On 2-Armed Gaussian Bandits and Optimization

William G. Macready
David H. Wolpert

SFI WORKING PAPER: 1996-03-009

SANTA FE INSTITUTE

On 2-armed Gaussian Bandits and Optimization

William G. Macready, David H. Wolpert

Santa Fe Institute
1399 Hyde Park Road
Santa Fe, NM, 87501

March 7, 1996

Abstract

We explore the 2-armed bandit with Gaussian payoffs as a theoretical model for optimization. We formulate the problem from a Bayesian perspective, and provide the optimal strategy for both 1 and 2 pulls. We present regions of parameter space where a greedy strategy is provably optimal. We also compare the greedy and optimal strategies to a genetic-algorithm-based strategy. In doing so we correct a previous error in the literature concerning the Gaussian bandit problem and the supposed optimality of genetic algorithms for this problem. Finally, we provide an analytically simple bandit model that is more directly applicable to optimization theory than the traditional bandit problem, and determine a near-optimal strategy for that model.

Contents

1 Introduction **1**

2 Bandits and Optimization **2**

3 Expected payoff for any strategy **3**
3.1 Notational preliminaries 3
3.2 Expected payoffs 5
3.2.1 The $\boldsymbol{m = 1}$ case 5
3.2.2 The $\boldsymbol{m = 2}$ case 6
3.2.3 Arbitrary $\boldsymbol{m}$ 7

4 Optimal strategies **7**
4.1 The $\boldsymbol{m = 1}$ case 8
4.2 The $\boldsymbol{m = 2}$ case 8
4.2.1 $\boldsymbol{\sigma_1 = \sigma_2}$ 9
4.2.2 $\boldsymbol{\sigma_1 \neq \sigma_2}$ 11
4.2.3 Evaluating the maximal expected second payoff 14

5 Optimality of the Greedy Algorithm **16**
5.1 Proof of optimality of greedy algorithm for $\boldsymbol{\sigma_1 = \sigma_2}$ 19
5.2 Other cases in which the greedy strategy is optimal 25

6 Previous analysis of bandits and genetic algorithms **28**

7 A better bandit **31**

8 Discussion and Conclusions **35**

1 Introduction

Multi-armed bandit problems are remarkably simple to state and yet deceptively difficult to solve. The problem consists in determining a strategy for sequential selections from $k \geq 2$ stochastic payoff processes so as to maximize total expected payoff over m selections. The strategy specifies which of the k processes to select for every set of partial history of selections together with their associated payoffs. Bandit problems have a long and rich history. First posed in the 1930's they have been used to model subjects as diverse as clinical trials in medicine [2], job search in economics [3], and optimization [4] in all its settings.

Our focus in this paper is on bandit problems as related to optimization. The optimization problem requires extremization of some figure of merit, such as cost minimization in operations research, energy minimization in physics, or fitness maximization in biology. In all cases the problem is to find an object in some search space which extremizes the figure of merit for that object. Bandit problems have been used in the theory of optimization to explore the balance between exploration and exploitation necessary in effective optimization. Exploration of the search space is important to identify regions in which good solutions may be found. Exploitation is also important to put the knowledge gained from exploration to use. This exploration/exploitation tradeoff is also the essence of the bandit problem. The bandit problem we consider here has also been used as a theoretical underpinning for genetic algorithms, a currently popular optimization technique.

We begin in Section 2 with an informal introduction to our bandit problem. We also discuss the connections between bandit problems and effective optimization focusing on the exploration/exploitation tradeoff. This background should be sufficient that readers unfamiliar with bandit problems should be able to understand our major results.

In Section 3 we formalize the bandit problem by defining strategies and expected total payoffs for these strategies. We adopt a Bayesian perspective that explicitly takes into account the effects of prior knowledge. The task of maximizing expected total payoff under a particular class of strategies is taken up in Section 4. We present complete solutions for one and two pulls and comment on the difficulty of determining optimal strategies for larger numbers of total pulls. Analytic results are complemented with numerical calculations whose results should aid one's intuition. Interestingly, we find the for some parameter settings the optimal strategy need never take account of the results from previous pulls.

Section 5 considers the optimality of a certain class of myopic (greedy, purely exploitive) strategies and shows that in many cases a simple greedy strategy is optimal. We determine regions of parameter space in which we have been able to prove that a greedy strategy is optimal. Some of these results apply to any bandit problem, not simply the Gaussian bandit problem that this paper concentrates on.

In Section 6 we review the relationship between our Gaussian bandit problem and the theory of genetic algorithms, a popular optimization method. We point out an error in previous analyses of this relationship and indicate how to correct it. Our findings call into question the relevance of bandit problems for the theory of genetic algorithms.

The difficulty of determining optimal strategies for Gaussian bandit problems calls into question the utility of the traditional Gaussian bandit problem as a model for optimization.

In light of this difficulty, we propose a new bandit problem that is simpler than the Gaussian one as well as more directly applicable to optimization. We formalize the new bandit problem and give a near optimal strategy. In Section 7 we conclude with a brief discussion of general issues and list directions for future work.

To ease the burden on the reader we have attempted to highlight the important sections and results. Major results are presented as theorems or lemmas and the background necessary to understand these results are presented in definitions. Section 4 is the lengthiest section but perhaps the least important. Readers not wishing to wade through the mathematical details can simply note the major results. A skimming of this section will not impair understanding of later sections. The most important results are found in Sections 5 and 7. Section 5 address cases in which the simple greedy strategy is optimal while Section 7 formulates an improved bandit. Those interested in genetic algorithms will find Section 6 of interest, where errors are pointed out in Holland's derivation of exponential allocation of trials to the observed better arm.

2 Bandits and Optimization

In this paper we focus on a particularly simple version of the general bandit problem. Consider a 2-armed bandit where it is known that each arm has one or the other of two possible Gaussian payoff distributions. The parameters of each distribution, the means, μ_1, μ_2 and associated standard deviations σ_1, σ_2, are known. It is not known a priori which Gaussian goes with which arm however; rather we have a prior probability of which arm has which Gaussian.

We can imagine each distribution describing the behavior of a one-armed bandit slot machine. The "payoff" for" pulling" a particular arm is a random number drawn from the (unknown to us) probability distribution associated with that arm. We will make a total of m pulls, with our strategy dictating which arm to pull at pull i based on the results of the previous pulls and the (known) prior probability of which arm has which Gaussian. The goal in determining our strategy at pull i is to maximize total expected payoff over the remaining $m - i$ pulls. In general the payoffs may be discounted into the future; we consider both uniform and geometric discounting.

For a finite set of possible payoffs (rather than the uncountably infinite set of possible payoffs considered here) optimal strategies have been constructed using dynamic programming [5]. The case considered here appears to be significantly more difficult, however.

A generic characteristic of bandit problems is the tradeoff between, on the one hand, the need to gather information about which arm is which, and on the other hand the need to maximize payoff as quickly as possible. Often these two goals are in direct opposition. For example, we might gain a lot more information about which arm is which by knowingly pulling an arm we believe will have lower expected payoff. Then it may be that this extra information can be used in subsequent pulls to more than recoup our losses. If such a "more than recoup" result is the average result, it makes sense to pull this low-immediate-payoff arm.

Then again, it may be best just to maximize payoff over the short term; it may be that the expected gain accruing from our extra information in making a low-expected-payoff pull will never offset the associated expected loss. Determining the optimal tradeoff involves balancing the benefit of gathering information (exploration) versus that of maximizing short term payoff (exploitation).

This same tension between exploration and exploitation is also one of the central difficulties in optimization. In this kind of optimization the goal is to determine an algorithm (strategy) which efficiently locates global extrema of some mapping (sometimes called a "fitness" or "cost" function) given a finite total set of allowed samples of that mapping. Without loss of generality assume that we are seeking minima. Then an "exploitive" strategy is a greedy strategy each of whose moves in the search space always decreases the cost. Such greedy algorithms quickly locate a local minima but are then at a loss on how to proceed further. Imagine that instead a certain amount of "exploration" is used in the algorithm, so that some of the moves in the search space increase cost. In practice, although this exploration will delay convergence to local minima, often it will result in a lower final cost. So just like the bandit problem, optimization has a potential exploration-exploitation tradeoff.

This connection between the two kinds of problems has been noted many times previously. In perhaps the most well-known application of bandit problems to optimization, John Holland [4] uses ideas from the Gaussian bandit problem considered in this paper to (purportively) prove that genetic algorithm search methods are near "optimal". As part of our analysis of bandits and optimization we will return to this claim. First though, we define the bandit problem formally.

3 Expected payoff for any strategy

In this section we define the bandit problem formally and derive the expected payoff in m pulls as a function of any strategy for pulling the arms. Having found this expectation we can then maximize the expected payoff with respect to strategies to find the optimal strategy for pulling the arms; this is done in the section following this one.

3.1 Notational preliminaries

Given a strategy, for the ith pull, $i \in [1, 2, \ldots, m]$, we denote the arm chosen by the strategy as g_i and the resulting payoff as p_i. If we label the arms as α and β, then $\mathcal{P}$ denotes the prior probability that arm α has mean μ_1 and the associated standard deviation σ_1 (and consequently with probability $\mathcal{P}$ arm β has mean μ_2 and standard deviation σ_2).

The total payoff over the m pulls is $p = \sum_{i=1}^{m} \gamma_i p_i$ where the factors, γ_i determine the discounting. Most commonly, discounting is either uniform, $\gamma_i = 1$, or geometric, $\gamma_i = \gamma^{i-1}$. We will consider both types of discounting.

We compress notation by defining $\theta_1 \equiv (\mu_1, \sigma_1)$ and $\theta_2 \equiv (\mu_2, \sigma_2)$ and let θ be a vector denoting which set of parameters governs which arm. So θ can either be $\{\text{arm } \alpha = \theta_1, \text{arm } \beta = \theta_2\}$, or $\{\text{arm } \alpha = \theta_2, \text{arm } \beta = \theta_1\}$; in short, θ denotes the state of the two-armed bandit. To

further simplify notation, we will sometimes indicate the value of θ by indicating the state of arm α only. So for example, if a θ-valued random variable is said to have the value θ_1, we mean $\{\text{arm } \alpha = \theta_1, \text{arm } \beta = \theta_2\}$.

Also, to simplify the calculations we define the matrices

$$\mu \equiv \begin{bmatrix} \mu_1 & \mu_2 \\ \mu_2 & \mu_1 \end{bmatrix}, \quad \sigma \equiv \begin{bmatrix} \sigma_1 & \sigma_2 \\ \sigma_2 & \sigma_1 \end{bmatrix}.$$

We will use these matrices as shorthand in the following manner: The first row indicates arm α and the second row indicates arm β. The column indicates the value of θ; the first column indicates that arm α is governed by θ_1 (so arm β is governed by θ_2) and the second column indicates that arm α is governed by θ_2 (so arm β is governed by θ_1).

Usually a row value will be specified by a choice of an arm to pull. We let $g_i \in \{\alpha, \beta\}$ represent the arm selected by a strategy for the i pull. The rows of μ and σ are then indexed by the value of g_i. A column value will usually be given by a specification of θ. So $\mu_{g,\theta}$ and $\sigma_{g,\theta}$ indicate the (g, θ) matrix elements of μ and σ respectively. As an example, $\mu_{g_1=\alpha,\theta=(\theta_2,\theta_1)}$ is the mean payoff if the first pull is arm α and if that arm has distribution θ_2; it equals μ_2, the entry in the first row and second column of the matrix μ.

Note that in keeping with our shorthand for θ, we will sometimes indicate $\mu_{g_1=\alpha,\theta=(\theta_2,\theta_1)}$ (for example) by $\mu_{g_1=\alpha,\theta=\theta_2}$, or even $\mu_{\alpha,2}$ when we need to be as concise as possible. (The context should always make the precise meaning of any such abbreviated notation clear.) Note how this notation relates to our matrix notation: $\mu_{\alpha,2}$ is the entry in the first row and second column of the matrix μ. The prior probabilities are expressed as $P(\theta = \{\text{arm } \alpha = \theta_1, \text{arm } \beta = \theta_2\}) = \mathcal{P}$ and $P(\theta = \{\text{arm } \alpha = \theta_2, \text{arm } \beta = \theta_1\}) = 1 - \mathcal{P}$ in this shorthand notation are denoted as $P(\theta_1) = \mathcal{P}$ and $P(\theta_2) = 1 - \mathcal{P}$.

Strategies for pulling arms are specified as conditional probability distributions relating the arm pulled to information at hand. Our probability notation is standard. $P(e_1|e_2)$ is the probability of event e_1 occurring given that the value e_2 is known. $E(e_1|e_2)$ denotes the expected value of e_2 given e_1, and is related to $P(e_1|e_2)$ by $E(e_1|e_2) = \sum_{e_1} e_1 P(e_1|e_2)$.

With this notation, $P(g_1|\theta)$ is our strategy for the first pull; it is the probability that the first pull will be g_1 given the characteristics of the distributions. Since we are not allowed see θ when choosing g_1 (or any other g_i), g_1 is independent of θ, so $P(g_1|\theta) = P(g_1)$.

Similarly, the second pull is specified as $P(g_2|p_1, g_1, \theta)$. This distribution is conditioned on p_1 and g_1 since these values will certainly enter into any decision as to which arm to pull for the second pull. Again, this is independent of θ, so the strategy for the second pull is specified by $P(g_2|p_1, g_1, \theta) = P(g_2|p_1, g_1)$.

A complete strategy for m pulls is a specification of m such probability distributions, where $P(g_m|p_1, p_2, \ldots p_{m-1}, g_{m-1}, \theta) = P(g_m|p_1, p_2, \ldots p_{m-1}, g_{m-1})$ is again independent of θ. As an immediate consequence of this it follows that $P(\theta_\alpha|p_1, g_1, \ldots p_{m-1}, g_{m-1}, g_m) = P(\theta_\alpha|p_1, g_1, \ldots p_{m-1}, g_{m-1})$. (To see this use Bayes' theorem to invert the g_m and the $\{\theta_\alpha = \theta_1\}$ terms.)

As we shall soon see, optimal strategies are deterministic; the associated probability distributions are delta functions about some optimal pull. For example, the optimal second

pull can be written $P(g_2|g_1, p_1, \theta) = \delta(g_2 - G(p_1, g_1, \theta))$ for some single-valued function $G(\cdot)$. We will use this fact repeatedly in our determination of Bayes optimal strategies.

Using these notational conveniences we are now in a position to determine expected payoffs.

3.2 Expected payoffs

We provide detailed expressions for expected payoffs for $m = 1, 2$, and 3 pulls and leave the extension to arbitrary m to the reader.

3.2.1 The $m = 1$ case

We begin with the simplest case of a single pull, $m = 1$. We need to calculate $E(p|m = 1)$ the expected payoff of the first pull. We can easily calculate this by summing over all possible payoffs, p_1, guesses, g_1, and possibilities for θ. We have

$$\begin{aligned} E(p|m=1) &= \int dp_1 \sum_\theta \sum_{g_1} E(p, g_1, p_1, \theta|m=1) \\ &= \int dp_1 \sum_\theta \sum_{g_1} E(p|g_1, p_1, \theta, m=1) P(g_1, p_1, \theta|m=1) \\ &= \int dp_1 \sum_\theta \sum_{g_1} p_1 P(p_1|g_1, \theta, m=1) P(g_1|\theta, m=1) P(\theta|m=1) \\ &= \int dp_1 \sum_\theta \sum_{g_1} p_1 P(p_1|g_1, \theta) P(g_1) P(\theta) \end{aligned}$$

where the payoff probability is given by a Gaussian distribution. Using our shorthand matrix notation,

$$P(p_1|g_1, \theta) = \frac{1}{\sqrt{2\pi}\sigma_{g_1,\theta}} \exp[-(p_1 - \mu_{g_1,\theta})^2/2\sigma_{g_1,\theta}].$$

The expression for the expected payoff can be simplified somewhat with the following definition:

Definition 1 Define the function f_1 of the first guess

$$f_1(g_1) \equiv \int dp_1 \sum_\theta p_1 P(p_1|g_1, \theta) P(\theta) = \sum_\theta P(\theta) E(p_1|g_1, \theta) = \sum_\theta \mu_{g_1,\theta} P(\theta) \tag{1}$$

Lemma 1 *With the above definition the expected payoff $E(p|m = 1)$ for a single pull is*

$$E(p|m=1) = \sum_{g_1} P(g_1) f_1(g_1). \tag{2}$$

It is important to recognize that $P(g_1)$ is determined by the strategy while $f_1(g_1)$ is instead determined by the parameters of the problem ($f_1(g_1)$ can be interpreted as the expected payoff of arm g_1).

To find the optimal strategy for a single pull we must maximize Equation (2) with respect to $P(g_1)$. Because the expected payoff is a linear equation in the strategy it will clearly be maximized by the g_1 for which $f_1(g_1)$ achieves its maximum value. Consequently, $P(g_1|\theta) = \delta(g_1 - \text{argmax}_{g_1} f_1(g_1))$.

Lemma 2 *The maximum expected payoff for a single pull is given by*

$$E_{opt}(p|m=1) = \max_{g_1} f_1(g_1)$$

The determination of this value will be taken up in Section 4.

3.2.2 The $m = 2$ case

Next we turn to the case of two pulls, $m = 2$. The calculation proceeds as before but now we must also expand over the possible choices, g_2 and outcomes, p_2, of the second pull. In addition to determining the first pull, the strategy must now also specify the second pull, $P(g_2|p_1, g_1, \theta)$. We have noted the explicit dependence of the second pull on the results from the first pull. The expected total payoff, $p = p_1 + p_2$ can be written as

$$\begin{aligned}
E(p|m=2) &= \iint dp_1 dp_2 \sum_{g_1,g_2} \sum_{\theta} E(p|g_1, p_1, g_2, p_2, \theta, m=2) P(p_1, g_1, p_2, g_2, \theta|m=2) \\
&= \iint dp_1 dp_2 \sum_{g_1,g_2} \sum_{\theta} (p_1 + p_2) P(p_2|g_2, \theta) P(g_2|p_1, g_1, \theta) P(p_1|g_1, \theta) P(g_1) P(\theta) \\
&= \sum_{g_1} P(g_1) \Bigg[\int dp_1 \Bigg(p_1 \sum_{\theta} P(p_1|g_1, \theta) P(\theta) + \\
&\qquad \sum_{g_2} P(g_2|p_1, g_1) \int dp_2 \; p_2 \sum_{\theta} P(p_1|g_1, \theta) P(p_2|g_2, \theta) P(\theta) \Bigg) \Bigg]
\end{aligned}$$

Lemma 3 *The expected payoff $E(p|m=2)$ for two pulls is given by*

$$E(p|m=2) = \sum_{g_1} P(g_1) \Bigg(f_1(g_1) + \sum_{g_2} \int dp_1 P(g_2|p_1, g_1) f_2(g_2; p_1, g_1) \Bigg) \tag{3}$$

where $f_1(g_1)$ is defined as before and

Definition 2 we define the function f_2 of the second guess and results from the first pull as

$$f_2(g_2; p_1, g_1) \equiv \int dp_2 \, p_2 \sum_{\theta} P(p_1|g_1, \theta) P(p_2|g_2, \theta) P(\theta) = \sum_{\theta} \mu_{g_2,\theta} \, P(p_1|g_1, \theta) P(\theta) \tag{4}$$

Just as f_1 was related to the expected payoff on the first pull, $f_2(g_2; p_1, g_1)$ is related to the expected payoff from pulling arm g_2 on the second pull. It is easy to show that $\int dp_1 f_2(g_2; p_1, g_1) = E(p_2|g_1)$, a result we shall use later.

In this $m = 2$ case we must maximize Equation (3) with respect to both $P(g_1)$ and $P(g_2|p_1, g_1)$. Again, the optimal algorithm will be deterministic and

Lemma 4 *the maximum expected payoff $E_{opt}(p|m = 2)$ for two pulls is*

$$E_{opt}(p|m = 2) = \max_{g_1} \left(f_1(g_1) + \int dp_1 \max_{g_2} f_2(g_2; p_1, g_1) \right)$$

3.2.3 Arbitrary m

The calculations are analogous for larger values of m. For example, the result for $m = 3$ is

$$E(p|m = 3) = \sum_{g_1} P(g_1) \left[f_1(g_1) + \sum_{g_2} \int dp_1 P(g_2|p_1, g_1) \left(f_2(g_2; p_1, g_1) + \sum_{g_3} \int dp_2 P(g_3|p_1, g_1, p_2, g_2) f_3(g_3; p_1, g_1, p_2, g_2) \right) \right]$$

where

$$\begin{aligned} f_3(g_3; p_1, g_1, p_2, g_2) &\equiv \sum_{\theta} \int dp_3 \, p_3 P(p_3|g_3, \theta) P(p_2|g_2, \theta) P(p_1|g_1, \theta) P(\theta) \\ &= \sum_{\theta} \mu_{g_3,\theta} P(p_2|g_2, \theta) P(p_1|g_1, \theta) P(\theta) \end{aligned}$$

The extension to the case of general m is straightforward but tedious.

Note that for all $m > 1$, the optimal strategy for the first pull depends on the strategy-based decisions made for subsequent pulls. This is where the exploration-exploitation tradeoff (also important in optimization) comes into play. For example, if the expected payoff from later pulls rises sufficiently to offset the expected loss associated with choosing the first pull so as to *minimize* $f_1(g_1)$, then it behooves us to "explore" on the first pull rather than exploit. On the other hand, there is no such exploration-exploitation tradeoff for the final of m pulls, since there are no subsequent pulls; you should always exploit maximally on the last pull.

4 Optimal strategies

We now focus on optimizing the expected payoff $E(p|m)$ with respect to strategies to find the optimal pulling strategies. We present complete solutions for $m = 1$ and $m = 2$. Solutions for larger m are difficult (if not impossible) to obtain analytically because of the integrals involved. We begin with the simplest case.

4.1 The $m = 1$ case

Recall from Lemma 2 that $E(p|m=1)$ is maximized by the strategy of choosing g_1 so as to maximize $f_1(g_1)$. Recalling Equation (1) we have

$$\begin{aligned} f_1(g_1) &= \sum_\theta \mu_{g_1,\theta} P(\theta) = \mu_{g_1,\theta_1}\mathcal{P} + \mu_{g_2,\theta_2}(1-\mathcal{P}) \\ &= \mathcal{P}(\mu_{g_1,\theta_1} - \mu_{g_1,\theta_2}) + \mu_{g_1,\theta_2} \end{aligned}$$

To determine the g_1^{opt} which maximizes this quantity we consider the difference $f_1(g_1 = \alpha) - f_1(g_1 = \beta)$ and pick $g_1^{opt} = \alpha$ if the difference is positive or pick $g_1^{opt} = \beta$ if the difference is negative. The difference can be written as

$$\begin{aligned} \Delta f_1 &= \mathcal{P}(\mu_{\alpha,\theta_1} - \mu_{\alpha,\theta_2} - \mu_{\beta,\theta_1} + \mu_{\beta,\theta_2}) + (\mu_{\alpha,\theta_2} - \mu_{\beta,\theta_2}) \\ &= \mathcal{P}(\mu_1 - \mu_2 - \mu_2 + \mu_1) + (\mu_2 - \mu_1) \\ &= (2\mathcal{P} - 1)(\mu_1 - \mu_2) \end{aligned}$$

where it should be recalled that "$\mu_{a,b}$" is shorthand for "$\mu_{g_1=a,\theta=b}$".

Thus the optimal strategy for a single pull is to select arm α iff $\mathcal{P} > 1/2$ and $\mu_1 > \mu_2$ or $\mathcal{P} < 1/2$ and $\mu_1 < \mu_2$. Otherwise the optimal strategy picks arm β. Since $(\mathcal{P}(\mu_1 - \mu_2) + \mu_2) + (\mathcal{P}(\mu_2 - \mu_1) + \mu_1) = \mu_1 + \mu_2$ then $(f_1(\alpha) + f_1(\beta))/2 = (\mu_1 + \mu_2)/2$, and

Lemma 5 *the expected payoff for the optimal strategy for a single pull is*

$$E_{opt}(p|m=1) = \frac{\mu_1 + \mu_2}{2} + \left|\frac{\Delta f_1}{2}\right| = \frac{\mu_1 + \mu_2}{2} + \left|\left(\mathcal{P} - \frac{1}{2}\right)(\mu_1 - \mu_2)\right|. \tag{5}$$

4.2 The $m = 2$ case

Now we consider the more complicated case of $m = 2$. Our starting point is Equation (3),

$$E(p|m=2) = \sum_{g_1} P(g_1)\left(f_1(g_1) + \sum_{g_2}\int dp_1 P(g_2|p_1,g_1) f_2(g_2;p_1,g_1)\right)$$

which must be maximized with respect to both g_1 and g_2. We work backwards, first finding g_2^{opt} as a function of g_1 and then determining g_1^{opt}.

The optimal g_2 is that which maximizes $f_2(g_2;p_1,g_1)$. Recall from Definition 2 that

$$\begin{aligned} f_2(g_2;p_1,g_1) &= \sum_\theta \mu_{g_2,\theta}\, P(p_1|g_1,\theta)P(\theta) \\ &= \mu_{g_2,\theta_1} P(p_1|g_1,\theta_1)\mathcal{P} + \mu_{g_2,\theta_2} P(p_1|g_1,\theta_2)(1-\mathcal{P}) \end{aligned}$$

To determine g_2^{opt} we calculate the difference, $\Delta f_2(p_1,g_1) \equiv f_2(g_2 = \alpha;p_1,g_1) - f_2(g_2 = \beta;p_1,g_1)$:

$$\begin{aligned} \Delta f_2(p_1,g_1) &= (\mu_{\alpha,\theta_1} - \mu_{\beta,\theta_1})P(p_1|g_1,\theta_1)\mathcal{P} + (\mu_{\alpha,\theta_2} - \mu_{\beta,\theta_2})P(p_1|g_1,\theta_2)(1-\mathcal{P}) \\ &= (\mu_1 - \mu_2)\Big[\mathcal{P}P(p_1|g_1,\theta_1) + (\mathcal{P}-1)P(p_1|g_1,\theta_2)\Big] \end{aligned} \tag{6}$$

Since we choose $g_2 = \alpha$ if $\Delta f_2(p_1, g_1) > 0$ and $g_2 = \beta$ otherwise, it is important to know where in p_1-space $\Delta f_2(p_1, g_1)$ changes sign. Rearranging the equation $\Delta f_2(p_1 = z, g_1) = 0$ and then taking logarithms we find

$$\ln P(z|g_1, \theta_1) - \ln P(z|g_1, \theta_2) = \ln\Big[\frac{1-\mathcal{P}}{\mathcal{P}}\Big];$$
$$\frac{(z-\mu_{g_1,\theta_2})^2}{2\sigma^2_{g_1,\theta_2}} - \frac{(z-\mu_{g_1,\theta_1})^2}{2\sigma^2_{g_1,\theta_1}} = \ln\Big[\frac{(1-\mathcal{P})\sigma_{g_1,\theta_1}}{\mathcal{P}\sigma_{g_1,\theta_2}}\Big].$$

There are two cases to consider. It may be that both arms have the same standard deviation, $\sigma_{g_1,\theta_1} = \sigma_{g_1,\theta_2}$ or they may differ. In the case where they are equal the above equation becomes linear in z; otherwise we must solve a quadratic expression. We begin with the linear case.

4.2.1 $\sigma_1 = \sigma_2$

When $\sigma_1 = \sigma_2 \equiv \sigma$ we have a linear equation for z whose solution is $z = \rho_0$ where

$$\rho_0 = \frac{\sigma^2}{\mu_{g_1,\theta_1} - \mu_{g_1,\theta_2}} \ln\Big[\frac{1-\mathcal{P}}{\mathcal{P}}\Big] + \frac{\mu_{g_1,\theta_1} + \mu_{g_1,\theta_2}}{2}$$

We next need to determine on which side of $p_1 = \rho_0$ the quantity $\Delta f_2(p_1, g_1)$ is positive. This can be determined by examining $\lim_{p_1\to\infty} \Delta f_2(p_1, g_1)$, since $\Delta f_2(p_1, g_1)$ is linear in p_1. So we write

$$\Delta f_2(p_1, g_1) \propto (\mu_1 - \mu_2)\Big(\mathcal{P} - (1-\mathcal{P})\exp\left[-p_1(\mu_{g_1,\theta_1} - \mu_{g_1,\theta_2})/\sigma^2\right]\exp\left[(\mu^2_{g_1,1} - \mu^2_{g_1,2})/2\sigma^2\right]\Big). \tag{7}$$

If $\mu_{g_1,\theta_1} > \mu_{g_1,\theta_2}$ (for example if $g_1 = \alpha$ and $\mu_1 > \mu_2$) then the exponential kills off the $(1-\mathcal{P})$ term for large p_1, so the sign of $\Delta f_2(p_1 \to \infty, g_1)$ is given by the sign of $(\mu_1 - \mu_2)$. Conversely, if $\mu_{g_1,\theta_1} < \mu_{g_1,\theta_2}$, then the sign of $\Delta f_2(p_1 \to \infty, g_1)$ is given by the sign of $(\mu_2 - \mu_1)$.

So for example, if indeed $\mu_{g_1,\theta_1} > \mu_{g_1,\theta_2}$ and $\mu_1 > \mu_2$, and if $p_1 > \rho_0$, then $\Delta f_2(p_1, g_1) > 0$. In such a case, by the results of the previous subsection, we should choose $g_2 = \alpha$. More generally, denote the region region $p_1 < \rho_0$ by $R_<$ and the region $p_1 > \rho_0$ by $R_>$. If p_1 is in region $R_>$ then the optimal second pull is $g_2^{opt}(R_>) = \alpha$ if $(\mu_1 - \mu_2)(\mu_{g_1,\theta_1} - \mu_{g_1,\theta_2}) > 0$ and $g_2^{opt}(R_>) = \beta$ otherwise. But notice that $(\mu_1 - \mu_2)(\mu_{g_1,\theta_1} - \mu_{g_1,\theta_2}) = (\mu_1 - \mu_2)^2$ for $g_1 = \alpha$ and $-(\mu_1 - \mu_2)^2$ for $g_1 = \beta$. Thus we have the following result:

Lemma 6 *If p_1 is in region $R_>$, $g_2^{opt}(R_>) = g_1$ and if it lies in region $R_<$ the guesses are reversed (i.e., for p_1 in that region $g_2^{opt}(R_<) = \beta$ if $g_1 = \alpha$ and $g_2^{opt} = \alpha$ if $g_1 = \beta$).*

Given ρ_0 and g_2^{opt} we can determine the expected payoff for the optimal second guess.

Recalling Definition 2 of f_2, we see that this payoff is

$$\begin{aligned}
F_2(g_1) &\equiv \int dp_1 \max_{g_2} f_2(g_2; p_1, g_1) \\
&= \int_{-\infty}^{\infty} \frac{dp_1}{\sqrt{2\pi}\sigma} \Big(\mu_{g_2^{opt},\theta_1} \mathcal{P} \exp[-(p_1 - \mu_{g_1,\theta_1})^2/2\sigma^2] + \mu_{g_2^{opt},\theta_2} (1-\mathcal{P}) \exp[-(p_1 - \mu_{g_1,\theta_2})^2/2\sigma^2] \Big) \\
&= \int_{-\infty}^{\rho_0} \frac{dp_1}{\sqrt{2\pi}\sigma} \Big(\mu_{g_2^{opt}(R_<),\theta_1} \mathcal{P} \exp[-(p_1 - \mu_{g_1,\theta_1})^2/2\sigma^2] + \\
&\qquad \mu_{g_2^{opt}(R_<),\theta_2} (1-\mathcal{P}) \exp[-(p_1 - \mu_{g_1,\theta_2})^2/2\sigma^2] \Big) + \\
&\quad \int_{\rho_0}^{\infty} \frac{dp_1}{\sqrt{2\pi}\sigma} \Big(\mu_{g_2^{opt}(R_>),\theta_1} \mathcal{P} \exp[-(p_1 - \mu_{g_1,\theta_1})^2/2\sigma^2] + \\
&\qquad \mu_{g_2^{opt}(R_>),\theta_2} (1-\mathcal{P}) \exp[-(p_1 - \mu_{g_1,\theta_2})^2/2\sigma^2] \Big)
\end{aligned}$$

This can be simplified somewhat in terms of the complimentary error function. From the definition

$$\mathrm{erfc}(p) = \frac{2}{\sqrt{\pi}} \int_p^{\infty} dt\, \mathrm{e}^{-t^2}$$

we find that

$$\int_{-\infty}^{\rho_0} dp_1 \frac{\exp -(p-a)^2/2b^2}{\sqrt{2\pi}b} = \frac{1}{2} \mathrm{erfc}\left[-\frac{\rho_0 - a}{\sqrt{2}b} \right]$$

and

$$\int_{\rho_0}^{\infty} dp_1 \frac{\exp -(p-a)^2/2b^2}{\sqrt{2\pi}b} = \frac{1}{2} \mathrm{erfc}\left[\frac{\rho_0 - a}{\sqrt{2}b} \right]$$

So in terms of complimentary error functions the expected optimal payoff on the second pull is

$$\begin{aligned}
F_2(g_1) = &\frac{\mu_{g_2^{opt}(R_<),\theta_1}}{2} \mathcal{P}\, \mathrm{erfc}\left[-\frac{\rho_0 - \mu_{g_1,\theta_1}}{\sqrt{2}\sigma} \right] + \frac{\mu_{g_2^{opt}(R_<),\theta_2}}{2} (1-\mathcal{P})\, \mathrm{erfc}\left[-\frac{\rho_0 - \mu_{g_1,\theta_2}}{\sqrt{2}\sigma} \right] + \\
&\frac{\mu_{g_2^{opt}(R_>),\theta_1}}{2} \mathcal{P}\, \mathrm{erfc}\left[\frac{\rho_0 - \mu_{g_1,\theta_1}}{\sqrt{2}\sigma} \right] + \frac{\mu_{g_2^{opt}(R_>),\theta_2}}{2} (1-\mathcal{P})\, \mathrm{erfc}\left[\frac{\rho_0 - \mu_{g_1,\theta_2}}{\sqrt{2}\sigma} \right]
\end{aligned}$$

To determine the optimal first pull we must now maximize $f_1(g_1) + F_2(g_1)$ with respect to g_1. We can do this by considering the difference $\Delta f_1 + \Delta F_2 = f_1(\alpha) - f_1(\beta) + F_2(\alpha) - F_2(\beta)$. Recalling that $\Delta f_1 = (2\mathcal{P} - 1)(\mu_1 - \mu_2)$, we see that if $\Delta F_2 > (1 - 2\mathcal{P})(\mu_1 - \mu_2)$, then $g_1^{opt} = \alpha$, while if $\Delta F_2 < (1 - 2\mathcal{P})(\mu_1 - \mu_2)$, it follows that $g_1^{opt} = \beta$.

4.2.2 $\sigma_1 \neq \sigma_2$

As mentioned above, when the standard deviations are unequal we must solve a quadratic expression. The solutions are $z = \rho_\pm(g_1)$ where

$$\rho_\pm = \frac{\mu_{g_1,\theta_2}\sigma^2_{g_1,\theta_1} - \mu_{g_1,\theta_1}\sigma^2_{g_1,\theta_2} \pm \sigma_{g_1,\theta_1}\sigma_{g_1,\theta_2}\sqrt{(\mu_{g_1,\theta_1} - \mu_{g_1,\theta_2})^2 + 2(\sigma^2_{g_1,\theta_1} - \sigma^2_{g_1,\theta_2})\ln\left[\frac{(1-\mathcal{P})\sigma_{g_1,\theta_1}}{\mathcal{P}\sigma_{g_1,\theta_2}}\right]}}{\sigma^2_{g_1,\theta_1} - \sigma^2_{g_1,\theta_2}}$$

For completeness we note a symmetry of our formula for the roots, $\rho_\pm$, namely, $\rho_\pm(\alpha, \mathcal{P}) = \rho_\mp(\beta, 1-\mathcal{P})$. This reflects the observation that $\alpha \leftrightarrow \beta$ and $\mathcal{P} \leftrightarrow 1-\mathcal{P}$ just amounts to a relabeling of the arms. We also note that $\sigma_{g_1,\theta_1} < \sigma_{g_1,\theta_2}$ iff $\rho_+ < \rho_-$.

Henceforth, ρ_+ is used to denote the larger of the two roots and ρ_- the smaller of the two roots whether or not they are obtained from the $+$ or $-$ sign in the above equation.

Depending on the value of the discriminant there may be either 0, 1, or 2 distinct roots. We consider each possibility in turn.

The no roots case

If we have 0 roots then the optimal second guess, g_2^{opt} doesn't depend up the first payoff. This situation occurs if the discriminant is negative, *i.e.*

$$(\sigma^2_{g_1,\theta_1} - \sigma^2_{g_1,\theta_2})\ln\left[\frac{(1-\mathcal{P})\sigma_{g_1,\theta_1}}{\mathcal{P}\sigma_{g_1,\theta_2}}\right] < -\frac{(\mu_{g_1,\theta_1} - \mu_{g_1,\theta_2})^2}{2}$$

If $\sigma_{g_1,\theta_1} > \sigma_{g_1,\theta_2}$, then since $(\mu_1 - \mu_2)^2 = (\mu_2 - \mu_1)^2$, this condition is met if

$$\mathcal{P} > \frac{\sigma_{g_1,\theta_1}}{\sigma_{g_1,\theta_2}\exp\left[-(\mu_1 - \mu_2)^2/2(\sigma^2_{g_1,\theta_1} - \sigma^2_{g_1,\theta_2})\right] + \sigma_{g_1,\theta_1}}$$

while if $\sigma_{g_1,\theta_1} < \sigma_{g_1,\theta_2}$ we require

$$\mathcal{P} < \frac{\sigma_{g_1,\theta_1}}{\sigma_{g_1,\theta_2}\exp\left[(\mu_1 - \mu_2)^2/2(\sigma^2_{g_1,\theta_2} - \sigma^2_{g_1,\theta_1})\right] + \sigma_{g_1,\theta_1}}$$

A qualitative picture of these two situations is given in Figure 1. In these situations the optimal second pull is given in a fixed way by the sign of $\Delta f_2(p_1, g_1)$, regardless of the value of p_1 at which $\Delta f_2(p_1, g_1)$ is evaluated (see Equation 6). These are the situations where the prior $\mathcal{P}$ is sufficiently strong so that no result from the first pull can alter the second guess.

The single root case

For there to be a single distinct root the discriminant must be zero. In this case $\Delta f_2(p_1, g_1)$ has the same sign for all p_1 except one, for which it equals zero. As usual, the optimal second pull is determined by the sign of $\Delta f_2(p_1, g_1)$. To determine this sign we write $\Delta f_2 = (\mu_1 - \mu_2)(T_1 - T_2)$ where both $T_1 \equiv \mathcal{P}P(p_1|g_1, \theta_1)$ and $T_2 \equiv (1-\mathcal{P})P(p_1|g_1, \theta_2)$ are positive (see Equation 6).

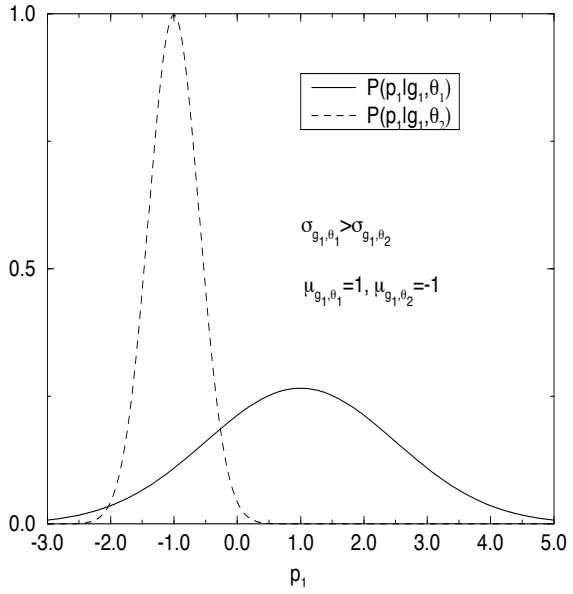

Figure 1: An example of the case where the optimal second guess g_2^{opt} is independent of the first payoff p_1. If the higher of the two μ's also has a large standard deviation, and our prior strongly biases our belief concerning which arm has that higher μ, then regardless of the first payoff the optimal strategy remains to choose that arm we believe to have the higher μ. (Intuitively, no value of p_1 can give strong evidence against the prior belief of which is the better arm, since any value of p_1 is relatively compatible with that belief, due to the better arm's large σ.) This can occur in two ways: (1) the variance of θ_1 is large and $\mathcal{P}$ is large, (2) the variance of θ_2 is large and $\mathcal{P}$ is small.

Given μ_1 and μ_2, the sign of Δf_2 is determined by the ratio T_1/T_2. Since Δf_2 has the same sign for all values of p_1 we can determine the sign by evaluating the ratio for any convenient p_1. In particular, in the limit $p_1 \to \infty$ we find

$$\lim_{p_1\to\infty} \frac{T_1}{T_2} = \frac{\mathcal{P}}{1-\mathcal{P}} \lim_{p_1\to\infty} \mathrm{e}^{-p_1^2(\sigma^2_{g_1,\theta_2} - \sigma^2_{g_1,\theta_1})/2\sigma^2_{g_1,\theta_1}\sigma^2_{g_1,\theta_2}}$$

If $\sigma_{g_1,\theta_1} < \sigma_{g_1,\theta_2}$ then this ratio is zero and so $T_1 - T_2 < 0$. Otherwise the ratio is ∞, which means that $T_1 - T_2 > 0$.

Definition 3 We define the function κ of the first guess and characteristics of the Gaussian distributions as

$$\kappa(g_1) = (\mu_1 - \mu_2)(\sigma_{g_1,\theta_1} - \sigma_{g_1,\theta_2}) \tag{8}$$

With this definition of $\kappa(g_1)$ we see that the optimal second pull is given by $g_2^{opt} = \alpha$ if $\kappa(g_1) > 0$ and $g_2^{opt} = \beta$ if $\kappa(g_1) < 0$. For the particular p_1 that gives $\Delta f_2(p_1, g_1) = 0$ it doesn't matter which arm we pick for the next guess since each has the same expected payoff.

The two roots case

When there are two distinct roots there are three regions with different optimal second guesses. We label these regions as $R_<$ for $p_1 \le \rho_-$, R_0 for $\rho_- < p_1 \le \rho_+$, and $R_>$ for $\rho_+ < p_1$. The optimal guess will change across each of the two boundaries, so determining the optimal second guess in any single region will determine the optimal guess in all other regions[1]. Consequently, it suffices to consider behavior in the region $R_>$ as p_1 gets very large. Just as in the case for a single root, we see that g_2^{opt} is determined solely by the sign of $\kappa(g_1)$:

$R_<$:

$$g_2^{opt}(R_<) = \begin{cases} \alpha & \text{if} \quad \kappa(g_1) > 0 \\ \beta & \text{if} \quad \kappa(g_1) < 0 \end{cases}$$

R_0:

$$g_2^{opt}(R_0) = \begin{cases} \alpha & \text{if} \quad \kappa(g_1) < 0 \\ \beta & \text{if} \quad \kappa(g_1) > 0 \end{cases}$$

$R_>$:

$$g_2^{opt}(R_>) = \begin{cases} \alpha & \text{if} \quad \kappa(g_1) > 0 \\ \beta & \text{if} \quad \kappa(g_1) < 0 \end{cases}$$

[1] Again, if p_1 lies on a boundary then the next pull doesn't matter since both arms have the same expected payoff

4.2.3 Evaluating the maximal expected second payoff

Having defined g_2^{opt} for all θ_1, θ_2, p_1, and g_1 we can now evaluate the maximal expected second payoff occuring for $g_2 = g_2^{opt}$. In the case where there are 0 or 1 roots g_2^{opt} doesn't depend on p_1, so we have

$$\begin{aligned} F_2(g_1) &\equiv \int_{-\infty}^{\infty} dp_1 \max_{g_2} f_2(g_2; p_1, g_1) \\ &= \int_{-\infty}^{\infty} \frac{dp_1}{\sqrt{2\pi}} \left(\mu_{g_2^{opt},\theta_1} \mathcal{P} \frac{\exp[-(p_1 - \mu_{g_1,\theta_1})^2/2\sigma_{g_1,\theta_1}^2]}{\sigma_{g_1,\theta_1}} + \mu_{g_2^{opt},\theta_2}(1-\mathcal{P}) \frac{\exp[-(p_1 - \mu_{g_1,\theta_2})^2/2\sigma_{g_1,\theta_2}^2]}{\sigma_{g_1,\theta_2}} \right) \\ &= \mathcal{P}\mu_{g_2^{opt},\theta_1} + (1-\mathcal{P})\mu_{g_2^{opt},\theta_2} \end{aligned}$$

where g_2^{opt} is given in the preceding section.

In the case where there are 2 roots and g_2^{opt} depends upon p_1 the situation is more complicated. Then we must divide up the integration over p_1 into the three regions, $R_<$, R_0, $R_>$:

$$\begin{aligned} F_2(g_1) = {} & \mu_{g_2^{opt}(R_<),\theta_1} \mathcal{P} \int_{-\infty}^{\rho_-} \frac{dp_1}{\sqrt{2\pi}\sigma_{g_1,\theta_1}} \exp\left[-(p_1 - \mu_{g_1,\theta_1})^2/2\sigma_{g_1,\theta_1}^2\right] + \\ & \mu_{g_2^{opt}(R_<),\theta_2}(1-\mathcal{P}) \int_{-\infty}^{\rho_-} \frac{dp_1}{\sqrt{2\pi}\sigma_{g_1,\theta_2}} \exp\left[-(p_1 - \mu_{g_1,\theta_2})^2/2\sigma_{g_1,\theta_2}^2\right] \\ & + \mu_{g_2^{opt}(R_0),\theta_1} \mathcal{P} \int_{-\rho_-}^{\rho_+} \frac{dp_1}{\sqrt{2\pi}\sigma_{g_1,\theta_1}} \exp\left[-(p_1 - \mu_{g_1,\theta_1})^2/2\sigma_{g_1,\theta_1}^2\right] + \\ & \mu_{g_2^{opt}(R_0),\theta_2}(1-\mathcal{P}) \int_{\rho_-}^{\rho_+} \frac{dp_1}{\sqrt{2\pi}\sigma_{g_1,\theta_2}} \exp\left[-(p_1 - \mu_{g_1,\theta_2})^2/2\sigma_{g_1,\theta_2}^2\right] \\ & + \mu_{g_2^{opt}(R_>),\theta_1} \mathcal{P} \int_{\rho_+}^{\infty} \frac{dp_1}{\sqrt{2\pi}\sigma_{g_1,\theta_1}} \exp\left[-(p_1 - \mu_{g_1,\theta_1})^2/2\sigma_{g_1,\theta_1}^2\right] + \\ & \mu_{g_2^{opt}(R_>),\theta_2}(1-\mathcal{P}) \int_{\rho_+}^{\infty} \frac{dp_1}{\sqrt{2\pi}\sigma_{g_1,\theta_2}} \exp\left[-(p_1 - \mu_{g_1,\theta_2})^2/2\sigma_{g_1,\theta_2}^2\right] \end{aligned}$$

Recalling the definition of the complimentary error function, erfc, we find

$$\begin{aligned} \int_{-\infty}^{\rho_-} dp_1 \frac{\exp\left(-(p_1-a)^2/2b^2\right)}{\sqrt{2\pi}b} &= \frac{1}{2}\operatorname{erfc}\left[-\frac{\rho_- - a}{\sqrt{2}b}\right] \\ \int_{\rho_-}^{\rho_+} dp_1 \frac{\exp\left(-(p_1-a)^2/2b^2\right)}{\sqrt{2\pi}b} &= \frac{1}{2}\left(\operatorname{erfc}\left[-\frac{\rho_+ - a}{\sqrt{2}b}\right] - \operatorname{erfc}\left[-\frac{\rho_- - a}{\sqrt{2}b}\right]\right) \\ \int_{\rho_+}^{\infty} dp_1 \frac{\exp\left(-(p_1-a)^2/2b^2\right)}{\sqrt{2\pi}b} &= \frac{1}{2}\operatorname{erfc}\left[\frac{\rho_+ - a}{\sqrt{2}b}\right] \end{aligned}$$

so that $F_2(g_1)$ can be expressed as

$$\begin{aligned}
2F_2(g_1) = {} & \mu_{g_2^{opt}(R_<),\theta_1}\mathcal{P}\ \mathrm{erfc}\left[-\frac{\rho_- - \mu_{g_1,\theta_1}}{\sqrt{2}\sigma_{g_1,\theta_1}}\right] + \mu_{g_2^{opt}(R_<),\theta_2}(1-\mathcal{P})\ \mathrm{erfc}\left[-\frac{\rho_- - \mu_{g_1,\theta_2}}{\sqrt{2}\sigma_{g_1,\theta_2}}\right] \\
& + \mu_{g_2^{opt}(R_0),\theta_1}\mathcal{P}\left(\mathrm{erfc}\left[-\frac{\rho_+ - \mu_{g_1,\theta_1}}{\sqrt{2}\sigma_{g_1,\theta_1}}\right] - \mathrm{erfc}\left[-\frac{\rho_- - \mu_{g_1,\theta_1}}{\sqrt{2}\sigma_{g_1,\theta_1}}\right]\right) \\
& + \mu_{g_2^{opt}(R_0),\theta_2}(1-\mathcal{P})\left(\mathrm{erfc}\left[-\frac{\rho_+ - \mu_{g_1,\theta_2}}{\sqrt{2}\sigma_{g_1,\theta_2}}\right] - \mathrm{erfc}\left[-\frac{\rho_- - \mu_{g_1,\theta_2}}{\sqrt{2}\sigma_{g_1,\theta_2}}\right]\right) \\
& + \mu_{g_2^{opt}(R_>),\theta_1}\mathcal{P}\ \mathrm{erfc}\left[\frac{\rho_+ - \mu_{g_1,\theta_1}}{\sqrt{2}\sigma_{g_1,\theta_1}}\right] + \mu_{g_2^{opt}(R_>),\theta_2}(1-\mathcal{P})\ \mathrm{erfc}\left[\frac{\rho_+ - \mu_{g_1,\theta_2}}{\sqrt{2}\sigma_{g_1,\theta_2}}\right]
\end{aligned}$$

This lengthy expression can be simplified by noting the following: $\mathrm{erfc}(z) + \mathrm{erfc}(-z) = 2$, and $\mu_{g_2^{opt}(R_>),\theta_1} = \mu_{g_2^{opt}(R_<),\theta_1}, \mu_{g_2^{opt}(R_>),\theta_2} = \mu_{g_2^{opt}(R_<),\theta_2}$, and for any a and b, $\mu_{a,\theta_2} - \mu_{b,\theta_2} = \mu_{b,\theta_1} - \mu_{a,\theta_1}$. With these simplifications we find

$$\begin{aligned}
F_2(g_1) = {} & \frac{\mu_{g_2^{opt}(R_<),\theta_1} - \mu_{g_2^{opt}(R_0),\theta_1}}{2}\left\{\mathcal{P}\left(\mathrm{erfc}\left[-\frac{\rho_- - \mu_{g_1,\theta_1}}{\sqrt{2}\sigma_{g_1,\theta_1}}\right] - \mathrm{erfc}\left[-\frac{\rho_+ - \mu_{g_1,\theta_1}}{\sqrt{2}\sigma_{g_1,\theta_1}}\right]\right) - \right. \\
& \left. (1-\mathcal{P})\left(\mathrm{erfc}\left[-\frac{\rho_- - \mu_{g_1,\theta_2}}{\sqrt{2}\sigma_{g_1,\theta_2}}\right] - \mathrm{erfc}\left[-\frac{\rho_+ - \mu_{g_1,\theta_2}}{\sqrt{2}\sigma_{g_1,\theta_2}}\right]\right)\right\} \\
& + \mathcal{P}\mu_{g_2^{opt}(R_<),\theta_1} + (1-\mathcal{P})\mu_{g_2^{opt}(R_0),\theta_1}
\end{aligned}$$

Having calculated both g_2^{opt} and $F_2(g_1)$ we are now in a position to determine g_1^{opt}. Recall from Lemma 4 that the optimal expected payoff is

$$E_{opt}(p|m=2) = \max_{g_1} f_1(g_1) + F_2(g_1)$$

To determine g_1^{opt} we calculate the difference $f_1(\alpha) - f_1(\beta) + F_2(\alpha) - F_2(\beta) = \Delta f_1 + \Delta F_2$. We have previously evaluated $\Delta f_1 = (2\mathcal{P}-1)(\mu_1 - \mu_2)$ in the $m=1$ case. Noting the explicit dependence of g_2^{opt} and $\rho_\pm$ on g_1 we find the difference ΔF_2 to be:

$$\begin{aligned}
& (\mu_{g_2^{opt}(R_<,\alpha),\theta_1} - \mu_{g_2^{opt}(R_<,\beta),\theta_1})\mathcal{P} + (\mu_{g_2^{opt}(R_0,\alpha),\theta_1} - \mu_{g_2^{opt}(R_0,\beta),\theta_1})(1-\mathcal{P}) \\
& + \frac{\mu_{g_2^{opt}(R_<,\alpha),\theta_1} - \mu_{g_2^{opt}(R_0,\alpha),\theta_1}}{2}\mathcal{P}\left(\mathrm{erfc}\left[-\frac{\rho_-(\alpha) - \mu_{\alpha,\theta_1}}{\sqrt{2}\sigma_{\alpha,\theta_1}}\right] - \mathrm{erfc}\left[-\frac{\rho_+(\alpha) - \mu_{\alpha,\theta_1}}{\sqrt{2}\sigma_{\alpha,\theta_1}}\right]\right) \\
& - \frac{\mu_{g_2^{opt}(R_<,\beta),\theta_1} - \mu_{g_2^{opt}(R_0,\beta),\theta_1}}{2}\mathcal{P}\left(\mathrm{erfc}\left[-\frac{\rho_-(\beta) - \mu_{\beta,\theta_1}}{\sqrt{2}\sigma_{\beta,\theta_1}}\right] - \mathrm{erfc}\left[-\frac{\rho_+(\beta) - \mu_{\beta,\theta_1}}{\sqrt{2}\sigma_{\beta,\theta_1}}\right]\right) \\
& - \frac{\mu_{g_2^{opt}(R_<,\alpha),\theta_1} - \mu_{g_2^{opt}(R_0,\alpha),\theta_1}}{2}(1-\mathcal{P})\left(\mathrm{erfc}\left[-\frac{\rho_-(\alpha) - \mu_{\alpha,\theta_2}}{\sqrt{2}\sigma_{\alpha,\theta_2}}\right] - \mathrm{erfc}\left[-\frac{\rho_+(\alpha) - \mu_{\alpha,\theta_2}}{\sqrt{2}\sigma_{\alpha,\theta_2}}\right]\right) \\
& + \frac{\mu_{g_2^{opt}(R_<,\beta),\theta_1} - \mu_{g_2^{opt}(R_0,\beta),\theta_1}}{2}(1-\mathcal{P})\left(\mathrm{erfc}\left[-\frac{\rho_-(\beta) - \mu_{\beta,\theta_2}}{\sqrt{2}\sigma_{\beta,\theta_2}}\right] - \mathrm{erfc}\left[-\frac{\rho_+(\beta) - \mu_{\beta,\theta_2}}{\sqrt{2}\sigma_{\beta,\theta_2}}\right]\right)
\end{aligned}$$

¿From the definition of g_2^{opt} in terms of $\kappa(g_1)$ we define

$$\tilde{g} \equiv \begin{cases} \alpha & \text{if} \quad (\mu_1 - \mu_2)(\sigma_1 - \sigma_2) > 0 \\ \beta & \text{if} \quad (\mu_1 - \mu_2)(\sigma_1 - \sigma_2) < 0 \end{cases}$$

Then in terms of $\tilde{g}$ we find the difference to be

$$\begin{aligned} \Delta F_2 =& (2\mathcal{P} - 1)(\mu_{\tilde{g},\theta_1} - \mu_{\tilde{g},\theta_2}) + \\ & (\mu_{\tilde{g},\theta_1} - \mu_{\tilde{g},\theta_2})\Bigg\{\mathcal{P}\Bigg(\operatorname{erfc}\left[-\frac{\rho_-(\alpha) - \mu_1}{\sqrt{2}\sigma_1}\right] + \operatorname{erfc}\left[-\frac{\rho_+(\beta) - \mu_2}{\sqrt{2}\sigma_2}\right] - \\ & \qquad \operatorname{erfc}\left[-\frac{\rho_+(\alpha) - \mu_1}{\sqrt{2}\sigma_1}\right] - \operatorname{erfc}\left[-\frac{\rho_-(\beta) - \mu_2}{\sqrt{2}\sigma_2}\right]\Bigg) + \\ & \qquad (1 - \mathcal{P})\Bigg(\operatorname{erfc}\left[-\frac{\rho_-(\alpha) - \mu_2}{\sqrt{2}\sigma_2}\right] + \operatorname{erfc}\left[-\frac{\rho_+(\beta) - \mu_1}{\sqrt{2}\sigma_1}\right] - \\ & \qquad \operatorname{erfc}\left[-\frac{\rho_+(\alpha) - \mu_2}{\sqrt{2}\sigma_2}\right] - \operatorname{erfc}\left[-\frac{\rho_-(\beta) - \mu_1}{\sqrt{2}\sigma_1}\right]\Bigg)\Bigg\} \end{aligned}$$

The sign of this difference determines the optimal first guess, g_1^{opt}. If $\Delta F_2 > -\Delta f_1$ then $g_1^{opt} = \alpha$ while if $\Delta F_2 < -\Delta f_1$ then $g_1^{opt} = \beta$.

Plots of the expected payoff for the optimal algorithm can be found in Figure 2 where $E(p|m = 2)$ for $\mu_1 = -1$, $\mu_2 = 1$ is plotted *versus* σ_1 and σ_2 for various $\mathcal{P}$.

Had we included a geometric discounting factor we would have to optimize the payoff $p = p_1 + \gamma p_2$. The optimal second guess would remain unaffected but g_1^{opt} would be modified to the g_1 which maximizes $f_1(g_1) + \gamma F_2(g_1)$. Consequently we would determine that guess by considering the difference $\Delta f_1 + \gamma \Delta F_2$. For some values of σ_1, σ_2, and $\mathcal{P}$ the optimal first pull may change as γ changes. To see this note that in Figure 3 the second expected payoff given g_1, $\Delta F_2(g_1)$ can assume either sign as a function of σ_1 and σ_2. In such scenarios, for certain $\gamma > 1$, g_1^{opt} may change from what it is for uniform discounting.

The difficulty of evaluating expected payoffs for the optimal algorithm increases exponentially with m. Solving this bandit problem for m pulls requires doing an m dimensional integral. Because of the Gaussian payoffs, the optimal guess for the ith pull will depend on which of three regions the previous $i-1$ dimensional payoff vector lies. Since integrands are over these optimal guesses, for m pulls there are 3^m regions to consider. While such integrals can be evaluated by Monte Carlo methods (at least for m not too large) we have not done so here.

5 Optimality of the Greedy Algorithm

Thompson [1] first posed the bandit problem and also suggested an often effective strategy — the "myopic" or "greedy" strategy. The greedy strategy differs from the optimal strategy in that g_1^{opt} is determined by maximizing $f_1(g_1)$ alone and then this value is used in $f_2(g_2; p_1, g_1^{opt})$

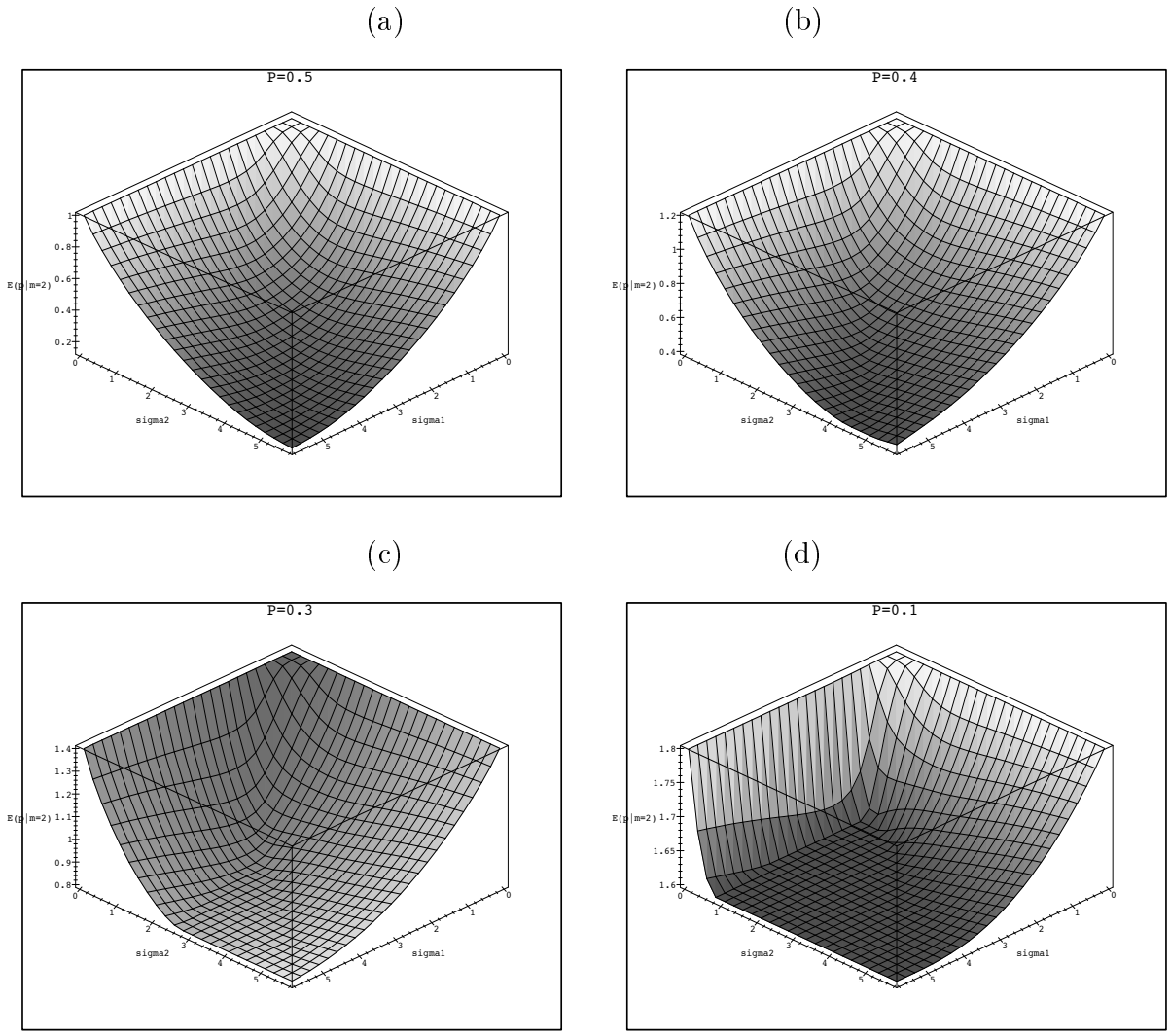

Figure 2: $E(p|m=2)$ for $(\mu_1, \mu_2) = (-1, 1)$ *vs* σ_1 and σ_2 for (a) $\mathcal{P} = 0.5$, (b) $\mathcal{P} = 0.4$, (c) $\mathcal{P} = 0.3$, and (d) $\mathcal{P} = 0.1$. $E(p|m=2)$ for $\mathcal{P} = 1/2 + \delta$ is identical to $\mathcal{P} = 1/2 - \delta$ since both reflect the same amount of certainty as to which arm has which θ.

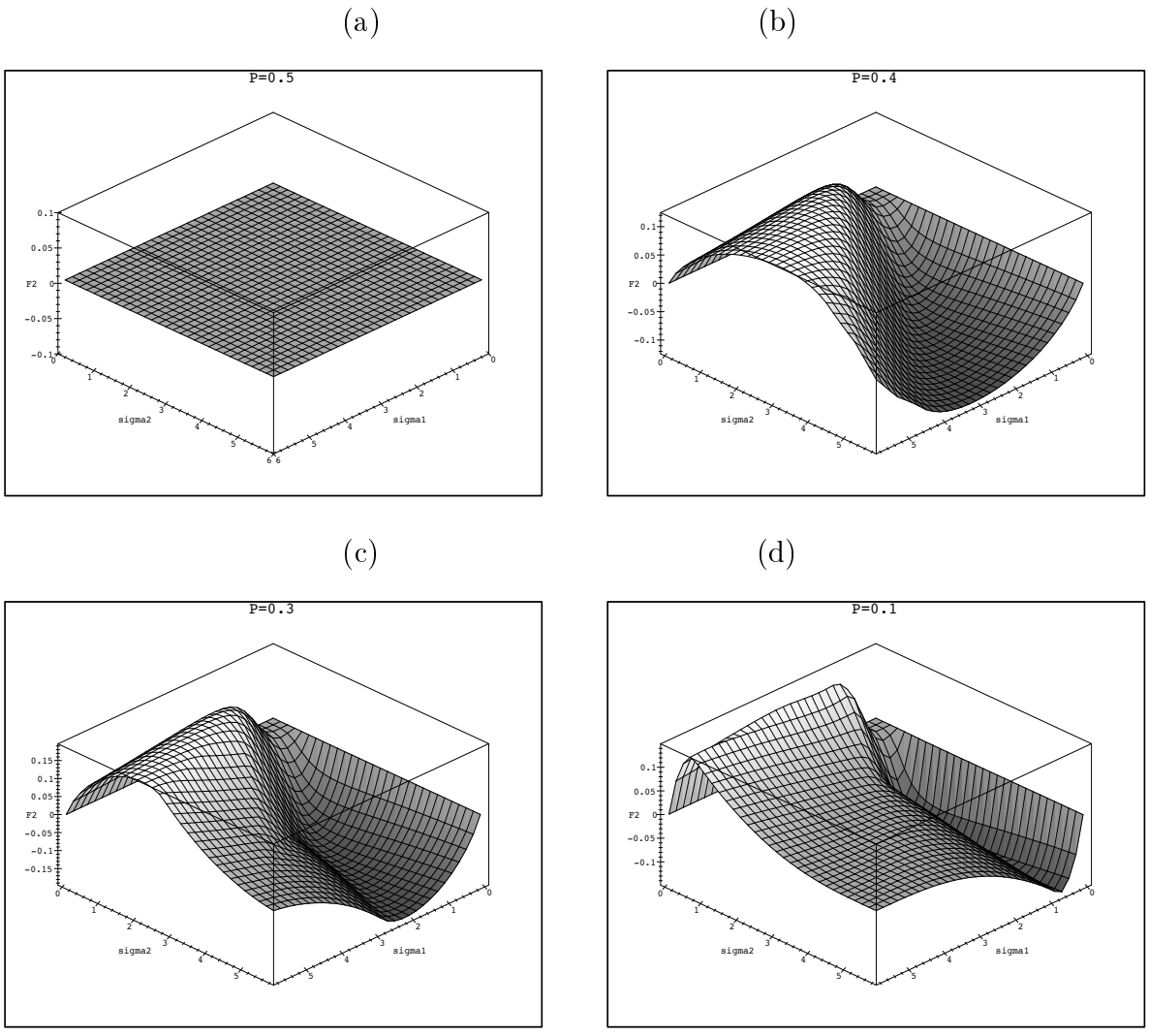

Figure 3: $\Delta F_2 = E(p_2|g_1 = \alpha) - E(p_2|g_1 = \beta)$ for $(\mu_1, \mu_2) = (-1, 1)$ *vs* σ_1 and σ_2 for (a) $\mathcal{P} = 0.5$, (b) $\mathcal{P} = 0.4$, (c) $\mathcal{P} = 0.3$, and (d) $\mathcal{P} = 0.1$. $E(p|m = 2)$ for $\mathcal{P} = 1/2 + \delta$ is identical to $\mathcal{P} = 1/2 - \delta$ since both reflect the same amount of certainty as to which arm has which θ.

to determine the optimal second pull. More generally, the greedy algorithm selects the arm which optimizes the payoff for the next pull, without any regard to performance on subsequent pulls. This is simple to do since the posterior probability that arm α is associated with distribution parameters θ_1 is straight-forward to calculate using Bayes' theorem.

For the $m = 2$ case discussed above the greedy algorithm will be optimal if the g_1 determined by a greedy method agrees with the optimal g_1. This is the case if $(\Delta f_1 + \Delta F_2)\Delta f_1 \geq 0$, which certainly holds if either $|\Delta F_2| \leq |\Delta f_1|$ or less interestingly if $\Delta F_2 > \Delta f_1$ and ΔF_2 and Δf_1 are both positive and . (See the end of Section 4.2.1.) Substituting $\Delta f_1 = (2\mathcal{P} - 1)(\mu_1 - \mu_2)$, we see that the greedy algorithm will be optimal if $|\Delta F_2| < |2\mathcal{P} - 1||\Delta\mu|$.

Numerically, this can be seen to hold for $m = 2$, and later we shall comment on how to prove this result. But is it the case that the greedy algorithm is in fact optimal for any m? In the case of Bernoulli payoffs it has been proven that the greedy strategy is optimal for all m [6]. This conjecture is least plausible. The task for the optimal strategy is mostly one of identification, to determine which distribution is associated with which arm. Moreover there are only two possibilities: either θ_1 is associated with arm α or with arm β. Because there are only these two cases, anything we learn about one arm is immediately transferable into information concerning the other arm. So there appears to be no reason to explore, (*i.e* no reason to pull an arm you think has lower μ) since you would learn as much on average from pulling the arm you think is better.

While we have been unable to prove that the greedy algorithm is optimal for all θ_1, θ_2 and $\mathcal{P}$ we have been able to prove it in a number of special cases. However the reader should bear in mind that it is certainly *not* true that the greedy algorithm is optimal for all discounting schedules. For example, we saw in Section 4.2.3 that even for $m = 2$, when $\gamma > 1$, so that the future is weighted more heavily, the greedy strategy is sub-optimal.

We first consider the case of uniform discounting when $\sigma_1 = \sigma_2$ and greedy can be proven to optimal.

5.1 Proof of optimality of greedy algorithm for $\sigma_1 = \sigma_2$

As always we consider m total pulls and imagine that there are $k + 1 > 2$ pulls remaining. The information on hand is the result of the previous $m - k - 1$ pulls, namely $\vec{p}_{m-k-1} \equiv \{p_1, \cdots, p_{m-k-1}\}$ and $\vec{g}_{m-k-1} \equiv \{g_1, \cdots, g_{m-k-1}\}$. Given this information we would like to know how the choice of the next pull, g_{m-k}, will affect subsequent payoffs. Consequently, we consider the sum of the expected payoffs p_j for the future pulls $j \in [m - k + 1, \ldots, m]$, conditioned on the information at hand and knowledge of our next pull:

$$E(\sum_j p_j \mid \vec{p}_{m-k-1}, \vec{g}_{m-k-1}, g_{m-k}) = E(\sum_j p_j \mid \vec{p}_{m-k-1}, \vec{g}_{m-k}), \tag{9}$$

where the restriction on the range of j is implicitly understood (j is at least two pulls beyond the ones whose resultant payoff we already know.)

If we can show that for the optimal algorithm this expectation is in fact independent of the next pull, g_{m-k}, this will mean that there is no reason not to pull the arm we think is

best at pull $m-k$, since expected subsequent payoff (recall $k > 1$) will not be affected by the choice of which arm to pull. To this end we make the following definition:

Definition 4 Define ω_{m-k} as the posterior probability after the $m-k-1$ pulls that arm α is associated with parameters θ_1:

$$\omega_{m-k} \equiv P(\theta_\alpha = \theta_1 | \vec{p}_{m-k-1}, \vec{g}_{m-k-1})$$

More formally, ω_{m-k} is a real-valued random variable which is a function of the random variables $\vec{p}_{m-k-1}$ and $\vec{g}_{m-k-1}$ (and of course of the parameters of the problem, $\mathcal{P}$, θ_1, and θ_2). The definition of ω_{m-k} presented above says how its value is set. As an example, in terms of our previous notation $\omega_1 = \mathcal{P}$.

Intuitively, ω_i is the "exploitable information" we have about the state of the bandit at the time we must decide on our i'th pull. Non-optimal algorithms do not always reflect this fact. Such algorithms can make different guesses g_i for two different sets of $\{\vec{p}_{i-1}, \vec{g}_{i-1}\}$ even if ω_i is the same for those two sets. The optimal algorithm can never do this. (This is established formally in the proof of the lemma below.)

Since the strategy subsequent to pull $m-k$ is optimal, knowing ω_{m-k+1} tells us all we need to know concerning $\vec{p}_{m-k}$ and $\vec{g}_{m-k}$ as far as calculating expected payoff for the pulls $j > m-k$ is concerned. Formally, the expectation value in Equation (9) can be written

Lemma 7 *The expected future payoff conditioned on the information available at the $(m-k)$'th pull is*

$$E(\sum_j p_j \mid \vec{p}_{m-k-1}, \vec{g}_{m-k}) = \int d\omega_{m-k+1} \, E(\sum_j p_j \mid \omega_{m-k+1}) P(\omega_{m-k+1} | \vec{p}_{m-k-1}, \vec{g}_{m-k}).$$

Proof: To prove this lemma it suffices to show that

$$E(\sum_j p_j | \omega_{m-k+1}, \vec{p}_{m-k-1}, \vec{g}_{m-k}) = E(\sum_j p_j | \omega_{m-k+1}).$$

We begin by expanding the left-hand side of this proposed equality as

$$\int dp_{m-k} \, E(\sum_j p_j \mid \omega_{m-k+1}, \vec{p}_{m-k}, \vec{g}_{m-k}) \, P(p_{m-k} | \omega_{m-k+1}, \vec{p}_{m-k-1}, \vec{g}_{m-k}).$$

Examining the integrand, it is clear that a sufficient condition for our proposed equality to hold is that

$$E(\sum_j p_j \mid \omega_{m-k+1}, \vec{p}_{m-k}, \vec{g}_{m-k}) = E(\sum_j p_j \mid \omega_{m-k+1})$$

for all p_{m-k} such that $\omega_{m-k+1} = P(\theta_\alpha = \theta_1 | \vec{p}_{m-k}, \vec{g}_{m-k})$.

It is this equality that we shall prove. First define $\vec{g}_{a;b\geq a} \equiv \{g_a, g_{a+1}, \ldots g_b\}$, and for completeness have $\vec{g}_{a;b<a}$ be the empty set. Define $\vec{p}_{a;b}$ similarly. Then we can expand

$$E(\sum_j p_j|\omega_{m-k+1}, \vec{p}_{m-k}, \vec{g}_{m-k}) = \sum_j \sum_\theta \int d\vec{p}_{m-k+1;j} \sum_{\vec{g}_{m-k+1;j}} p_j P(\theta, \vec{p}_{m-k+1;j}, \vec{g}_{m-k+1;j}|\omega_{m-k+1}, \vec{p}_{m-k}, \vec{g}_{m-k})$$

Now break up the last term in the integrand:

$$P(\theta, \vec{p}_{m-k+1;j-1}, \vec{g}_{m-k+1;j-1}|\omega_{m-k+1}, \vec{p}_{m-k}, \vec{g}_{m-k}) = P(\theta|\vec{p}_{m-k}, \vec{g}_{m-k}, \omega_{m-k+1}) \times P(\vec{p}_{m-k+1;j}, \vec{g}_{m-k+1;j}|\theta, \vec{p}_{m-k}, \vec{g}_{m-k}, \omega_{m-k+1})$$

Collecting terms, we get

$$E(\sum_j p_j \mid \omega_{m-k+1}, \vec{p}_{m-k}, \vec{g}_{m-k}) = \sum_\theta P(\theta|\vec{p}_{m-k}, \vec{g}_{m-k}, \omega_{m-k+1}) \sum_j \int d\vec{p}_{m-k+1;j} \sum_{\vec{g}_{m-k+1;j}} p_j P(\vec{p}_{m-k+1;j}, \vec{g}_{m-k+1;j}|\theta, \vec{p}_{m-k}, \vec{g}_{m-k}, \omega_{m-k+1})$$

To proceed expand the last term in our last integrand as

$$P(\vec{p}_{m-k+1;j}, \vec{g}_{m-k+1;j}|\theta, \vec{p}_{m-k}, \vec{g}_{m-k}, \omega_{m-k+1}) = \prod_{i=m-k+1}^{j} P(p_i|g_i, \theta) \prod_{i=m-k+1}^{j} P(g_i|\vec{p}_{i-1}, \vec{g}_{i-1}, \omega_{m-k+1})$$

Now define $\omega_{m-k+1}(\theta) \equiv \omega_{m-k+1}$ if θ corresponds to $\theta_\alpha = \theta_1$, and $\omega_{m-k+1}(\theta) \equiv 1 - \omega_{m-k+1}$ otherwise. Then $P(\theta|\vec{p}_{m-k}, \vec{g}_{m-k}, \omega_{m-k+1}) = \omega_{m-k+1}(\theta)$. So writing it all out, we have

$$E(\sum_j p_j|\omega_{m-k+1}, \vec{p}_{m-k}, \vec{g}_{m-k}) = \sum_\theta \omega_{m-k+1}(\theta) \times \sum_j \int dp_{m-k+1} \ldots \int dp_j \sum_{g_{m-k+1}} \ldots \sum_{g_j} p_j \prod_{i=m-k+1}^{j} P(p_i|g_i, \theta) P(g_i|\vec{p}_{i-1}, \vec{g}_{i-1})$$

Collecting terms common to more than j, this can be rewritten as

$$\begin{aligned} E(\sum_j p_j|\omega_{m-k+1}, \vec{p}_{m-k}, \vec{g}_{m-k}) = \sum_\theta \omega_{m-k+1}(\theta) \times \\ \sum_{g_{m-k+1}} P(g_{m-k+1}|\vec{p}_{m-k}, \vec{g}_{m-k}, \omega_{m-k+1}) \int dp_{m-k+1} P(p_{m-k+1}|g_{m-k+1}, \theta) \\ \Big[p_{m-k+1} + \sum_{g_{m-k+2}} P(g_{m-k+2}|\vec{p}_{m-k+1}, \vec{g}_{m-k+1}, \omega_{m-k+1}) \int dp_{m-k+2} P(p_{m-k+2}|g_{m-k+2}, \theta) \\ [p_{m-k+2} + \ldots \\ \vdots \\ \sum_{g_m} P(g_m|\vec{p}_{m-1}, \vec{g}_{m-1}, \omega_{m-k+1}) \int dp_m p_m P(p_m|g_m, \theta)] \ldots] \Big] \end{aligned} \tag{10}$$

As usual, the strategy—that which the arm-puller can vary—is the distribution $P(g_i|\vec{p}_{i-1}, \vec{g}_{i-1})$.

Now perform a similar decomposition of $E(\sum_j p_j \mid \omega_{m-k+1})$ to get

$$
\begin{aligned}
E(\sum_j p_j|\omega_{m-k+1}) = \int d\vec{p}_{m-k} \sum_{\vec{g}_{m-k}} P(\vec{p}_{m-k}, \vec{g}_{m-k}|\omega_{m-k+1}) \sum_\theta \omega_{m-k+1}(\theta) \times \\
\sum_{g_{m-k+1}} P(g_{m-k+1}|\vec{p}_{m-k}, \vec{g}_{m-k}, \omega_{m-k+1}) \int dp_{m-k+1} P(p_{m-k+1}|g_{m-k+1}, \theta) \\
\Big[p_{m-k+1} \; + \sum_{g_{m-k+2}} P(g_{m-k+2}|\vec{p}_{m-k+1}, \vec{g}_{m-k+1}, \omega_{m-k+1}) \int dp_{m-k+2} P(p_{m-k+2}|g_{m-k+2}, \theta) \\
[p_{m-k+2} \; + \; \ldots \\
\vdots \\
\sum_{g_m} P(g_m|\vec{p}_{m-1}, \vec{g}_{m-1}, \omega_{m-k+1}) \int dp_m p_m P(p_m|g_m, \theta)] \ldots]\Big] \qquad (11)
\end{aligned}
$$

Examine the last term in this sum,

$$
\begin{aligned}
\int d\vec{p}_{m-k} \sum_{\vec{g}_{m-k}} P(\vec{p}_{m-k}, \vec{g}_{m-k} \mid \omega_{m-k+1}) \\
\int dp_{m-k+1} \ldots \int dp_m \sum_{g_{m-k+1}} \ldots \sum_{g_m} p_m \prod_{i=m-k+1}^{m} P(g_i|\vec{p}_{i-1}, \vec{g}_{i-1}) \\
\sum_\theta [\; \omega_{m-k+1}(\theta) \prod_{i=m-k+1}^{m} P(p_i|g_i, \theta) \;] \\
\equiv \\
\int d\vec{p}_{m-k} \sum_{\vec{g}_{m-k}} P(\vec{p}_{m-k}, \vec{g}_{m-k} \mid \omega_{m-k+1}) \\
\int dp_{m-k+1} \ldots \int dp_m \sum_{g_{m-k+1}} \ldots \sum_{g_m} p_m \prod_{i=m-k+1}^{m} P(g_i|\vec{p}_{i-1}, \vec{g}_{i-1}) \\
F_m(\omega_{m-k+1}, \vec{p}_{m-k+1;m}, \vec{g}_{m-k+1;m})
\end{aligned}
$$

and rewrite this as

$$\int d\vec{p}_{m-k} \sum_{\vec{g}_{m-k}} P(\vec{p}_{m-k}, \vec{g}_{m-k} \mid \omega_{m-k+1})$$
$$\int dp_{m-k+1} \ldots \int dp_{m-1} \sum_{g_{m-k+1}} \ldots \sum_{g_{m-1}} \prod_{i=m-k+1}^{m-1} P(g_i|\vec{p}_{i-1}, \vec{g}_{i-1})$$
$$\sum_{g_m} P(g_m|\vec{p}_{m-1}, \vec{g}_{m-1}) \int dp_m p_m F_m(\omega_{m-k+1}, \vec{p}_{m-k+1;m}, \vec{g}_{m-k+1;m})$$
$$\equiv$$
$$\int d\vec{p}_{m-k} \sum_{\vec{g}_{m-k}} P(\vec{p}_{m-k}, \vec{g}_{m-k} \mid \omega_{m-k+1})$$
$$\int dp_{m-k+1} \ldots \int dp_{m-1} \sum_{g_{m-k+1}} \ldots \sum_{g_{m-1}} \prod_{i=m-k+1}^{m-1} P(g_i|\vec{p}_{i-1}, \vec{g}_{i-1})$$
$$\sum_{g_m} P(g_m|\vec{p}_{m-1}, \vec{g}_{m-1}) \hat{F}_m(\omega_{m-k+1}, \vec{p}_{m-k+1;m-1}, \vec{g}_{m-k+1;m})$$

By inspection, the optimal strategy for the m'th pull depends only on ω_{m-k+1}, $p_{m-k+1;m-1}$, and $g_{m-k+1;m-1}$. (That optimal strategy is to pull the arm g_m that maximizes the quantity $\hat{F}(\omega_{m-k+1}, \vec{p}_{m-k+1;m-1}, \vec{g}_{m-k+1;m})$.) Accordingly, we can rewrite $P(g_m|\vec{p}_{m-1}, \vec{g}_{m-1})$ as $P(g_m|\omega_{m-k+1}, \vec{p}_{m-k+1;m-1}, \vec{g}_{m-k+1;m-1})$. Note that this is independent of $\{\vec{p}_{m-k}, \vec{g}_{m-k}\}$ for all $\{\vec{p}_{m-k}, \vec{g}_{m-k}\}$ consistent with the specified value of ω_{m-k+1}.

We can now repeat the process, and examine the second-to-last term in the sum in Equation (11). Our function $F_{m-1}(\omega_{m-k+1}, \vec{p}_{m-k+1;m-1}, \vec{g}_{m-k+1;m-1})$ will now depend on the quantity $\max_{g_m} \hat{F}_m(\omega_{m-k+1}, \vec{p}_{m-k+1;m-1}, \vec{g}_{m-k+1;m})$. The end result will again be that the values of $\{\vec{p}_{m-k}, \vec{g}_{m-k}\}$ do not matter, so long as they are consistent with the specified value of ω_{m-k+1}:

$$P(g_{m-1}|\vec{p}_{m-2}, \vec{g}_{m-2}) = P(g_{m-1}|\omega_{m-k+1}, \vec{p}_{m-k+1;m-2}, \vec{g}_{m-k+1;m-2})$$

Continuing we see that the argument of the $\int d\vec{p}_{m-k} \sum_{\vec{g}_{m-k}} P(\vec{p}_{m-k}, \vec{g}_{m-k}|\omega_{m-k+1})$ in Eq. (11) is independent of the values $\{\vec{p}_{m-k}, \vec{g}_{m-k}\}$, so long as those values are consistent with the specified value of ω_{m-k+1}. Accordingly, that $\int d\vec{p}_{m-k} \sum_{\vec{g}_{m-k}} P(\vec{p}_{m-k}, \vec{g}_{m-k}|\omega_{m-k+1})$ evaluates to 1.

This means that the expressions in Eq. (11) and Eq. (10) are identical which completes the proof.

So to prove that our expectation value is independent of the next pull g_{m-k}, it suffices to prove that $P(\omega_{m-k+1}|\vec{p}_{m-k-1}, \vec{g}_{m-k-1}, g_{m-k})$ is independent of g_{m-k}.

Now we know that

$$P(\omega_{m-k+1}|\vec{p}_{m-k-1}, \vec{g}_{m-k-1}, g_{m-k}) = \int dp_{m-k}\, \delta\Big(\omega_{m-k+1} - P(\theta_\alpha = \theta_1|\vec{p}_{m-k}, \vec{g}_{m-k})\Big)$$
$$\times P(p_{m-k}|\vec{p}_{m-k-1}, \vec{g}_{m-k-1}, g_{m-k})$$

However by using Bayes' theorem, $P(\theta_\alpha = \theta_1 | \vec{p}_{m-k}, \vec{g}_{m-k})$ is easily calculated as

$$P(\theta_\alpha = \theta_1 | \vec{p}_{m-k}, \vec{g}_{m-k}) = \frac{P(\vec{p}_{m-k}, \vec{g}_{m-k} | \theta_\alpha = \theta_1) P(\theta_\alpha = \theta_1)}{\sum_{\theta_\alpha} P(\vec{p}_{m-k}, \vec{g}_{m-k} | \theta_\alpha = \theta_1) P(\theta_\alpha = \theta_1)}$$

Expanding, we can write

$$\begin{aligned} P(\vec{p}_{m-k}, \vec{g}_{m-k} | \theta_\alpha = \theta_1) = & P(p_{m-k} | \vec{p}_{m-k-1}, \vec{g}_{m-k}, \theta_\alpha = \theta_1) \times \\ & P(g_{m-k} | \vec{p}_{m-k-1}, \vec{g}_{m-k-1}, \theta_\alpha = \theta_1) \times \\ & P(p_{m-k-1} | \vec{p}_{m-k-2}, \vec{g}_{m-k-1}, \theta_\alpha = \theta_1) \times \\ & \vdots \\ & P(g_1 | \theta_\alpha = \theta_1) \end{aligned}$$

We note that $P(p_i | \vec{p}_{i_1}, \vec{g}_i, \theta_\alpha = \theta_1) = P(p_i | g_i, \theta_\alpha = \theta_1)$ and for all i $P(g_i | \vec{p}_{i-1}, \vec{g}_{i-1}, \theta_\alpha = \theta_1) = P(g_i | \vec{p}_{i-1}, \vec{g}_{i-1})$ since strategies can not be conditioned on information that is unavailable. Plugging these results in and cancelling terms we have

$$P(\theta_\alpha = \theta_1 | \vec{p}_{m-k}, \vec{g}_{m-k}) = \frac{P(\theta_\alpha = \theta_1) \prod_{i=1}^{i=m-k} P(p_i | g_i, \theta_\alpha = \theta_1)}{\sum_{\theta_\alpha} P(\theta_\alpha = \theta_1) \prod_{i=1}^{i=m-k} P(p_i | g_i, \theta_\alpha = \theta_1)}$$

or more explicitly

$$P(\theta_\alpha = \theta_1 | \vec{p}_{m-k}, \vec{g}_{m-k}) = \frac{P(\vec{p}_{m-k} | \vec{g}_{m-k}, \theta_\alpha = \theta_1) \omega_1}{P(\vec{p}_{m-k} | \theta_\alpha = \theta_1, \vec{g}_{m-k}) \omega_1 + P(\vec{p}_{m-k} | \theta_\alpha = \theta_2, \vec{g}_{m-k})(1 - \omega_1)}$$

Finally, in the case of $\sigma_1 = \sigma_2 = \sigma$ we have

$$P(\vec{p}_{m-k} | \theta_\alpha = \theta_1, \vec{g}_{m-k}) = \frac{1}{(\sqrt{2\pi}\sigma)^{m-k}} \exp\left[\frac{-1}{2\sigma^2} \sum_{i=1}^{m-k} (p_i - \mu_{g_i,\theta_1})^2\right]$$

Similarly

$$\begin{aligned} P(p_{m-k} | \vec{p}_{m-k-1}, \vec{g}_{m-k-1}, g_{m-k}) &= \sum_\theta P(p_{m-k} | \theta, g_{m-k}) P(\theta | \vec{p}_{m-k-1}, \vec{g}_{m-k-1}) \\ &= \sum_\theta \frac{\exp[-(p_{m-k} - \mu_{g_{m-k},\theta})^2 / 2\sigma^2]}{\sqrt{2\pi}\sigma} P(\theta | \vec{p}_{m-k-1}, \vec{g}_{m-k-1}) \end{aligned}$$

Having defined all the necessary quantities we can now do the integration and compare $E(p_j | \vec{p}_{m-k-1}, \vec{g}_{m-k-1}, g_{m-k})$ for the two different choices of g_{m-k}.

Definition 5 To facilitate the calculation we make the following definitions:

$$N(\vec{p}_{m-k}, \vec{g}_{m-k}) \equiv \exp\left[\frac{-1}{2\sigma^2} \sum_{i=1}^{m-k} (p_i - \mu_{g_i,\theta_1})^2\right] \frac{\omega_1}{(\sqrt{2\pi}\sigma)^{m-k}}$$

$$D(\vec{p}_{m-k}, \vec{g}_{m-k}) \equiv \exp\left[\frac{-1}{2\sigma^2} \sum_{i=1}^{m-k} (p_i - \mu_{g_i,\theta_1})^2\right] \frac{\omega_1}{(\sqrt{2\pi}\sigma)^{m-k}} + \exp\left[\frac{-1}{2\sigma^2} \sum_{i=1}^{m-k} (p_i - \mu_{g_i,\theta_2})^2\right] \frac{1 - \omega_1}{(\sqrt{2\pi}\sigma)^{m-k}}$$

Using Equation (5.1) we can write the posterior probability of ω_{m-k+1} as

$$P(\omega_{m-k+1}|\vec{p}_{m-k-1}, \vec{g}_{m-k-1}, g_{m-k}) = \int dp_{m-k}\, \delta\left(\omega_{m-k+1} - \frac{N(\vec{p}_{m-k}, \vec{g}_{m-k})}{D(\vec{p}_{m-k}, \vec{g}_{m-k})}\right) D(\vec{p}_{m-k}, \vec{g}_{m-k})$$

We want to know how this distribution changes when g_{m-k} is changed to the other arm. To do this note a symmetry of the N and D functions, namely that if we define $\tilde{p}_{m-k} = \mu_1 + \mu_2 - p_{m-k}$ then

$$\begin{aligned} N(\vec{p}_{m-k}, \vec{g}_{m-k}) &= N(\vec{p}_{m-k-1}, \tilde{p}_{m-k}, \vec{g}_{m-k-1}, \neg g_{m-k}) \\ D(\vec{p}_{m-k}, \vec{g}_{m-k}) &= D(\vec{p}_{m-k-1}, \tilde{p}_{m-k}, \vec{g}_{m-k-1}, \neg g_{m-k}) \end{aligned}$$

where $\neg g_{m-k}$ is the arm other than g_{m-k}. So by changing the variable of integration we have

$$\begin{aligned} &P(\omega_{m-k+1}|\vec{p}_{m-k-1},\vec{g}_{m-k-1}, g_{m-k}) \\ &\quad = \int dp_{m-k} \left[\delta\left(\omega_{m-k+1} - \frac{N(\vec{p}_{m-k-1}, \tilde{p}_{m-k}(p_{m-k}), \vec{g}_{m-k-1}, \neg g_{m-k})}{D(\vec{p}_{m-k-1}, \tilde{p}_{m-k}(p_{m-k}), \vec{g}_{m-k-1}, \neg g_{m-k})}\right)\right. \\ &\qquad\qquad \left. \times D(\vec{p}_{m-k-1}, \tilde{p}_{m-k}(p_{m-k}), \vec{g}_{m-k-1}, \neg g_{m-k})\right] \\ &\quad = -\int_{\infty}^{-\infty} d\tilde{p}_{m-k} \left[\delta\left(\omega_{m-k+1} - \frac{N(\vec{p}_{m-k-1}, \tilde{p}_{m-k}, \vec{g}_{m-k-1}, \neg g_{m-k})}{D(\vec{p}_{m-k-1}, \tilde{p}_{m-k}, \vec{g}_{m-k-1}, \neg g_{m-k})}\right)\right. \\ &\qquad\qquad \left. \times D(\vec{p}_{m-k-1}, \tilde{p}_{m-k}, \vec{g}_{m-k-1}, \neg g_{m-k})\right] \\ &\quad = P(\omega_{m-k+1}|\vec{p}_{m-k-1}, \vec{g}_{m-k-1}, \neg g_{m-k}) \end{aligned}$$

Thus we have proven that the probability of obtaining a particular posterior ω_{m-k} is independent of the next pull, g_{m-k}, and thus any quantity like future payoffs that is dependent on ω_{m-k} alone will also be independent of the next pull. In this case the greedy algorithm is optimal, and we can do no better than follow the strategy of optimizing the payoff for the next pull. Thus we have proven the following result:

Theorem 1 *For $\sigma_1 = \sigma_2$, the greedy algorithm is optimal.*

5.2 Other cases in which the greedy strategy is optimal

Trivially, in addition to the result of the preceding subsection, it is also true that if the μ's are the same but the σ's are different, then the expected payoffs are identical for all algorithms, so again the greedy strategy is optimal.

We can also prove that the greedy algorithm is optimal for arbitrary θ_1 and θ_2 for special values of $\mathcal{P}$. Intuitively, this follows from the fact that when $\mathcal{P}$ is close to either 0 or 1, so there is strong prior certainty about which arm is which, for small enough m no results of

preceding pulls can alter our choice for the current pull. Accordingly, when choosing a pull in the past, we knew with certainty what the current pull would be; no information gained by "exploring" can have any effect and so the greedy algorithm is optimal.

To establish this formally, here we derive lower bounds on the amount $\mathcal{P}$ can deviate from either 0 or 1 with the greedy algorithm still being optimal. We first consider the case of $m = 2$ and then generalize to arbitrary m.

We proceed as above by showing that $E(p_2|g_1)$ is independent of g_1. As before we expand in the posterior probability that $\theta_\alpha = \theta_1$, ω_2, and write the expected second payoff for the optimal strategy as

$$\begin{aligned} E(p_2|g_1) &= \int d\omega_2 \, E(p_2|\omega_2) P(\omega_2|g_1) \\ &= \frac{\mu_1 + \mu_2}{2} + |\mu_1 - \mu_2| \int d\omega_2 \, |\omega_2 - 1/2| P(\omega_2|g_1) \end{aligned}$$

(see Equation (5) of Section 4.1).

So the change in the expected p_2 for different initial guesses g_1 is

$$\begin{aligned} \Delta E(p_2) &\equiv E(p_2|g_1 = \alpha) - E(p_2|g_1 = \beta) \\ &= |\mu_1 - \mu_2| \int d\omega_2 \, |\omega_2 - 1/2| \Big(P(\omega_2|g_1 = \alpha) - P(\omega_2|g_1 = \beta) \Big) \end{aligned}$$

As we have previously noted, the greedy algorithm is optimal if $|\Delta E(p_2)| < |\Delta E(p_1)|$. So it is optimal if

$$\int d\omega_2 \, |\omega_2 - 1/2| |P(\omega_2|g_1 = \alpha) - P(\omega_2|g_1 = \beta)| < |(2\mathcal{P} - 1)|$$

(again, see Section 4.1).

We will evaluate the integral on the left hand side of this inequality for the worst case, and thereby determine worst-case conditions for the greedy strategy to be optimal. (A rough picture of $|\omega_2 - 1/2|$, and typical $P(\omega_2|g_1 = \alpha)$, and $P(\omega_2|g_1 = \beta)$ is given in Figure 4.)

To find our worst-case scenario for a particular $\mathcal{P}$, we maximize the difference $P(\omega_2|g_1 = \alpha) - P(\omega_2|g_1 = \beta)$. Now

$$E(\omega_2|g_1) = \int d\omega_2 \, \omega_2 P(\omega_2|g_1) = \int d\omega_2 dp_1 \, \omega_2 \delta(\omega_2 - P(\theta_\alpha = \theta_1|p_1, g_1)) P(p_1|g_1).$$

but by interchanging the order of integration we can evaluate this as $P(\theta_\alpha = \theta_1|g_1)$. So $E(\omega_2|g_1) = \mathcal{P}$, independent of g_1. Therefore to maximize our difference we should have

$$\begin{aligned} P(\omega_2|g_1 = \alpha) &= \mathcal{P}\delta(\omega_2 - 1) + (1 - \mathcal{P})\delta(\omega_2) \\ P(\omega_2|g_1 = \beta) &= \delta(\omega_2 - \mathcal{P}) \end{aligned}$$

For this extreme case, our integral gives $\Delta E(p_2) = 1/2 - |\mathcal{P} - 1/2|$.

Using this result in our inequality we see that in the worst case the greedy strategy will be optimal if $1/2 - |\mathcal{P} - 1/2| < |2\mathcal{P} - 1|$, i.e., either $\mathcal{P} < 1/3$ or $\mathcal{P} > 2/3$.

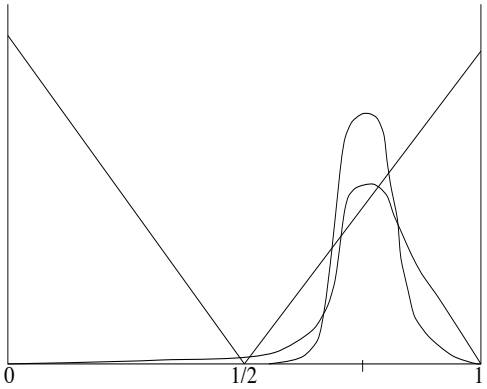

Figure 4: A schematic view of $|\omega_2 - 1/2|$ and the two probability distributions $P(\omega_2|g_1 = \alpha)$ and $P(\omega_2|g_1 = \beta)$. We seek the $P(\omega_2|g_1 = \alpha) - P(\omega_2|g_1 = \beta)$ which when integrated against $|\omega_2 - 1/2|$ gives the largest possible value, given that the two distributions must have the same mean.

Note that this worst case analysis is fairly weak in that (for example) it does not capture the fact that for $m = 2$ the greedy algorithm is in fact optimal for all $\mathcal{P}$. However this type of analysis has the advantage that it is trivial to extend it to arbitrary m. For such a case the greedy algorithm will be optimal if the absolute value of

$$\Delta E\Big(\sum_{i=2}^{m} p_i\Big) \equiv E\Big(\sum_{i=2}^{m} p_i | g_1 = \alpha\Big) - E\Big(\sum_{i=2}^{m} p_i | g_1 = \beta\Big)$$

is less then $|\Delta E(p_1)|$. Again using worst case analysis, we find

$$\Delta E\Big(\sum_{i=2}^{m} p_i\Big) \leq |\mu_1 - \mu_2|(m-1)(1/2 - |\mathcal{P} - 1/2|)$$

This can be seen by bounding $\Delta E(p_i) = \int d\omega_i |\omega_i - 1/2| |(P(\omega_i|g_1 = \alpha) - P(\omega_i|g_1 = \beta))$ as above and then bounding $\Delta E(\sum_{i=2}^{m} p_i)$ by summing the $m-1$ associated bounds on the individual $\Delta E(p_i)$. Thus we have established the following:

Theorem 2 *If either $\mathcal{P} < 1/(m+1)$ or $\mathcal{P} > m/(m+1)$, then the greedy algorithm is optimal.*

It should be emphasized just how poor these worst case bounds really are. It is hard to imagine how a distant future expected payoff, $E(p_i|g_1)$ for large i, can be as sensitive to g_1 as this worst case analysis requires. So this worst case analysis must overestimate $\Delta E(\sum_i p_i)$ significantly, and therefore it must significantly underestimate the range of $\mathcal{P}$ for which the greedy strategy is optimal.

Note that nowhere in this proof of the optimality of the greedy strategy did we make use of the fact that our payoff distributions are Gaussian. Consequently, the results derived above hold *for all payoff distributions*. There may be ways to extend these bounds considerably by

exploiting properties of particular payoff distributions (e.g., the properties associated with Gaussian distributions).

We could also have done these calculations with a discounting factor included. For geometric discounting with $\gamma < 1$, which weights the near future more importantly than the distant future, we would expect the greedy algorithm to be optimal for even larger ranges of $\mathcal{P}$. However, in mapping the bandit problem to optimization where we are interested in extremal values the proper discounting might be that which strongly weights extremal payoffs while largely ignoring all the others. In this case the discounting is far from uniform and we might expect a greedy strategy to perform very poorly indeed. In this situation it is not at all clear how relevant the present bandit model is to optimization.

Before we address these drawbacks by presenting a bandit model more directly related to issues of importance in optimization, we turn to a previous analysis of this bandit problem as it relates to genetic optimization algorithms.

6 Previous analysis of bandits and genetic algorithms

The bandit problem analyzed in this paper has been invoked by Holland [4] as one of the theoretical motivations for genetic optimization algorithms. Genetic algorithms are optimization techniques loosely based on biological metaphors where a population of candidate solutions "breeds" with each other to produce a new and hopefully improved population. We make two points regarding Holland's analysis in this section. We point out a fatal flaw in Holland's analysis and its resultant (supposed) justification for genetic algorithms. Then we demonstrate how poorly Holland's genetic algorithm-oriented strategy performs when compared to even the most simple of alternative allocation strategies.

Holland's work considers the case of m total pulls with no discounting. Though he states that "the object ... is to discover a procedure for distributing an arbitrary number of trials ... so as to maximize the expected payoff" he goes on to restrict himself to a severly limited class of strategies. Only strategies which allocate n pulls each to arms α and β and then assign the remaining $m - 2n$ pulls to the observed best arm are considered. (It is not at all clear why one should believe Holland's assertion that results concerning such a limited class of strategies should have implications about the optimal strategy for multi-armed Gaussian bandits.)

Given this class of strategies, Holland seeks to determine n^*, the value of n which optimizes the expected payoff within this class of strategies. In [4] he finds

$$n^* = b^2 \ln \left[\frac{m^2}{16\pi b^4 \ln m} \right]$$

where $b = \sigma_1/(\mu_1 - \mu_2)$ and σ_1 is the σ associated with the higher mean, μ_1. This result seems very odd since n^* does not depend upon σ_2. After all, if (for example) $\sigma_2 = 0$ a single pull of either arm categorically determines which arm is which while if σ_2 is larger more work is needed to identify the arms. So the fact that Holland's result is independent of σ_2 is *ipso facto* reason to believe it is wrong.

This questionable nature of Holland's result can be understood by going over Holland's calculation carefully. The problem is that, unfortunately, he makes a serious mathematical error early in his calculation, an error that leads directly to his incorrect result. This error is as follows:

Holland calculates the expected loss in net payoff that would result were we to pull the arm observed (based on the initial $2n$ pulls) to have the *lower* mean for the remaining $m-2n$ pulls. He then uses this to determine n^*. He calculates this expected loss as $(m-n)|\mu_1-\mu_2|$. However, this quantity is the *unconditioned* expected loss, it is *not* the loss conditioned on the information available after the $2n$ pulls. The proper quantity to calculate is instead the expected loss *conditioned on the fact that the arm being pulled in the remaining $m-2n$ pulls was observed to have the higher payoff in the first $2n$ pulls.* This quantity will in general be much smaller than the unconditioned loss, and may actually even be negative, indicating a gain from using the arm which appears to have a lower mean! (For example, appropriate values of $\mathcal{P}$ can result in such a phenomenon.) Indeed, intuitively, if $\mathcal{P} = 1$ or 0 (something Holland never precludes), why wouldn't the proper conditioned expected loss be minimized by having $n^* = 0$?

We now sketch how a proper calculation leads to the determination of n^*. We begin by writing the expected payoff as

$$E(p|m,n) = n(\mu_1+\mu_2) + (m-2n)\int d\vec{p}^{\,\alpha} d\vec{p}^{\,\beta} \sum_g \left(P(\theta_\alpha=\theta_1)\mu_{g,\theta_1} + P(\theta_\alpha=\theta_2)\mu_{g,\theta_2}\right) P(g|\vec{p}^{\,\alpha},\vec{p}^{\,\beta},n) P(\vec{p}^{\,\alpha},\vec{p}^{\,\beta}|n) \tag{12}$$

where $\vec{p}^{\,\alpha}$ and $\vec{p}^{\,\beta}$ are the n-vectors of payoffs for arms α and β respectively in each one's first n pulls, g labels the arm selected at the decision point, and the probability $P(\vec{p}^{\,\alpha},\vec{p}^{\,\beta}|n)$ is given by $\prod_{i=1}^n \sum_\theta P(p_i^\alpha|\alpha,\theta)P(p_i^\beta|\beta,\theta)P(\theta)$. The arm pulled at the decision point $P(g|\vec{p}^{\,\alpha},\vec{p}^{\,\beta})$ is the only undetermined quantity for the class of strategies Holland considers. If we follow Holland and select the observed best arm we have

$$P(g|\vec{p}^{\,\alpha},\vec{p}^{\,\beta},n) = \delta(g-\alpha)\,\theta\left(\sum_{i=1}^n (p_i^\alpha - p_i^\beta)\right) + \delta(g-\beta)\,\theta\left(\sum_{i=1}^n (p_i^\beta - p_i^\alpha)\right)$$

where $\theta(\cdot)$ is the Heaviside step function. Substituting this result into Equation (12) and doing all integrations leaves $E(p|m,n)$ as a function of m and n alone. It is this function which should be maximized with respect to n to determine $n^* = \text{argmax}_n\, E(p|m,n)$. We will not undertake that calculation here since nothing interesting can be learned from doing so; Holland does not use this strategy (rather he uses what he argues is a good approximation to it), nor is this strategy optimal.

Although determining the optimal strategy in practice appears to be quite difficult, in Holland's problem it is straightforward to determine the greedy strategy. One can then use that greedy strategy—a strategy with only exploitation and no exploration at all—to see how well Holland's strategy performs. It turns out that Holland's strategy performs quite poorly even in comparison to the greedy strategy. To see this consider the case of $m = 100$

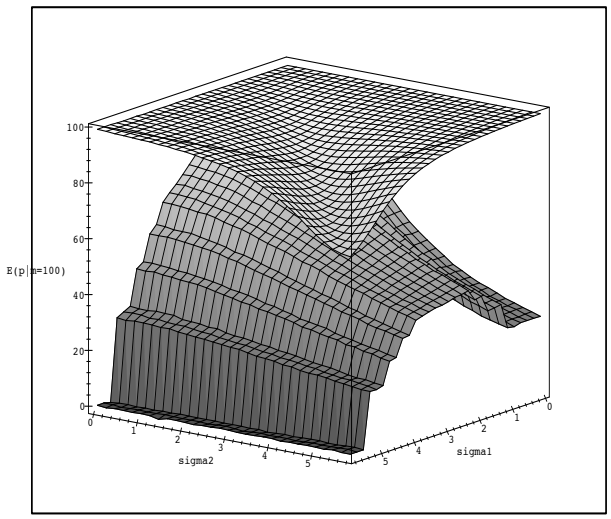

Figure 5: Monte Carlo determination of expected payoffs $E(p|m = 100)$ for 100 pulls and $\mathcal{P} = 0.5$ for both greedy strategies (top surface) and genetic-algorithm-like strategies (bottom surface).

pulls with $\mathcal{P} = 0.5$ (Holland also focussed on $\mathcal{P} = 0.5$). We ran Monte Carlo simulations to estimate the expected payoffs for this scenario, $E(p|m = 100, \mathcal{P} = 0.5)$. The results are presented in Figure 5. For no values of σ_1 or σ_2 does Holland's strategy perform better than the simple greedy strategy. (As an aside, note that because Equation (12) for n^* contains σ_1 but not σ_2, the payoffs for the genetic-algorithm-like strategy are not symmetric about the line $\sigma_1 = \sigma_2$.)

The implications of this for genetic algorithms are very strong; genetic algorithms are based on the premise that one should engage in exploration, yet for the very problem that Holland invokes to justify genetic algorithms, the strategy of not exploring at all performs far better than Holland-style ("genetic-algorithm-like") exploring algorithms. Of course, this does not mean that one should never explore when performing optimization. Rather it means that Holland's argument justifying exploration is flawed. (In fact it has been proven that there is no optimization technique that works better than any other, across the set of all optimization problems, and in this neither using exploration nor not using exploration is *a priori* preferable. See [8].)

One reason that Holland's strategy performs so poorly is that it makes very poor use of data in making its decisions. After all, in Holland's strategy there is only a single decision point. In this regard it is unlike most efficient optimization methods. Indeed, even the greedy strategy described earlier makes a decision after every pull. Holland was not unaware of this shortcoming in his strategy; he himself suggests that his strategy might be improved be having a number of decision points separated by an exponentially increasing number of pulls. This will do little to alleviate the problem though because during exponentially

long periods the strategy is committed to a single arm in spite of what new data might be indicating. And in any case, as proven above, Holland's strategy is based on fallacious mathematics.

7 A better bandit

We have seen how deceptively difficult the 2-armed bandit problem is. Due to this difficulty, analytic insights into optimization through a study of this bandit problem will be hard to obtain. Moreover, in some of the cases where we have been able to obtain results, the greedy algorithm is optimal, indicating that in those cases there is no reason to explore to gain information. In contrast, it is rarely the case in real-world optimization that greedy algorithms are optimal. Usually there is *some* benefit to exploring, because even though the short term payoff might be lower, with such a strategy there is the potential for very high long-term payoffs. More concretely, in practice it is usually the case that a good optimization algorithm will in some circumstances make a decision whose associated payoff distribution has low mean, as long as the distribution has a very long tail extending into the high payoff region.

In this section we present an improved bandit model that matches real-world optimization better than the Holland-style bandit problem discussed so far. This new bandit problem is also analytically simpler than the Holland-style problem. In sum, this new problem has none of the shortcomings of the Holland problem discussed above, and it is straightforward to use it to investigate the trade-off between exploration and exploitation. Bandit problems similar to the one presented in this Section have been studied previously [7] though almost always for Bernoulli payoffs.

To minimize the differences from the Holland-style bandit problem already analyzed we consider the case where all payoff distributions are again Gaussian. However, in the present case we assume that arm α is characterized by parameters θ_2 (*i.e.*, we know beforehand the payoff distribution of arm α), but that arm β is characterized by either parameters θ_1 or θ_3. For simplicity we assume that $\sigma_1 = \sigma_3 = \sigma$. The goal as always is to maximize the total payoff over m pulls. We do not include any discounting, though it would be simple to do so. There is a prior probability $\mathcal{P}$ that arm β has parameters given by θ_1 and a prior probability $1 - \mathcal{P}$ that arm β has parameters θ_3. Without loss of generality we assume that $\mu_1 > \mu_3$.

For simplicity, rather than investigate the optimal algorithm, we consider pseudo-optimal strategies, $\mathcal{S}$, where $\mathcal{S}$ pulls arm β n times, and then based on the n dimensional payoff vector $\vec{p}$ deduces which of θ_1 or θ_3 has the higher posterior probability. $\mathcal{S}$ then chooses either α or β depending on this posterior, and pulls that arm for the remaining $m - n$ pulls. (Note the similarity of this class of strategies to Holland's strategies for his bandit problem.) We assume that $\mathcal{P}\mu_1 + (1 - \mathcal{P})\mu_3 < \mu_2$ so that a greedy strategy, $\mathcal{G}$, would always guess α for all m pulls with an expected payoff of $E(p|\mathcal{G}) = m\mu_2$.

We now determine the payoff of the pseudo-optimal strategy, $\mathcal{S}$. The calculation parallels that found for the 2-armed bandit. We define the n dimensional vector $\vec{p}$ to be the first n payoffs and $\vec{q}$ to be the $m - n$ dimensional vector of the remaining payoffs. We also let g

label the choice of arm after the n pulls, and let θ_β run over the two possibilities for arm β. With this notation the expected payoff is

$$E(p|\mathcal{S},m,n) = \int d\vec{p}\, d\vec{q} \sum_{g,\theta_\beta} \left(\sum_{i=1}^{n} p_i + \sum_{i=1}^{m-n} q_i \right) P(\vec{p},\vec{q},\theta_\beta,g|\mathcal{S},m,n)$$
$$= \int d\vec{p}\, d\vec{q} \sum_{g,\theta_\beta} \left(\sum_{i=1}^{n} p_i + \sum_{i=1}^{m-n} q_i \right) P(\vec{q}|g,\theta_\beta,m,n)\, P(g|\vec{p},\theta_\beta)\, P(\vec{p}|\theta_\beta)\, P(\theta_\beta)$$

We split the sum into two terms, one for $\vec{p}$ and one for $\vec{q}$ and obtain:

$$E(p|\mathcal{S},m,n) = \int d\vec{p} \sum_{\theta_\beta} \sum_{i=1}^{n} p_i P(\vec{p}|\theta_\beta)\, P(\theta_\beta) +$$
$$\int d\vec{p} \sum_{g,\theta_\beta} \left(\int d\vec{q} \sum_{i=1}^{m-n} q_i \right) P(\vec{q}|g,\theta_\beta,m,n)\, P(g|\vec{p},\theta_\beta)\, P(\vec{p}|\theta_\beta)\, P(\theta_\beta)$$

Both terms in the above equation can be simplified since the i payoffs are independent, $P(\vec{p}|\theta_\beta) = \prod_{i=1}^{n} P(p_i|\theta_\beta)$ and $P(\vec{q}|g,\theta_\beta,m,n) = \prod_{i=1}^{m-n} P(q_i|g,\theta_\beta)$. This means we obtain the same result for each p_i in the associated sum p_i and the same result for each q_i in its associated sum. Thus we can write the above as

$$E(p|\mathcal{S},m,n) = n \int dp \sum_{g,\theta_\beta} p\, P(p|\theta_\beta)\, P(\theta_\beta) +$$
$$(m-n) \sum_{g} \int dq\, q P(q|g,m,N,\theta_\beta) \int d\vec{p} \sum_{\theta_\beta} P(g|\vec{p},\theta_\beta)\, P(\vec{p}|\theta_\beta)\, P(\theta_\beta)$$
$$= n[\mathcal{P}(\mu_1 - \mu_3) + \mu_3] + (m-n)\Big(\mu_2 \mathcal{P}^\alpha + \mu_1 P_c^\beta + \mu_3 P_i^\beta\Big)$$
$$= n[\mathcal{P}(\mu_1 - \mu_3) + \mu_3] + (m-n)\Big(\mu_2 + (\mu_1 - \mu_2) P_c^\beta + (\mu_3 - \mu_2) P_i^\beta\Big)$$

where P_i^β is the probability that $\mathcal{S}$ deduces the payoff distribution for β incorrectly and chooses arm β, P_c^β is the probability $\mathcal{S}$ it deduces it correctly and chooses arm β, and P^α is the probability that $\mathcal{S}$ selects arm α. (The fact that these three probabilities must sum to 1 has been used in deriving the last line.) Explicitly these probabilities are:

$$P_c^\beta = \int d\vec{p} P(g=\beta|\vec{p}) P(\vec{p}|\theta_1) P(\theta_1) = \mathcal{P} \int d\vec{p} P(g=\beta|\vec{p}) P(\vec{p}|\theta_1)$$
$$P_i^\beta = \int d\vec{p} P(g=\beta|\vec{p}) P(\vec{p}|\theta_3) P(\theta_3) = (1-\mathcal{P}) \int d\vec{p} P(g=\beta|\vec{p}) P(\vec{p}|\theta_3)$$

To continue we next must calculate $P(g=\beta|\vec{p})$. After obtaining the n dimensional payoff vector $\vec{p}$ from arm β we can calculate the probability the the arm β is associated with either θ_1 or θ_2. This probability is then used to determine g. To evaluate this probability we use Bayes' theorem:

$$P(\theta_\beta = \theta_1|\vec{p}) = \frac{1}{\mathcal{N}} P(\vec{p}|\theta_\beta = \theta_1) P(\theta_\beta = \theta_1) = \frac{1}{(\sqrt{2\pi}\sigma)^n \mathcal{N}} \exp\left[-\chi_1^2/2\sigma^2\right] \mathcal{P}$$
$$P(\theta_\beta = \theta_3|\vec{p}) = \frac{1}{\mathcal{N}} P(\vec{p}|\theta_\beta = \theta_3) P(\theta_\beta = \theta_3) = \frac{1}{(\sqrt{2\pi}\sigma)^n \mathcal{N}} \exp\left[-\chi_3^2/2\sigma^2\right] (1-\mathcal{P})$$

where $\chi_i^2 \equiv \sum_{j=1}^n (p_j - \mu_i)^2$ and $i = 1$ or 3, and $\mathcal{N} \equiv P(\theta_\beta = \theta_1|\vec{p}) + P(\beta = \theta_3|\vec{p})$ is a normalization constant.

Our strategy is to stick with arm β for our remaining $m - n$ pulls if in light of our data $\vec{p}$ we believe it has a higher expected payoff than arm α, *i.e.,* if $\mu_1 P(\theta_\beta = \theta_1|\vec{p}) + \mu_3 P(\theta_\beta = \theta_3|\vec{p}) > \mu_2 \mathcal{N}$. Rather than evaluate this condition, which is analytically messy, we simplify once again, and consider an algorithm $\mathcal{S}'$ that uses an alternate criterion for choosing g. (As shown below, even this simplified algorithm has much better performance than the greedy algorithm $\mathcal{G}$, in contrast to the situation with Holland's algorithm and his bandit problem.)

The strategy $\mathcal{S}'$ chooses to stick with arm β simply if it is more likely, given the data, that arm β is described by the parameters μ_1 rather than the parameters μ_3; we remain with arm β if $P(\theta_\beta = \theta_1|\vec{p}) > P(\theta_\beta = \theta_3|\vec{p})$.

With this new strategy the normalization constant, $\mathcal{N}$, no longer matters. Substituting in for the relevant probabilities we find the that $\mathcal{S}'$ says we should continue with arm β if

$$\exp\left[(\chi_3^2 - \chi_1^2)/2\sigma^2\right] > \frac{1-\mathcal{P}}{\mathcal{P}}$$

Accordingly, under this strategy,

$$P(g = \beta|\vec{p}) = \theta\left(\exp\left[(\chi_3^2 - \chi_1^2)/2\sigma^2\right] - \frac{1-\mathcal{P}}{\mathcal{P}}\right)$$

where as usual $\theta(\cdot)$ is the Heaviside function.

To calculate the expected payoff $E(p|\mathcal{S}', m, n)$ we need to evaluate the probabilities P_i^β and P_c^β introduced earlier. These probabilities depend on the boundaries in $\vec{p}$ space where the choice of $\mathcal{S}'$ for g changes. Thus we are lead to consider the boundary where $\exp\left[\chi_3^2 - \chi_1^2/2\sigma^2\right] = (1-\mathcal{P})/\mathcal{P}$, i.e., where $\chi_3^2 - \chi_1^2 = 2\sigma^2 \ln[(1-\mathcal{P})/\mathcal{P}]$. As a function of $\vec{p}$, this equation describes a hyperplane in an n-dimensional space with normal vector equal to $\vec{1} \equiv (1, 1, \cdots, 1)$. We translate and rotate our coordinate axes so that the origin lies at $\vec{\mu}_3 \equiv \mu_3 \vec{1}$ and $\vec{1}$ lies in the $\hat{r}_1$ direction, $(1, 0, 0, \ldots)$. The length of the vector $(\mu_1 - \mu_3)\vec{1}$ is $\sqrt{n}(\mu_1 - \mu_3)$ so that in this new coordinate system $(\mu_1 - \mu_3)\vec{1} = \vec{\mu}_1 = (\sqrt{n}(\mu_1 - \mu_3), 0, \ldots, 0)$. The hyperplane intersects the $\hat{r}_1$ axis at $(z, 0, \ldots, 0)$ where $z^2 - (z - \sqrt{n}(\mu_1 - \mu_3)^2 = 2\sigma^2 \ln[(1-\mathcal{P})/\mathcal{P}])$. Solving for z we find

$$z = \frac{2\sigma^2 \ln\left[(1-\mathcal{P})/\mathcal{P}\right] - n(\mu_1 - \mu_3)^2}{2\sqrt{n}(\mu_1 - \mu_3)}$$

In these coordinates P_c^β is simple to calculate. With $\vec{r}$ an n-dimensional dummy variable, we get

$$\begin{aligned} P_c^\beta &= \mathcal{P} \int d\vec{r}\, \frac{\exp\left[-(\vec{r} - \vec{\mu}_1)^2/2\sigma^2\right]}{(\sqrt{2\pi}\sigma)^n}\, \theta(r_1 > z) = \mathcal{P} \int_z^\infty dr_1\, \frac{\exp\left[-\left(r_1 - \sqrt{n}(\mu_1 - \mu_3)\right)^2/2\sigma^2\right]}{\sqrt{2\pi}\sigma} \\ &= \frac{\mathcal{P}}{2}\, \mathrm{erfc}\left[\frac{z - \sqrt{n}(\mu_1 - \mu_3)}{\sqrt{2}\sigma}\right] \end{aligned} \tag{13}$$

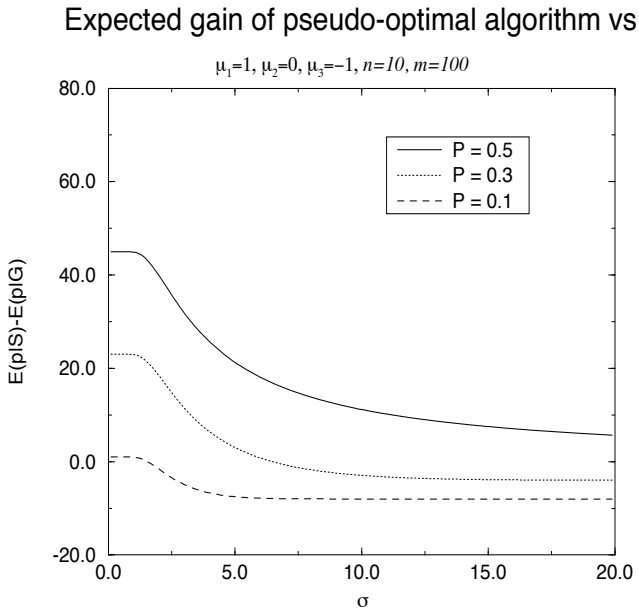

Figure 6: Gain in payoff of the pseudo-optimal algorithm $\mathcal{S}'$ over a greedy algorithm for $n = 10$, and $m = 100$ and various prior probabilities, $\mathcal{P}$. All $\mathcal{P}$ values plotted obey $\mathcal{P}\mu_1 + (1-\mathcal{P})\mu_3 \leq \mu_2$ so that the greedy strategy always guesses arm α.

Similarly we can calculate P_i^β as

$$P_i^\beta = (1-\mathcal{P}) \int d\vec{r}\, \frac{\exp\left(-[\vec{r}-\vec{\mu}_3]^2/2\sigma^2\right)}{(\sqrt{2\pi}\sigma)^n}\, \theta(z > r_1) = (1-\mathcal{P}) \int_{-\infty}^{z} dr_1\, \frac{\exp\left[-r_1^2/2\sigma^2\right]}{\sqrt{2\pi}\sigma}$$
$$= \frac{1-\mathcal{P}}{2}\, \mathrm{erfc}\left[\frac{-z}{\sqrt{2}\sigma}\right] \tag{14}$$

Using Equations (13) and (14) in Equation (13) we have an expression for the expected payoff of the pseudo-optimal algorithm $\mathcal{S}'$, which can be compared to the expected payoff for the greedy algorithm, $\mathcal{G}$. In that the greedy algorithm never explores, this allows us to investigate the trade-off between exploration (done by $\mathcal{S}'$) and pure exploitation (done by$\mathcal{G}$).

The expected payoff of $\mathcal{S}'$ is

$$E(p|\mathcal{S}') = n[(\mathcal{P}(\mu_1-\mu_3)+\mu_3] + (m-n)\left(\mu_2 + \frac{\mathcal{P}(\mu_1-\mu_2)}{2}\, \mathrm{erfc}\left[\frac{z-\sqrt{n}(\mu_1-\mu_3)}{\sqrt{2}\sigma}\right]\right.$$
$$\left. + \frac{(1-\mathcal{P})(\mu_3-\mu_2)}{2}\, \mathrm{erfc}\left[\frac{-z}{\sqrt{2}\sigma}\right]\right)$$

A plot of the expected gain of the pseudo-optimal algorithm $\mathcal{S}'$ over the greedy algorithm—a plot of how much exploration can help—is presented in Figure 6. Parameters are chosen so that the greedy algorithm always guesses arm α. Note that as $\mathcal{P}$ increases, it becomes more likely that $\theta_\beta = \theta_1$, and the pseudo-optimal algorithm does increasingly well. But even when

it is less likely that arm β has the higher mean, our pseudo-optimal algorithm $\mathcal{S}'$ can still outperform the greedy algorithm, depending on the value of σ.

Using our expression for the expected payoff of $\mathcal{S}'$ we could also determine the optimal amount of time, n_{opt}, to spend gathering information given a fixed number of total pulls, m. This would be done by maximizing our expression for the expected payoff of $\mathcal{S}'$ with respect to n. Such an optimization would improve the performance of $\mathcal{S}'$ over the greedy algorithm even further.

Such a calculation is very much along the lines of what we would like to accomplish in optimization. After all, in real-world optimization we are given some fixed time in which to locate an extremal point and we would like to calculate the optimal balance between an exploratory phase and an exploitive phase so as to locate a good point. However performing such a calculation is beyond the scope of this already lengthy paper.

8 Discussion and Conclusions

We have considered a much simplified bandit problem in the hopes of learning about the exploration/exploitation tradeoff important in optimization. But as Section 4 amply demonstrated, even for the deceptively simple bandit problem that we consider, it is very difficult to solve for the optimal strategy. Despite this though, it must be kept in mind that incorporating domain knowledge in constructing effective optimization strategies (as we have tried to do here) is vital. Indeed, theorems exist [8] showing that when no domain specific knowledge is utilized all optimization algorithms perform equally poorly on average.

Previous analyses of this bandit problem have been used as theoretical justification for genetic optimization algorithms. We have demonstrated that the Holland's analysis of the bandit problem is flawed and its connection to the supposed optimality of genetic algorithms is seriously called into question.

Because of the difficulty of this bandit problem there is much room for future work. In particular, it would be very interesting to determine the "phase diagram" of ranges of relevant parameters for which the greedy strategy is optimal. Even if analytic progress cannot be made in this direction Monte Carlo simulations could go a long way towards answering this question. We have not spent alot of effort on the effects of different discounting schedules. More detailed work on the effects of discounting schedules would prove very illuminating and presumably increase the range over which the greedy algorithm is optimal.

Given these difficulties with our original bandit problem, we considered a modified one that is not only simpler, but also exhibits more interesting behavior from an optimization point of view. Our modified bandit problem can be extended in a number of interesting ways to more closely mimic the kinds of problems that occur in real optimization.

The algorithm $\mathcal{S}'$ we investigated for our modified problem naturally suggests an improved algorithm—a recursive version of $\mathcal{S}'$. In the recursive version we imagine applying the same criterion used to determine the single guess to be made for all pulls following the preliminary explorative phase (*i.e.,* all pulls following the first n pulls) to pulls both during the exploratory phase and after it. This should further increase total expected payoff, and

thereby result in an even more pronounced advantage to exploration. (Such advantage is very hard to come by in our original bandit problem.) It would also be interesting to explore the effects of discounting schedules on this bandit problem.

References

[1] W.R. Thompson, *Biometrika*, **25**, 275, (1933). W.R. Thompson, *Amer. J. Math.*, **57**, 450, (1935).

[2] O.E. Percus, J.K. Percus, *Comput. Biol. Med.*, **14**, 127, (1984). A.J. Petkau, *J. Amer. Statist. Assoc.*, **73**, 328, (1978).

[3] W.K. Viscusi, *Internat. Econom. Rev.*, **20**, 29, (1979).

[4] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press:Ann Arbor, (1975).

[5] W.D. Sudderth, *Dynamic Programming* in *Encyclopedia of Statistical Sciences*, S. Kotz and N.L. Johnson ed., Wiley:New York, (1982).

[6] D. Feldman, *Ann. Math. Statist.*, **33**, 847, (1962).

[7] D.A. Berry, B. Fristedt, *Bandit Problems, Sequential Allocation of Experiments*, Chapman and Hall:London, (1985).

[8] D.H. Wolpert, W.G. Macready, *No Free Lunch Theorems for Search*, in review at *Oper. Res.*, (1995).