \geq \langle\bar{x},k\rangle - f(\bar{x}) > -\infty.$$

Note that if $x\notin\operatorname{dom}f$ then $\langle x,k\rangle - f(x) = -\infty$, so we have that

$$f^*(k) = \sup_{x\in\operatorname{dom}f}\{\langle x,k\rangle - f(x)\} = \sup_{x\in\operatorname{dom}f}\{\phi_x(k)\}$$

where $\phi_x(k) := \langle x,k\rangle - f(x)$ are affine (in particular convex), continuous functions on $\mathbb{R}^n$. Then f^* is lower semicontinuous by Lemma C.1, and convex by Lemma C.2. □

Applying the LF transform twice leads to the *double LF transform.*

Definition C.5. *Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a proper function. The double LF transform f^{**} is defined by*

$$f^{**}(x) = \sup_{k\in\mathbb{R}^n} \{\langle x, k\rangle - f^*(k)\}.$$

A central problem in the study of the LF transform is to answer when $f^{**}(x) = f(x)$. Whenever this holds, we say that the LF transform is *involutive.* In light of Theorem C.1 this can only hold if f is convex, since f^{**} is always convex, being the LF transform of f^*.

A simple observation is that if $g \le f$ in the sense that $g(x) \le f(x)$ for all x, then

$$g^*(k) = \sup_{k\in\mathbb{R}}\{kx - g(x)\} \ge \sup_{k\in\mathbb{R}}\{kx - f(x)\} = f^*(k),$$

and we write $g^* \ge f^*$. We say that the LF transform is *order reversing.*

Theorem C.2. *For all $x \in \mathbb{R}^n$ we have $f^{**}(x) \le f(x)$.*

Proof. Begin by fixing $x, k \in \mathbb{R}^n$. By the Fenchel-Young inequality (C.2) we have $f(x) + f^*(k) \ge \langle x, k\rangle$. Hence $f(x) \ge \langle x, k\rangle - f^*(k)$, and this holds for all $k \in \mathbb{R}^n$. We conclude that

$$f(x) \ge \sup_{k\in\mathbb{R}^n} \{\langle x, k\rangle - f^*(k)\} = f^{**}(x).$$

The theorem follows since x was arbitrary. □

We now move on to give a complete description of when the LF transform is involutive. First, we will need the fundamental result from convex analysis that a closed convex set can always be separated from a point not in the set, with a hyperplane. There are many separating hyperplane theorems, but we only need the most fundamental one here, which we state without proof.

Theorem C.3 (Separation of a convex set and a point)**.** *Let X be a nonempty closed convex set in R^n and $y \notin X$. Then there exists a vector $p \neq 0$ and $\alpha \in \mathbb{R}$ such that $\langle p, y\rangle > \alpha$ and $\langle p, x\rangle \le \alpha$ for all $x \in X$. We say that the hyperplane $H = \{x : \langle p, x\rangle = \alpha\}$ separate X and y.*

The proof of this important theorem can be found in any textbook on convex analysis or optimization, e.g. [31].

Lemma C.3. *If $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is convex and lower semicontinuous, then*

$$f(x) = \sup_{a\le f} a(x)$$

for all $x \in \mathbb{R}^n$, where the supremum is taken over all affine functions such that $a(x) \le f(x)$ for all x, the affine minorants of f.

Proof. By assumption, epi(f) is closed and convex. Let $(x_0, y_0) \notin \text{epi}(f)$, so that in particular $y_0 < f(x_0)$. By the separating hyperplane theorem (Theorem C.3) there exists some vector $(p_1^t, p_2)^t \neq 0$ $(p_1 \in \mathbb{R}^n, p_2 \in \mathbb{R})$, and a real α, such that

$$\langle (p_1^t, p_2)^t, (x_0, y_0) \rangle = \langle p_1, x_0 \rangle + p_2 y_0 > \alpha,$$

and

$$\langle (p_1^t, p_2)^t, (x, t) \rangle = \langle p_1, x \rangle + p_2 t \leq \alpha$$

for all $(x, t) \in \text{epi}(f)$. In particular,

$$\langle p_1, x_0 \rangle + p_2 f(x_0) \leq \alpha.$$

If $p_2 = 0$ then $\langle p_1, x \rangle \leq \alpha$ for all $x \in \mathbb{R}^n$ but also $\langle p_1, x_0 \rangle > \alpha$, so we must have $p_2 \neq 0$. Then we can scale it so that $p_2 = -1$ which means that

$$f(x_0) \geq \langle \tilde{p}_1, x_0 \rangle - \tilde{\alpha} > y_0.$$

So, for any point $(x, y) \notin \text{epi}(f)$, there exists p, α such that $f(x) \geq \langle p, x \rangle - \alpha > y$. Since $\sup_{y > f(x)} y = f(x)$ we must have

$$f(x) = \sup_{\langle p, x \rangle - \alpha \leq f(x)} (\langle p, x \rangle - \alpha) = f(x) = \sup_{a \leq f} a(x)$$

which concludes the proof. □

Theorem C.4 (Fenchel-Moreau)**.** *If $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is convex and lower semicontinuous, then $f^{**}(x) = f(x)$ for all $x \in \mathbb{R}^n$*

Before proving this important result, we need to establish that it holds in the special case when f is affine, i.e $f(x) = \langle p, x \rangle - \alpha$ for some $p \in \mathbb{R}^n$ and real α. We then have

$$f^*(k) = \sup_{x \in \mathbb{R}^n} \{\langle k, x \rangle - \langle p, x \rangle + \alpha\} = \begin{cases} \alpha, & k = p \\ \infty, & k \neq p \end{cases},$$

which in turn means that

$$f^{**}(x) = \sup_{k \in \mathbb{R}^n} \left\{ \langle k, x \rangle - \begin{cases} \alpha, & k = p \\ \infty, & k \neq p \end{cases} \right\} = \langle p, x \rangle - \alpha = f(x).$$

Proof of Theorem C.4. Assume first that f is convex and lower semicontinuous. We know already that $f^{**}(x) \leq f(x)$, so we wish to establish the reverse inequality. Let a be an affine minorant of f, that is a is affine and $a(x) \leq f(x)$ for all x. Then, by the order reversing property of the LF transform, we have $a^* \geq f^*$. Repeating the argument yields $a^{**} \leq f^{**}$. But as we have seen, $a^{**} = a$ since a is affine, so we have shown that any affine minorant of f is also an affine minorant of f^{**}. By Theorem C.1 f^{**} is convex and lower semicontinuous, so by Lemma C.3, $f^{**} = f$. Conversely, assume that $f^{**} = f$. Then, by Theorem C.1, f is convex and lower semicontinuous. □

Note that the Fenchel-Moreau theorem implies that the LF transform f^* of f is involutive regardless of the shape of f, which gives the following overall structure:

$$f \to f^* \rightleftharpoons f^{**}$$

where the arrows stand for LF transform. Of course, for convex lower semicontinuous functions, the diagram reduces to

$$f \rightleftharpoons f^*.$$

C.2 The geometry of the LF transform

Let us first introduce the subdifferential commonly encountered in convex analysis.

Definition C.6. *Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a proper convex function. A vector v is called a subgradient of f at the point x_0 if, for all $x \in \mathbb{R}^n$,*

$$f(x) \geq f(x_0) + \langle v, x - x_0 \rangle.$$

The set of subgradients of f at x_0 is called the subdifferential of f at x_0, and is denoted $\partial f(x_0)$. If $\partial f(x_0) \neq \emptyset$ we say that f is subdifferentiable at x_0.

The subgradient v of f at x_0 has the following geometric meaning if $f(x_0) < \infty$: The graph of the affine function $h(x) = f(x_0) + \langle v, x - x_0 \rangle$ is a non-vertical supporting hyperplane to epi(f) at $(x_0, f(x_0))$. Clearly, if f is differentiable at x_0, $\partial f(x_0) = \nabla f(x_0)$. In fact, the subgradient fills much the same function as the gradient of a differentiable function. For example, it is clear from the definition that if $0 \in \partial f(x_0)$, then $f(x) \geq f(x_0) + \langle 0, x - x_0 \rangle = f(x_0)$ for all $x \in R^n$. Hence, $f(x_0)$ is a global minimum of f. It is also simple to prove the converse; if $f(x_0)$ is a global minimum of f, then $0 \in \partial f(x_0)$.

There are several nice connections between subgradients and the LF transform. Initially, if we have equality in the Fenchel-Young inequality, k is a subgradient to f, as is shown in the next theorem.

Theorem C.5. *Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be convex and proper. Then for any $x_0 \in domf$, the vector k is a subgradient to f at x_0 if and only if*

$$f(x_0) + f^*(k) = \langle x, k \rangle.$$

Proof. Assume first that $k \in \partial f(x_0)$. Then by the definition of subgradient, $f(x) \geq f(x_0) + \langle k, x - x_0 \rangle = \langle k, x \rangle - \langle k, x_0 \rangle$ so that $f(x_0) + \langle k, x \rangle - f(x) \leq \langle k, x_0 \rangle$ for all $x \in \mathbb{R}^n$. But then

$$f(x_0) + \sup_{x \in \mathbb{R}^n} \{\langle k, x \rangle - f(x)\} = f(x_0) + f^*(k) \leq \langle k, x_0 \rangle.$$

The opposite inequality is just the Fenchel-Young inequality with $x = x_0$. So we get $f(x_0) + f^*(k) = \langle x, k\rangle$.

Conversely, assume $f(x_0)+f^*(k) = \langle x, k\rangle$. The Fenchel-Young inequality implies $f^*(k) \geq \langle x, k\rangle - f(x)$ for all $x \in \mathbb{R}^n$, so we get

$$\langle x_0, k\rangle = f(x_0) + f^*(k) \geq f(x_0) + \langle x, k\rangle - f(x),$$

which, noting that the inner product on a real vector space is symmetric, yields

$$f(x) \geq f(x_0) + \langle k, x - x_0\rangle.$$

In other words, k is a subgradient to f at x_0 which concludes the proof. □

The subgradient of f has an interesting connection to the subgradient of f^*.

Theorem C.6. *If $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is proper and convex, then for all $x_0, k_0 \in \mathbb{R}^n$ we have*

$$k_0 \in \partial f(x_0) \quad \textit{if and only if} \quad x_0 \in \partial f^*(k_0).$$

The meaning of the theorem can be summarized as "slope is transformed into position by the LF transform", that is, if f has a subgradient k_0 at x_0, then x_0 is a subgradient to f^* at k_0. To prove the theorem, we first establish a special case:

Lemma C.4. *If $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is proper and convex with $f(0) = 0$, then for all $k_0 \in \mathbb{R}^n$ we have*

$$k_0 \in \partial f(0) \quad \textit{if and only if} \quad 0 \in \partial f^*(k_0).$$

Proof. Let $k_0 \in \partial f(0)$. Then for all $x \in \mathbb{R}^n$,

$$f(x) \geq f(0) + \langle k_0, x - 0\rangle = \langle k_0, x\rangle,$$

implying that

$$0 \geq \sup_{x\in\mathbb{R}^n} \{\langle k_0, x\rangle - f(x)\} = f^*(k_0).$$

Moreover, for any $k \in \mathbb{R}^n$,

$$f^*(k) = \sup_{x\in\mathbb{R}^n} \{\langle k, x\rangle - f(x)\} \geq \langle k, 0\rangle - f(0) = 0$$

so $f^*(k_0)$ is the minimum of f^* which means that $f^*(k) \geq f^*(k_0)$ for any $k \in \mathbb{R}^n$. Adding $\langle 0, k - k_0\rangle = 0$ to the right-hand side yields

$$f^*(k) \geq f^*(k_0) + \langle 0, k - k_0\rangle$$

for all $k \in \mathbb{R}^n$. In other words, $0 \in \partial f^*(k_0)$.

Conversely, assume $0 \in \partial f^*(k_0)$ so that for all $k \in \mathbb{R}^n$ we have

$$f^*(k_0) \geq f^*(k_0) + \langle 0, k - k_0 \rangle,$$

implying

$$\langle 0, k_0 \rangle - f^*(k_0) \geq \langle 0, k \rangle - f^*(k)$$

for all $k \in \mathbb{R}^n$. Thus,

$$\sup_{k \in \mathbb{R}^n} \{\langle 0, k \rangle - f^*(k)\} = \langle 0, k_0 \rangle - f^*(k_0) = f^{**}(0) = f(0) = 0$$

since f^* is convex and lower semicontinuous by Theorem C.1. But then we have

$$\langle 0, k_0 \rangle = f^*(k_0) \geq \langle k_0, x \rangle - f(x)$$

for all $x \in \mathbb{R}^n$ by definition of the LF transform, which implies

$$f(x) \geq \langle k_0, x \rangle - \langle 0, k_0 \rangle = \langle k_0, x \rangle$$

so that $k_0 \in \partial f(0)$ which proves the lemma. □

Armed with this special case, we are ready to prove the full theorem.

Proof of Theorem C.6. Define $g(x) := f(x + x_0) - f(x_0)$. Note that $f(x_0) < \infty$ since f has a subgradient at x_0. Then $g(0) = 0$ and $\partial g(0) = \partial f(x_0)$ so by Lemma C.4, $k_0 \in \partial f(x_0)$ if and only if $0 \in \partial g^*(k_0)$. We now have to relate the subdifferentials of g^* to that of f^*. To this end note that

$$\begin{aligned} g^*(k) &= \sup_{x \in \mathbb{R}^n} \{\langle k, x \rangle - g(x)\} \\ &= \sup_{x \in \mathbb{R}^n} \{\langle k, x \rangle - f(x + x_0) + f(x_0)\} \\ &= \sup_{x \in \mathbb{R}^n} \{\langle k, x + x_0 \rangle - f(x + x_0) + f(x_0) - \langle k, x_0 \rangle\} \\ &= f^*(k) - \langle k, x_0 \rangle + f(x_0). \end{aligned}$$

Now, $0 \in \partial g^*(k_0)$ if and only if for all $k \in \mathbb{R}^n$, $g^*(k) \geq g^*(k_0)$, and using the above we see that this is equivalent to

$$f^*(k) - \langle k, x_0 \rangle + f(x_0) \geq f^*(k_0) - \langle k_0, x_0 \rangle + f(x_0).$$

Equivalently,

$$f^*(k) \geq f^*(k_0) + \langle x_0, k - k_0 \rangle$$

for all $k \in \mathbb{R}^n$. Hence, $0 \in \partial g^*(k_0)$ if and only if $x_0 \in f^*(k_0)$. This finishes the proof. □

References

1. Aliu, O. G., Imran, A., Imran, M. A. & Evans, B. A survey of self organisation in future cellular networks. *IEEE Communications Surveys & Tutorials* **15,** 336–361 (2012).
2. Partov, B., Leith, D. J. & Razavi, R. Utility fair optimization of antenna tilt angles in LTE networks. *IEEE/ACM Transactions On Networking* **23,** 175–185 (2014).
3. Farooq, H., Imran, A. & Jaber, M. *Ai empowered smart user association in lte relays hetnets* in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)* (2019), 1–6.
4. Dandanov, N., Al-Shatri, H., Klein, A. & Poulkov, V. Dynamic self-optimization of the antenna tilt for best trade-off between coverage and capacity in mobile networks. *Wireless Personal Communications* **92,** 251–278 (2017).
5. Balevi, E. & Andrews, J. G. Online antenna tuning in heterogeneous cellular networks with deep reinforcement learning. *IEEE Transactions on Cognitive Communications and Networking* **5,** 1113–1124 (2019).
6. Shafin, R. *et al.* Self-tuning sectorization: Deep reinforcement learning meets broadcast beam optimization. *IEEE Transactions on Wireless Communications* **19,** 4038–4053 (2020).
7. Galindo-Serrano, A. & Giupponi, L. Distributed Q-learning for aggregated interference control in cognitive radio networks. *IEEE Transactions on Vehicular Technology* **59,** 1823–1834 (2010).
8. Vannella, F., Jeong, J. & Proutiere, A. *Off-policy learning for remote electrical tilt optimization* in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)* (2020), 1–5.
9. Bouton, M. *et al.* Coordinated Reinforcement Learning for Optimizing Mobile Networks. *arXiv preprint arXiv:2109.15175* (2021).
10. Silver, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *nature* **529,** 484–489 (2016).
11. Kocsis, L. & Szepesvári, C. *Bandit based monte-carlo planning* in *European conference on machine learning* (2006), 282–293.
12. Kandasamy, K., Dasarathy, G., Poczos, B. & Schneider, J. The multi-fidelity multi-armed bandit. *Advances in neural information processing systems* **29** (2016).
13. Bubeck, S. & Cesa-Bianchi, N. Regret analysis of stochastic and non-stochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721* (2012).
14. Williams, D. *Probability with martingales* (Cambridge university press, 1991).

15. Akaike, H. in *Selected papers of hirotugu akaike* 199–213 (Springer, 1998).

16. Kullback, S. & Leibler, R. A. On information and sufficiency. *The annals of mathematical statistics* **22,** 79–86 (1951).

17. Botev, Z. I., Kroese, D. P., Rubinstein, R. Y. & L'Ecuyer, P. in *Handbook of statistics* 35–59 (Elsevier, 2013).

18. Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* **25,** 285–294 (1933).

19. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A. & Talwalkar, A. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research* **18,** 6765–6816 (2017).

20. Auer, P., Cesa-Bianchi, N. & Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning* **47,** 235–256 (2002).

21. Robbins, H. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society* **58,** 527–535 (1952).

22. Lai, T. L., Robbins, H., *et al.* Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* **6,** 4–22 (1985).

23. Bubeck, S. *Bandits Games and Clustering Foundations* Theses (Université des Sciences et Technologie de Lille - Lille I, June 2010). `https://tel.archives-ouvertes.fr/tel-00845565`.

24. Kandasamy, K., Dasarathy, G., Oliva, J., Schneider, J. & Poczos, B. Multifidelity gaussian process bandit optimisation. *Journal of Artificial Intelligence Research* **66,** 151–196 (2019).

25. Browne, C. B. *et al.* A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games* **4,** 1–43 (2012).

26. Kocsis, L., Szepesvári, C. & Willemson, J. Improved monte-carlo search. *Univ. Tartu, Estonia, Tech. Rep* **1** (2006).

27. Asplund, H., Johansson, M., Lundevall, M. & Jaldén, N. *A set of propagation models for site-specific predictions* in *12th European Conference on Antennas and Propagation (EuCAP 2018)* (2018), 1–5.

28. James, S., Konidaris, G. & Rosman, B. *An analysis of monte carlo tree search* in *Thirty-First AAAI Conference on Artificial Intelligence* (2017).

29. Silver, D. *et al.* A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* **362,** 1140–1144 (2018).

30. Rockafellar, R. T. in *Convex analysis* (Princeton university press, 2015).

31. Bazaraa, M. S., Sherali, H. D. & Shetty, C. M. *Nonlinear programming: theory and algorithms* (John Wiley & Sons, 2013).