NESTED BATCH MODE LEARNING AND STOCHASTIC OPTIMIZATION WITH AN APPLICATION TO SEQUENTIAL MULTI-STAGE TESTING IN MATERIALS SCIENCE *

YINGFEI WANG†, KRISTOFER G. REYES‡, KEITH A. BROWN§, CHAD A. MIRKIN§, AND WARREN B. POWELL‡

Abstract. We consider the nested-batch decision problem where we need to make a first stage choice (e.g. the size of a nanoparticle) after which we then need to run a series of experiments in batch selecting several second stage choices (e.g. testing different densities of the nanoparticle). Since these experiments are time consuming and expensive, we propose to estimate the value of information from the choice of the first stage decision (the size), to help guide the scientist in the selection of the next batch of experiments to run. The batch experiments are designed assuming that we maximize the value of information for an entire batch. The value of information, known as the Knowledge Gradient, requires calculating the expected maximum of a function. Since the calculation of the expected maximum is computationally intractable, we propose a Monte Carlo-based approach to address this hurdle in the context of both the batch and nested-batch problems. We empirically demonstrate the effectiveness of our approach on the material design problem of maximizing output current of a photoactive device, where it is competitive with a fully sequential optimal learning strategy and significantly outperforms pure exploration, pure exploitation and ϵ-greedy strategies with regard to the opportunity cost metric (8.1).

Key words. optimal learning, materials science, sequential design of experiments, decision making, dynamic programming, knowledge gradient

AMS subject classifications. 68T05, 62F07, 62F15, 93E35 , 90C39, 90C40

1. Introduction.

Our work is motivated by problems in the laboratory sciences where we have to select a series of parameters (e.g. size, shape, density and concentration) that guide the design of a material where we are trying to achieve a particular goal (e.g. maximum strength, conductivity, or reflexivity). For example, in this paper we are interested in identifying the density, size and type of nanostructures on the surface of a photoactive device that maximizes output current (see Section 2 for more details). The number of potential parameter settings is much larger than we can explore experimentally, especially when we consider that an experiment can take hours or even days. This is exacerbated by the complication that certain parameters may be more difficult to vary than others in a serial fashion.

There are several factors contributing to this. First is the curse of dimensionality, in which the set of potential experiments (identified by a selection of tunable parameters) increases exponentially with the number of tunable parameters. Second is the continuous nature of certain parameters. For example, the density or concentration of a solute in solution may often be varied within several orders of magnitude, and yet the optimum selection of density could occur within a small window of values. This problem of separation of scales may be naively dealt with by using a refined discretization, which results in a large number of experimental alternatives. Third is

*This research was supported in part by AFOSR grant contract FA9550-12-1-0200 for Natural Materials, Systems and Extremophiles.

†Department of Computer Science, Princeton University, Princeton, NJ 08540 (yingfei@cs.princeton.edu)

‡Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08540 (kreyes@gmail.com, powell@princeton.edu)

§International Institute for Nanotechnology, Northwestern University, Evanston, IL 60208 (brownka@gmail.com, chadnano@northwestern.edu)

the fact that physically, varying one parameter may be more difficult than another. For example, in our reflexivity problem (see Section 2), a selection of "type" and size of nanostructure (e.g. nanorods, nanodots or some other geometrical shape) entails physically fabricating such a structure, which may take a day in the laboratory. Contrast this with a selection of density, which can be varied more readily by an appropriate choice of solution concentration.

Scientists can draw on an extensive body of literature on the classic design of experiments [8, 30, 21] whose goal is to decide what observations to make when fitting a function. Yet in the laboratory settings considered in this paper, the decisions need to be guided by a well-defined objective function (for example, maximizing output current) aside from the intent to learn the fitting. Moreover, many such classical techniques fail to account for practicalities such as the difficulties in performing potentially vastly different experiments in a sequential manner. Previous work [12, 24, 11, 22] develops a sequential design of experiments in a Bayesian setting, but is suited for problems in which experiments are expensive and must be run one at a time.

This sequential design of experiments fails to account for the realities encountered by experimentalists, who may be able to run several parallel experiments in batches. For example, an experimenter can easily vary nanoparticle density over a sample, effectively performing parallel, batch experiments through a single sample. While the idea of batch experimentation is well established throughout all of the physical sciences, recently new experimental tools have provided experimentalists the ability to vary parameters such as surface feature lengths and areas on the nanometer length scale [14]. As a second constraint, it may be difficult or expensive for a scientist to explore the set of experiments in the order prescribed by the knowledge gradient policy, which often suggests consecutively vastly different experiments. For example, the choice of nanoparticle size described above cannot be readily changed between experiments since their fabrication is expensive. The choice of nanoparticle size and density can therefore be modeled as a nested decision in which a nanoparticle size is first selected, and several densities are chosen to maximize the marginal value of information given the fixed nanoparticle size. Such batch and nested batch experimental modes must be taken into consideration in designing a sequence of experiments.

In this paper, we extend the knowledge gradient concept to handle both batch experiments, as well as nested experiments that are performed within a batch. We derive the marginal value of information for each possible experiment which the scientists can use as a guide.

Commonly used sequential decision making policies [3, 5, 12, 16] allocate only one alternative at a time and are not directly applicable to the above mentioned batch and nested batch experimental setting. More relevant to this setting, there exists literature on stochastic and/or adversarial bandit problems addressing the problem of multi-plays (playing several alternatives at the same time), which can be viewed as batch-mode decision making [1, 2, 29, 17, 6, 25, 15]. However, the bandit objective is to maximize the cumulative rewards over time, which is not suitable for our laboratory setting where the objective is to find the controllable parameters that maximizes some utility function. Chen and Krause [7] have studied batch mode active learning and more general information-parallel stochastic optimization problems. But their objective is to let a set function exceed a threshold value while at the same time minimizing the number of items allocated. Moreover, the proposed algorithm in [7] is specifically designed for batch-mode active learning and cannot be generalized to other information-parallel stochastic optimization problems.

The most related models are the stochastic subset selection problems introduced in [26, 27], where the choice in each round is a subset of alternatives while the objective is to find the set of alternatives that maximizes some function on such sets. This differs fundamentally from finding one alternative that maximizes some utility function through batch measurements, as is the case in our setting. In [26, 27], the way to recommend a set of alternatives in each round is to treat each subset of alternatives as a singly super alternative in the space of subsets and construct beliefs over the set function values rather than the function values of the alternatives. The number of subsets with B elements out of M elements is $\binom{M}{B}$, which grows exponentially with the number of alternatives. As the number of alternatives increases, even storing and updating the requisite $\binom{M}{B} \times \binom{M}{B}$ covariance matrix becomes problematic. For example, the size of the choice set considered in [27] was $\binom{10}{5} = 252$. Instead, to address this, we derive the policies presented in this paper with beliefs on the function values of the alternatives whose number is far smaller than the super alternatives.

In this paper, we extend the knowledge gradient policy to address both batch and nested-batch measurements. While the technique presented is generally applicable to the experiments where a batch or nested batch measurement procedure is utilized, we focus on a single motivating application: that of optimizing the output current of a photoactive device whose surface has been functionalized by gold nanoparticles. This problem is presented in Section 2. We then describe the formalism of the optimization problem known as ranking and selection in Section 3 and briefly review simple KG policy for performing the optimization in Section 4. Section 5 describes the proposed formal model for batch and nested-batch setting, while Sections 6 and 7 outlines the adapted KG policy for these cases. Lastly in Section 8, we apply these new algorithms to solving the example optimization problem of maximizing output current. Here we present simulation results to numerically show that the batch and nested-batch algorithms perform well for the model application presented in this paper, finding alternatives that yield large values of output current on average.

2. Motivating application. As a motivating example and as discussed briefly above, we consider a photoactive device in which anisotropic gold nanoparticles (NPs) are immobilized on the surface of the device. The immobilization is performed using a DNA-mediated approach using thiol-gold chemistry in which both the surface and the NPs are functionalized by complimentary DNA strands that subsequently bind to hold the particles onto surface [28]. The NP's role is to enhance the photocurrent of the device via photonic and plasmonic phenomena, thereby potentially increasing the photoelectric efficiency of such a device. Understanding the particular configuration of the device that yield optimal photoactivity is desirable in applications such as efficient solar cells.

Among the tunable parameters that describe the device's configuration are NP size and the density of NPs functionalized onto the surface of the device. Fabrication of NPs of a particular size is done via thermal or photochemical methods, and requires several hours to days to fabricate [20, 18]. In contrast, once NPs are fabricated, it is straightforward to immobilize them onto the device at some prescribed density [28, 23], and often several such densities can be considered in parallel. Therefore, in selecting which configurations (i.e. a choice of NP size and density) to experimentally test, we are naturally led to a nested, batch decision. Different densities may be run in a batch setting, provided that the size of the NPs is the same within the batch.

While the exact mapping between the tunable parameters of NP size and density and the response output current is not well established, we may make some qualitative

statements using domain expert prior knowledge. Specifically, both NP size and density affect the phenomena of surface plasmon resonance, photon absorption, and scattering, which subsequently influence output current. The full description of the effect of the parameters on these physical phenomena and output current is beyond the scope of this paper (see e.g. [9, 13, 10] for a general treatment of discussions on optical and electrical properties of nanostructured devices). However, we state that due to competing effects, there exists a critical value of both NP size and density that optimizes output current, and further assume that there exists a single such extrema in the domain of interest. Our task is to find this critical value under uncertainty of the true physics of the system. To this end, we consider a third-order polynomial approximation of the output current $I(d, \rho)$ with respect to size d and the logarithm of NP density ρ:

$$I(d,\rho) = c_1 + c_2 d + c_3\rho + c_4 d^2 + c_5 d\rho + c_6\rho^2 + c_7 d^3 + c_8 d^2\rho + c_9 d\rho^2 + c_{10}\rho^3. \tag{2.1}$$

This polynomial regression model is meant as a third-order local approximation to the true response function. A cubic polynomial was specifically selected to provide a balance between the accuracy of this approximation without containing too many terms, which would expose the model to overfitting noisy measurements. The unknown regression coefficients c_i are learned along the way through sequential measurements and are subsequently used to provide an estimation of which configurations (d, ρ) optimize I. In what follows, we describe the technique employed to adaptively and iteratively select those configurations to test in order to maximize output current in a nested and batch setting.

3. The Sequential Ranking and Selection Problem. The ranking and selection (R&S) problem is defined as follows. Suppose we have a finite set of alternatives $\mathcal{X} = \{1, 2, ..., M\}$, each of which can be measured sequentially to estimate its constant but unknown underlying mean μ_x. Each element x represents the choices that have to be made when running an experiment. For example, in our motivating application an alternative x is a two dimensional vector representing a particular choice of NP size and density. We begin with a prior multivariate normal distribution of belief about the performance μ_x for each alternative $x \in \mathcal{X}$, $\mu \sim \mathcal{N}(\theta^0, \Sigma^0)$, where $\mu = (\mu_x)_{x\in\mathcal{X}}, \theta^0 = (\theta^0_x)_{x\in\mathcal{X}}$ and Σ^0 is the covariance in our belief about the alternatives.

At the nth iteration (starting with $n = 0$), we choose one alternative x^n to measure. Let ϵ^{n+1} be the measurement error which is assumed to be normally distributed with known variance $\lambda^W = \sigma^2_W$ and is independent conditionally on x^n. We simplify our notation by assuming that our measurement variance is the same across all alternatives, but if this is not the case, we can replace λ^W with λ^W_x throughout. The resulting observation is $W^{n+1} = \mu_{x^n} + \epsilon^{n+1}$, i.e., a noisy perturbation from the truth.

For convenience, we introduce the σ-algebras $\mathcal{F}^n$ for any $n = 0, 1, ..., N-1$ which is formed by the previous n measurement choices and outcomes, $x^0, W^1, ..., x^{n-1}, W^n$. We define $\theta^n = \mathbb{E}[\mu|\mathcal{F}^n]$ and $\Sigma^n = \mathrm{Cov}[\mu|\mathcal{F}^n]$. Then conditionally on $\mathcal{F}^n$, $\mu \sim \mathcal{N}(\theta^n, \Sigma^n)$. By Bayes rule and the Sherman-Morrison formula, taking $x = x^n$ to temporarily simplify subscripts, the updating equations can be written as

$$\theta^{n+1} = \theta^n + \frac{W^{n+1} - \theta^n_x}{\lambda^W + \Sigma^n_{xx}}\Sigma^n e_x, \tag{3.1}$$

$$\Sigma^{n+1} = \Sigma^n - \frac{\Sigma^n e_x (e_x)^T \Sigma^n}{\lambda^W + \Sigma^n_{xx}}, \tag{3.2}$$

where e_x is a vector with 1 at index x and zeros everywhere else. Let $S^n = (\theta^n, \Sigma^n)$ be our state of knowledge. A decision function $X^\pi(S^n)$ is defined as a mapping from the knowledge state to $\mathcal{X}$. We refer to the decision function X^π and the policy π interchangeably.

If we are limited to N measurements, the objective is to maximize the expected reward of the final recommended alternative:

$$\max_{\pi \in \Pi} \mathbb{E}\left[\mu_{x^N}\right], \tag{3.3}$$

where $x^N = \arg\max_{x \in \mathcal{X}} \theta_x^N$ and $x^n = X^\pi(S^n)$ for $n = 0, 1, ..., N-1$.

Alternatively, we can formulate the problem within a dynamic programming framework [11]. Define the state space $\mathcal{S}$ to be the cross-product of $\mathbb{R}^M$ and the space of positive semi-definite matrices. We next define the transition function from the updating equations (3.1) (3.2). Define a vector valued function $\tilde{\sigma}$ as

$$\tilde{\sigma}(\Sigma, x) = \frac{\Sigma e_x}{\sqrt{\lambda^W + \Sigma_{xx}}}, \tag{3.4}$$

where Σ is any covariance matrix. Next define the random variable

$$Z^{n+1} = \frac{W_{x^n}^{n+1} - \theta_{x^n}^n}{\sqrt{\mathrm{Var}[W_{x^n}^{n+1} - \theta_{x^n}^n | \mathcal{F}^n]}},$$

which is a one-dimensional standard normal random variable when conditioned on $\mathcal{F}^n$.

We can write (3.1) as

$$\theta^{n+1} = \theta^n + \tilde{\sigma}(\Sigma^n, x^n) Z^{n+1}. \tag{3.5}$$

Update (3.2) can also be rewritten as

$$\Sigma^{n+1} = \Sigma^n - \tilde{\sigma}(\Sigma^n, x^n)\big(\tilde{\sigma}(\Sigma^n, x^n)\big)^{\mathrm{T}}. \tag{3.6}$$

Now we can define the transition function.

DEFINITION 3.1. *The transition function $T : \mathcal{S} \times \mathcal{X} \times \mathbb{R}$ is defined as*

$$T\bigg((\theta, \Sigma), x, z\bigg) := \bigg(\theta + \tilde{\sigma}(\Sigma, x) z, \Sigma - \tilde{\sigma}(\Sigma, x)\big(\tilde{\sigma}(\Sigma, x)\big)^{\mathrm{T}}\bigg), \tag{3.7}$$

so that $S^{n+1} = T(S^n, x^n, Z^{n+1})$. Here θ is a vector, Σ is a covariance matrix, $z \in \mathbb{R}$ and Z^{n+1} is a one-dimensional standard normal random variable.

We then define the value function $V^n : \mathcal{S} \mapsto \mathbb{R}$ at times $n = 0, 1, ..., N$ as

$$V^n(s) := \max_\pi \mathbb{E}^\pi\big[\max_x \theta_x^N | S^n = s\big], \forall s \in \mathcal{S}. \tag{3.8}$$

By noting that $\max \theta^N$ is $\mathcal{F}^N$-measurable, the terminal value function V^N can be computed directly as:

$$V^N(s) = \max_{x \in \mathcal{X}} \theta_x, \forall s = (\theta, \Sigma) \in \mathcal{S}. \tag{3.9}$$

The dynamic programming principle tells us that the value function at times $n = 0, 1, ..., N-1$, V^n is given recursively by :

$$V^n(s) = \max_{x \in \mathcal{X}} \mathbb{E}[V^{n+1}\big(T(s, x, Z^{n+1})\big)], s \in \mathcal{S}. \tag{3.10}$$

4. The Knowledge Gradient Policy for R&S problems. For R&S problems, the knowledge gradient policy is a stationary policy that at the nth iteration chooses its $(n+1)$st measurement from $\mathcal{X}$ to maximize the single-period expected increase in value [12]. To be more specific,

DEFINITION 4.1. *The knowledge gradient of measuring an alternative x at state s is*

$$\nu_x^{KG}(s) := \mathbb{E}[V^N\big(T(s,x,Z)\big) - V^N(s)], \tag{4.1}$$

where Z is a one-dimensional standard normal random variable. Recall from (3.9) that $V^N(S^n) = \max_{x\in\mathcal{X}} \theta_x^n$. Suppose we are at knowledge state $S^n = (\theta^n, \Sigma^n)$; if we choose to measure $x^n = x$ right now, allowing us to observe W_x^{n+1}, then we transition to a new state of knowledge $S^{n+1} = (\theta^{n+1}, \Sigma^{n+1})$. At iteration n, θ_x^{n+1} is a random variable since we do not yet know what W^{n+1} is going to be. The knowledge gradient of measuring x is then

$$\nu_x^{\mathrm{KG}}(S^n) = \mathbb{E}[\max_{x'} \theta_{x'}^{n+1} - \max_{x'} \theta_{x'}^n | x^n = x, S^n].$$

One property of the knowledge gradient is that $\nu_x^{KG}(s) \geq 0$ for any $s \in \mathcal{S}$ [12]. The knowledge gradient policy will never evaluate an alternative that yields zero value of information.

DEFINITION 4.2. *The Knowledge Gradient (KG) policy is defined as:*

$$X^{KG}(S^n) = \arg\max_{x\in\mathcal{X}} \nu_x^{KG}(S^n).$$

The algorithm for calculating the knowledge gradient can be found in [11].

The knowledge gradient policy can handle a variety of belief models such as linear [22] or nonparametric [22, 19, 4].

5. From Sequential Decision Making to Nested Batch Mode Decision Making. In this section, we first give the formal model for batch learning and then we will extend it to nested batch mode decision making.

5.1. Batch Mode Learning Model. In real world applications, it often occurs that information collectors do not simply take one measurement at a time. For example, in a pharmaceutical company, researchers might test the efficiency of a medicine by taking measurements of five different concentrations simultaneously, observing all the outcomes, and then measuring the next five concentrations. Or in the motivating application, if we fix a NP size, then we can test on different densities simultaneously. This leads us to the idea of batch measurements.

Suppose we have a collection $\mathcal{X} = \{1, 2, ..., M\}$ of M alternatives. Instead of sequentially measuring some alternatives to estimate the constant but unknown underlying mean μ_x, we can measure a batch of alternatives simultaneously at each step. We begin with a prior multivariate normal distribution of belief about the performance μ_x for each alternative $x \in \mathcal{X}$, $\mu \sim \mathcal{N}(\theta^0, \Sigma^0)$, where $\mu = (\mu_x)_{x\in\mathcal{X}}, \theta^0 = (\theta_x^0)_{x\in\mathcal{X}}$ and Σ^0 is the covariance in our belief about the alternatives. Denote the batch size by B and the total number of batches by K. Then the total number of measurements allowed is $N = BK$. At the kth batch (starting with $n = 0$), instead of choosing one alternative to measure as in Section 3, we choose to measure B alternatives $x^{k,0}, x^{k,1}, ..., x^{k,B-1}$. Let ϵ^{k+1} be the measurement error which is assumed to be normally distributed with known variance $\lambda^W = \sigma_W^2$. The resulting observations are $W^{k+1,0} \sim \mathcal{N}(\mu_{x^{k,0}}, \sigma_W), W^{k+1,1} \sim \mathcal{N}(\mu_{x^{k,1}}, \sigma_W), ..., W^{k+1,B-1} \sim \mathcal{N}(\mu_{x^{k,B-1}}, \sigma_W)$.

We modify our notations to fit batch measurements. The superscript (k,b) for some $k = 0, 1, ..., K-1$ and $b = 1, 2, ..., B-1$ should be understood as meaning that we have done k batches and use $x^{k,0}, ..., x^{k,b-1}, W^{k+1,0}, W^{k+1,...,b-1}$ to update our belief. Thus the prior multivariate normal belief can be rewritten as $(\theta^{0,0}, \Sigma^{0,0})$. The new updating equations can be written as

$$\theta^{k,b+1} = \theta^{k,0} + \sum_{j=0}^{b} \frac{W^{k+1,j} - \theta^{k,j}_{x^{k,j}}}{\lambda^W + \Sigma^{k,j}_{x^{k,j}x^{k,j}}} \Sigma^{k,j} e_{x^{k,j}}, \tag{5.1}$$

$$\Sigma^{k,b+1} = \Sigma^{k,b} - \frac{\Sigma^{k,b} e_{x^{k,b}} (e_{x^{k,b}})^T \Sigma^{k,b}}{\lambda^W + \Sigma^{k,b}_{x^{k,b}x^{k,b}}}, \tag{5.2}$$

where $k = 0, 1, ..., K-1$, $b = 0, 1, ..., B-1$, $\theta^{k+1,0} = \theta^{k,B}$ and $\Sigma^{k+1,0} = \Sigma^{k,B}$. It is worth emphasizing that in the batch setting the covariance matrix would be updated within a batch since it is determined by the measurement decisions and is independent of the observations, whereas the mean values θ^n are only updated after the observations are collected for the whole batch. Additionally, the updating formula (5.1) is not affected by whether the observations are obtained sequentially or in batch.

A decision function $X^\pi(S^n)$ is defined as a mapping from the knowledge states to $\mathcal{X}^B$, where S^n is short for $S^{n,0} = (\theta^{n,0}, \Sigma^{n,0})$.

If we are limited to $N = KB$ measurements, the objective is to maximize the expected reward of the final recommended alternative:

$$\max_{\pi \in \Pi} \mathbb{E}\left[\mu_{x^K}\right], \tag{5.3}$$

where $x^K = \arg\max_{x \in \mathcal{X}} \theta^K_x$ and $\{x^{k,0}, ..., x^{k,B-1}\} = X^\pi(S^k)$ for $k = 0, 1, ..., K-1$.

We can also formulate the problem within a dynamic programming framework. We first define the transition function from the updating equations.

For convenience, we introduce the σ-algebras $\mathcal{F}^{k,b}$ for any $b = 0, 1, ..., B-1$ which is formed by the previous k batch measurement outcomes and the first b observations in the current batch. The idea is that even when performing experiments in batch, we can model the updating as if each outcome is collected sequentially. Suppose we are at the $k+1$th batch and have made the measurement decisions for the whole batch. For any $b = 0, 1, ..., B-1$, define the random variable $Z^{k+1,b}$ as

$$Z^{k+1,b} := \frac{W^{k+1,b} - \theta^{k,b}_{x^{k,b}}}{\sqrt{\mathrm{Var}[W^{k+1,b} - \theta^{k,b}_{x^{k,b}} | \mathcal{F}^{k,b}]}}.$$

Since $\theta^{k,b}_x \in \mathcal{F}^{k,b}$,

$$\mathrm{Var}[W^{k+1,b} - \theta^{k,b}_{x^{k,b}} | \mathcal{F}^{k,b}] = \mathrm{Var}[\mu_{x^{k,b}} + \epsilon^{k+1} | \mathcal{F}^{k,b}] = \Sigma^{k,b}_{x^{k,b}x^{k,b}} + \lambda^W.$$

It is important to note that if conditioned on $Z^{k+1,0}, ..., Z^{k+1,b-1}$, or in other words, $\mathcal{F}^{k,b}$, the $Z^{k+1,b}$ is a standard normal distribution.

Recalling from (3.4) the definition of $\tilde{\sigma}$, we can rewrite (5.1) and (5.2) as

$$\theta^{k,b+1} = \theta^{k,0} + \sum_{j=0}^{b} \tilde{\sigma}(\Sigma^{k,j}, x^{k,j}) Z^{k+1,j}, \tag{5.4}$$

$$\Sigma^{k,b+1} = \Sigma^{k,b} - \tilde{\sigma}(\Sigma^{k,b}, x^{k,b}) \big(\tilde{\sigma}(\Sigma^{k,b}, x^{k,b})\big)^{\mathrm{T}}. \tag{5.5}$$

Now we can define the transition function for batch mode learning recursively by pretending the outcomes are obtained sequentially.

DEFINITION 5.1. *The transition function* $T^B : \mathcal{S} \times \mathcal{X}^B \times \mathbb{R}^B$ *is defined as*

$$T^B\Big((\theta,\Sigma),(x_1,...,x_B),(z_1,...,z_B)\Big) := T(...T((\theta,\Sigma),x_1,z_1),..,x_B,z_B), \tag{5.6}$$

so that $S^{k+1,0} = T^B\big(S^{k,0},(x^{k,0},...,x^{k,B-1}),(Z^{k+1,0},...,Z^{k+1,B-1})\big)$. *Here* θ *is a vector,* Σ *is a covariance matrix,* $z^{k+1,j} \in \mathbb{R}$, $Z^{k+1,j}$ *is a one-dimensional standard normal random variable and* T *is the transition function defined in Definition 3.1.*

We then define the value function $V^{B,k} : \mathcal{S} \mapsto \mathbb{R}$ after k batch measurements at times $k = 0, 1, ..., K-1$ as

$$V^{B,k}(s) := \max_{\pi} \mathbb{E}^{\pi}\big[\max_{x} \theta_x^K | S^k = s\big], \forall s \in \mathcal{S}.$$

By noting that θ^K is deterministic given S^K, the terminal value function $V^{B,K}$ can be computed directly as:

$$V^{B,K}(s) = \max_{x\in\mathcal{X}} \theta_x, \forall s = (\theta,\Sigma) \in \mathcal{S}. \tag{5.7}$$

The dynamic programming principle tells us that the value function at times $k = 0, 1, ..., K-1$, $V^{B,k}$ is given recursively by :

$$V^{B,k}(s) = \max_{(x_i)_{i=1}^B \in \mathcal{X}^B} \mathbb{E}\big[V^{B,k+1}\big(T^{\mathrm{B}}(s,(x_i)_{i=1}^B,(Z_i)_{i=1}^B\big)\big], s \in \mathcal{S}, \tag{5.8}$$

where Z_i is a one dimensional standard normal variable.

A Knowledge-Gradient policy is provided for batch learning model in section 6.

5.2. Nested Batch Mode Learning Model. Motivated by the applications given by the real world applications in Section 2, right now we have a collection $\mathcal{X}_1 \times \mathcal{X}_2$ of M alternatives, where at each decision step, we choose one $x \in \mathcal{X}_1$ and a set $\mathcal{Y} \in \mathcal{X}_2^B$, constructing B alternatives to measure simultaneously (e.g. design 10nm triangle particles and experiment with densities of 3%, 10%, 27%, 78% and 92% with a batch size $B = 5$).

As before, we begin with a prior multivariate normal distribution of belief about the performance $\mu_{(x,y)}$ for each alternative $x \in \mathcal{X}_1$ and $y \in \mathcal{X}_2$, $\mu \sim \mathcal{N}(\theta^0, \Sigma^0)$, where $\mu = (\mu_{(x,y)})_{(x,y)\in\mathcal{X}_1\times\mathcal{X}_2}$, $\theta^0 = (\theta^0_{(x,y)})_{(x,y)\in\mathcal{X}_1\times\mathcal{X}_2}$ and Σ^0 is a $M \times M$ covariance matrix.

Let K be the total number of batches. At any decision step $k = 0, 1, ..., K-1$ after we make the B measurement decisions $(x^k, y^{k,0}), (x^k, y^{k,1}), ...(x^k, y^{k,B-1})$ and get their outcomes, we can also pretend that the information is collected sequentially. So the updating equations are the same as those in the batch mode model when treating (x, y) as the alternative and replacing $x^{k,j}$ with $(x^k, y^{k,j})$. It is worth noting here, we are not only updating our belief about the alternatives with x^k, but we are also updating our belief about all M alternatives.

A decision function $X^\pi(S^n)$ is defined as a mapping from the knowledge state to $\mathcal{X}_1 \times \mathcal{X}_2^B$. The objective is to maximize the expected reward of the final recommended alternative:

$$\max_{\pi\in\Pi} \mathbb{E}\left[\mu_{(x^K,y^K)}\right], \tag{5.9}$$

where $(x^K, y^K) = \arg\max_{(x,y)\in\mathcal{X}_1\times\mathcal{X}_2} \theta^K_{(x,y)}$ and $\{x^k, y^{k,0}, ..., y^{k,B-1}\} = X^\pi(S^k)$ for $k = 0, 1, ..., K-1$.

We formulate the problem within a dynamic programming framework. By a similar argument as that in batch mode, we can define the transition function as

DEFINITION 5.2. *Define the transition function* $T^{NB}: \mathcal{S} \times (\mathcal{X}_1 \times \mathcal{X}_2^B) \times \mathbb{R}^B$ *as*

(5.10)
$$T^{NB}\Big((\theta,\Sigma),(x,y_1,...,y_B),(z_1,...,z_B)\Big) := T\big(...T\big((\theta,\Sigma),(x,y_1),z_1\big),..,(x,y_B),z_B\big),$$

so that $S^{k+1} = T^{NB}\big(S^k,(x^k,y^{k,0},...,y^{k,B-1}),(Z^{k+1,0},...,Z^{k+1,B-1})\big)$. *Here* θ *is a vector,* Σ *is a covariance matrix,* $z^{k+1,j} \in \mathbb{R}$, $Z^{k+1,j}$ *is a one-dimensional standard normal random variable and* T *is the transition function defined in Definition 3.1.*

We then define the value function $V^{NB,k}: \mathcal{S} \mapsto \mathbb{R}$ after k nested batch measurements at times $k = 0, 1, ..., K-1$ as

$$V^{NB,k}(s) := \max_{\pi} \mathbb{E}^{\pi}\big[\max_{(x,y)} \theta^K_{(x,y)} | S^k = s\big], \forall s \in \mathcal{S}.$$

The terminal value function $V^{NB,K}$ can be computed directly as:

$$V^{NB,K}(s) = \max_{(x,y)\in\mathcal{X}_1\times\mathcal{X}_2} \theta_{(x,y)}, \forall s = (\theta,\Sigma) \in \mathcal{S}. \tag{5.11}$$

The dynamic programming principle tells us that the value function at times $k = 0, 1, ..., K-1$, $V^{B,k}$ is given recursively by :

(5.12)
$$V^{NB,k}(s) = \max_{(x,\mathcal{Y})\in\mathcal{X}_1\times\mathcal{X}_2^B} \mathbb{E}\big[V^{NB,k+1}\big(T^{\mathrm{NB}}(s,(x,y_1,...y_B),(Z_1,...Z_B))\big)\big], s \in \mathcal{S},$$

where Z_i is a one dimensional standard normal variable.

A KG-type policy is provided for nested batch learning in the section 7.

6. Batch Knowledge Gradient (BKG) Policy. In this section, we extend the original idea of the KG policy for batch mode learning. We first give the formal definition of the batch knowledge gradient policy and then provide a Monte Carlo algorithm for any given batch size.

6.1. Definition of BKG Policy. Following the basic idea of the knowledge gradient, we would like to design a policy that seeks to measure the B alternatives that provide the single-period expected increment as a batch. We first define the value of information from measuring a batch of alternatives.

DEFINITION 6.1. *The knowledge gradient for measuring a batch of* j *alternatives* $\{x_1,...x_j\}$ *at state* s *is defined as*

$$\nu^{BKG}_{x_1,...x_j}(s) := \mathbb{E}\left[V^{B,K}\big(T^B(s,(x_1,...x_j),(Z_1,...,Z_j))\big) - V^{B,K}(s)\right], \tag{6.1}$$

where Z_i *is a one-dimensional standard normal random variable.*

Recall from (5.7) that $V^{B,K}(S^k) = \max_{x\in\mathcal{X}} \theta^k_x$. Thus, suppose we are in knowledge state $S^k = (\theta^k,\Sigma^k) = (\theta^{k,0},\Sigma^{k,0})$. If we choose to measure $(x^{k,0} = x_1, ..., x^{k,j-1} = x_j)$ right now, allowing us to observe $(W^{k+1,0}_{x^{k,0}},...,W^{k+1,j-1}_{x^{k,j-1}})$, then we transition to a new state of knowledge $S^{n+1} = (\theta^{n+1},\Sigma^{n+1})$. At iteration k, θ^{k+1} is a random vector since we do not yet know what W^{k+1} is going to be. The knowledge gradient of measuring $(x_1,...,x_j)$ is then

$$\nu^{\mathrm{BKG}}_{x_1,...x_j}(S^k) = \mathbb{E}[\max_x \theta^{k+1}_x - \max_x \theta^k_x | x^{k,0} = x_1, \cdots, x^{k,j-1} = x_j, S^k]. \tag{6.2}$$

One way to design a policy π' using the knowledge gradient concept is to directly find the $\{x_1, ..., x_j\}$ that maximizes $\nu^{\text{BKG}}_{x_1,...x_j}(S^k)$ subject to $j \leq B$. Since the measurement is noisy, measuring the same x_i several times will most likely give different observations and thus it is meaningful if we measure some alternative x_i more than once within a batch. For example, in the motivating application, we can choose to test on 5 densities $(\rho_1, \rho_1, \rho_3, \rho_3, \rho_7)$ all at once. Thus the batch decision procedure is analogous to multi-set function maximization problems. Let $\mathcal{X}$ be a finite set of M elements. Define the multi-set function $f : \mathbb{N}^{\mathcal{X}} \mapsto \mathbb{R}$. The problem is to find a multi-set A of cardinality less than or equal to some specified number B, such that $f(A)$ is the maximum:

$$\max_{A \subset \mathbb{N}^{\mathcal{X}}} \{f(A) : |A| \leq B\}. \tag{6.3}$$

The problem with π' is that it involves testing all $\sum_{b=0}^{B-1} \binom{b+M-1}{M-1}$ which would be computationally costly when B and M are large. Alternatively, as a common technique to deal with set function maximization problems, we can use a greedy heuristic to start from the null set and add elements one at a time. We first claim that the more measurements, the larger the value of information. Thus, if we are limited to B measurements in a batch, we will indeed measure B alternatives in each batch.

PROPOSITION 6.2. **(Benefits of Measurement)**
$\nu^{BKG}_{x_1,...x_{j+1}}(s) \geq \nu^{BKG}_{x_1,...x_j}(s)$ for all $j \geq 0$, $s \in \mathcal{S}$ and $x_i \in \mathcal{X}$.

Proof. In the following proof, we use properties of conditional expectations $\mathbb{E}[\mathbb{E}[U|V]] = \mathbb{E}[U]$ for any random variables U and V.

$$\begin{aligned}
&\nu^{\text{BKG}}_{x_1,...x_{j+1}}(s) - \nu^{\text{BKG}}_{x_1,...x_j}(s) \\
&= \mathbb{E}\left[V^{B,K}\big(T^{\text{B}}(s, (x_i)_{i=1}^{j+1}, (Z_i)_{i=1}^{j+1})\big) - V^{B,K}\big(T^{\text{B}}(s, (x_i)_{i=1}^{j}, (Z_i)_{i=1}^{j})\big)\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[V^{B,K}\big(T^{\text{B}}(s, (x_i)_{i=1}^{j+1}, (Z_i)_{i=1}^{j+1})\big) - V^{B,K}\big(T^{\text{B}}(s, (x_i)_{i=1}^{j}, (Z_i)_{i=1}^{j})\big)|(x_i)_{i=1}^{j}, (z_i)_{i=1}^{j}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[V^{B,K}\big(T(s', x_{j+1}, Z_{j+1})\big) - V^{B,K}(s')\right]\right],
\end{aligned}$$

where $s' = T^{\text{B}}\big(s, (x_i)_{i=1}^{j}, (z_i)_{i=1}^{j}\big)$, $T(s, x, z)$ is the transition function defined in Definition 3.1 and in the last equation the first expectation is taken over the random choices of s' or equivalently the choices of $(z_i)_{i=1}^{j}$ and the second expectation is taken over Z_{j+1}. By the definition of T and $V^{B,K}$, we have $V^{B,K}\big(T(s', x_{j+1}, Z_{j+1})\big) = \max_{x \in \mathcal{X}} \big(\theta'_x + \tilde{\sigma}_x(\Sigma', x_{j+1})Z_{j+1}\big)$ and $V^{B,K}(s') = \max_x \theta'_x$. By Jensen's inequality, we have

$$\begin{aligned}
\mathbb{E}\left[V^{B,K}\big(T(s', x_{j+1}, Z_{j+1})\big)\right] &= \mathbb{E}\left[\max_{x \in \mathcal{X}} \big(\theta'_x + \tilde{\sigma}_x(\Sigma', x_{j+1})Z_{j+1}\big)\right] \\
&\geq \max_{x \in \mathcal{X}} \mathbb{E}\left[\big(\theta'_x + \tilde{\sigma}_x(\Sigma', x_{j+1})Z_{j+1}\big)\right] \\
&= \max_{x \in \mathcal{X}} \theta'_x \\
&= V^{B,K}(s').
\end{aligned}$$

Since this inequality holds for any realization of s', the proposition follows. □

COROLLARY 6.3. *The knowledge gradient of measuring a batch of j alternatives at any state s is always non-negative, $\nu^{BKG}_{x_1,...x_j}(s)$ for all $j \geq 0$, $s \in \mathcal{S}$ and $x_i \in \mathcal{X}$.*

Proof. It follows from Proposition 6.2 by noting that $\nu^{\text{BKG}}_{\emptyset}(s) = 0$ for any $s \in \mathcal{S}$. □

Since the more measurements the better, if we are limited to at most B measurements at each time step, we will exactly choose to make B measurements. We thus can define the batch knowledge gradient (BKG) policy that greedily adds in each alternative that maximizes the expected increment of value one at a time until B alternatives are chosen.

DEFINITION 6.4. *The Batch Knowledge Gradient (BKG) policy has the decision function*

$$x^{k,b} := X_b^{BKG}(S^k) = \arg\max_{x\in\mathcal{X}} \nu^{BKG}_{x^{k,0},\ldots,x^{k,b-1},x^{k,b}=x}(S^k), \tag{6.4}$$

for any $b = 0, ..., B-1$ *and decision points* $k = 0, 1, ..., K-1$.

The above formulation tells us that we make each measurement decision in the batch by conditioning on the earlier decisions made in the same batch and the state. With (5.4) and (6.2), we can rewrite (6.4) as

(6.5)
$$X_b^{\mathrm{BKG}}(S^k) = \arg\max_{x\in\mathcal{X}} \mathbb{E}\left[\max_{x'}\left(\theta^{k,0} + \sum_{j=0}^{b-1}\tilde{\sigma}(\Sigma^{k,j}, x^{k,j})Z^{k+1,j} + \tilde{\sigma}(\Sigma^{k,b}, x)Z^{k+1,b}\right)\right],$$

where $x^{k,j}$, $j \le b$ are fixed when choosing $x^{k,b}$ and $\Sigma^{k,j}$ can be updated within a batch according to (5.2). This formula will be of use in the following computations.

6.2. Computation. We notice from (6.4) that at each batch decision point k, we can find the first measurement decision explicitly by carrying out the original KG calculation described since the objective function (6.5) to be maximized for the first decision in the batch is exactly the same as that described in Section 4.

Since an analytic expression for the expected maximization as in (6.5) is unknown, we utilize Monte Carlo sampling to approximate the expectation. After the first measurement decision $x^{k,0}$ is made, the following decisions are made one at a time to find $x^{k,b}$ according to (6.5) using Monte Carlo Simulation. To be more specific, the second decision is made by randomly generating both $Z^{k+1,0}$ and $Z^{k+1,1}$ for Q times, where $Z^{k+1,i}$ are independent standard normal variables. We then define the second decision $x^{k,1}$ as:

$$\arg\max_{x\in\mathcal{X}} \frac{1}{Q}\sum_{q=1}^{Q}\left[\max_{x'}\left(\theta^{k,0} + \tilde{\sigma}(\Sigma^{k,0}, x^{k,0})z_q^0 + \tilde{\sigma}(\Sigma^{k,1}, x)z_q^1\right)\right],$$

where z_q^0 and z_q^1 are realizations of $Z^{k+1,0}$ and $Z^{k+1,1}$ respectively and $\Sigma^{k,1}$ is updated according to (5.2). We then have $x^{k,0}$ and $x^{k,1}$ fixed, and proceed to find $x^{k,2}$ similarly by sampling $Z^{k+1,0}$, $Z^{k+1,1}$ and $Z^{k+1,2}$ for Q times and finding the alternative that maximizes the analogous expression coming from (6.5) that contains these three random variables. It is worth re-emphasizing here that all three variables are standard normal when we generate their realizations after fixing the previous decisions.

In general, after we get the first b decisions within a batch, we are looking to find the solution to

$$x^{k,b} = \arg\max_{x\in\mathcal{X}} \frac{1}{Q}\sum_{q=1}^{Q}\left[\max_{x'}\left(\theta^{k,0} + \sum_{j=0}^{b-1}\tilde{\sigma}(\Sigma^{k,j}, x^{k,j})z_q^j + \tilde{\sigma}(\Sigma^{k,b}, x)z_q^b\right)\right],$$

where $\Sigma^{k,j}$ are updated within this batch according to (5.2).

The pseudo-code of the algorithms are presented below. Algorithm 1 is the BKG policy for the kth batch decision, which calls Algorithm 2 to find the next measurement decision for B times.

Algorithm 1: Batch Knowledge Gradient Policy

input : $\theta^{k,0}, \Sigma^{k,0}$ and the number of sample Q for the Monte Carlo simulation
Use the sequential KG policy presented in Section 4 to find $x^{k,0}$;
$\tilde{\sigma}^0 \leftarrow \tilde{\sigma}(\Sigma^{k,0}, x^{k,0})$;
Update $\Sigma^{k,1}$ according to (5.2);
for $b = 1$ *to* $B-1$ **do**
  Use Algorithm 2 below to find $x^{k,b}$;
  $\tilde{\sigma}^b \leftarrow \tilde{\sigma}(\Sigma^{k,b}, x^{k,b})$;
  Update $\Sigma^{k,b+1}$ according to (5.2);
end
output: batch decisions $x^{k,0}, x^{k,1}, ..., x^{k,B-1}$

Algorithm 2: Monte Carlo Simulation for the $(b+1)$th decision within a batch

input : b, $\theta^{k,0}, \tilde{\sigma}^0, \tilde{\sigma}^1, ..., \tilde{\sigma}^{b-1}, \Sigma^{k,b}$ and Q
for *each* $x \in \mathcal{X}$ **do**
  $\text{sum}_x = 0$;
  for $q = 1$ *to* Q **do**
    for $j = 0$ *to* b **do**
      Generate a realization z_q^j of $Z^{k,j}$;
    end
    temp $\leftarrow \max_{x'} \left(\theta_{x'}^{k,0} + \sum_{j=0}^{b-1} \tilde{\sigma}_{x'}^j z_q^j + \tilde{\sigma}(\Sigma^{k,b}, x) z_q^b\right)$;
    $\text{sum}_x \leftarrow \text{sum}_x +$ temp;
  end
end
$x^{k,b} = \arg\max_{x \in \mathcal{X}} \text{sum}_x$;
output: the $b+1$th decision $x^{k,b}$ within the batch

7. Nested Batch Knowledge Gradient (NBKG) Policy. A nested batch decision may involve the selection of a particular NP size and subsequent selection of several NP densities (given the NP size fixed in the first stage of the decision). In general, we would like to design a policy that seeks to measure the B alternatives $(x, y_1), ..., (x, y_B)$ that provide the largest single period value of information. We first define the knowledge gradient of measuring a nested batch of alternatives.

DEFINITION 7.1. *The knowledge gradient of measuring a nested batch of j alternatives $\{(x, y_1), ..., (x, y_j)\}$ for any $x \in \mathcal{X}_1$ and $y_i \in \mathcal{X}_2$ at state s is defined as*

$$\nu^{NBKG}_{x;y_1,...y_j}(s) := \mathbb{E}\left[V^{NB,K}\left(T^{NB}(s, (x, y_1, ...y_j), (Z_1, ..., Z_j))\right) - V^{NB,K}(s)\right], \tag{7.1}$$

where Z_i is a one-dimensional standard normal random variable.

By a similar argument as Proposition 6.2, we can show that if we are limited to B measurements in a batch we will indeed evaluate B alternatives.

We define the Nested Batch Knowledge Gradient policy as directly finding out $\{x, y_1, ..., y_B\}$ that maximizes $\nu^{\text{NBKG}}_{x;y_1,...y_j}(s)$ at any decision point $k = 0, 1, ..., K$. For clarity, we use $\mathcal{Y}$ to denote the multi-set $\{y_1, ..., y_B\}$ since the alternatives being measured in each batch are not necessarily distinct.

DEFINITION 7.2. *The Nested Batch Knowledge Gradient (NBKG) policy has the*

decision function

$$X^{NBKG}(S^k) = \arg\max_{(x,\mathcal{Y})} \nu^{NBKG}_{x;\mathcal{Y}}(S^k), \tag{7.2}$$

for any decision points $k = 0, 1, ..., K-1$.

We can show analytically that

$$\begin{cases} x^* &= \arg\max_x \left(\max_{\mathcal{Y}} \nu^{\mathrm{NBKG}}_{x;\mathcal{Y}} \right) \\ \mathcal{Y}^* &= \arg\max_{\mathcal{Y}} \nu^{\mathrm{NBKG}}_{x^*;\mathcal{Y}} \end{cases}$$

is a solution to the optimization problem (7.2). This gives us a two-stage decision process. At the first step, for each $x \in \mathcal{X}_1$, find the multi-set (a batch) $\mathcal{Y}_x$ that gives the most value of information; i.e. $\max_{\mathcal{Y}} \nu^{\mathrm{NBKG}}_{x;\mathcal{Y}}$. This can be done by using the Batch Knowledge Gradient policy for each fixed x with the value function $\nu^{\mathrm{NBKG}}_{x;y_1,\ldots,y_B}$ instead of ν^{BKG}. Namely, for example, when calculating a similar expression as (6.2):

$$\nu^{\mathrm{NBKG}}_{x;y_1,\ldots y_j}(S^k) = \mathbb{E}[\max_{(x',y')} \theta^{n+1}_{(x',y')} - \max_{(x',y')} \theta^{n}_{(x',y')} | x^k = x, y^{k,0} = y_1, \cdots, y^{k,j-1} = y_j, S^k], \tag{7.3}$$

it should be noted that even though the BKG is constructed for each $x \in \mathcal{X}_1$, when taking the maximization inside the expectation, x', y' should include all the choices in the domain $\mathcal{X}_1 \times \mathcal{X}_2$. Since calculating the expected maximum is needed to make the decision, Monte Carlo sampling is used as in Algorithm 1 to approximate the expectation.

We next define the nested knowledge gradient ν^{NKG}_x for each $x \in \mathcal{X}_1$ at state s in the nested dimensions as

$$\nu^{\mathrm{NKG}}_x(s) = \max_{\mathcal{Y}} \nu^{\mathrm{NBKG}}_{x;\mathcal{Y}}(s). \tag{7.4}$$

8. Numerical Experiments on NBKG and Optimizing Photocurrent. In this section, we present simulation results for the material science application described in Section 2: optimizing the photocurrent of a photoactive device that has anisotropic nanoparticles immobilized onto its surface. Recall in Equation (2.1) we had approximated the output current by a third-order polynomial expansion in the variables d and ρ, respectively representing NP size (units nm) and log-density. We wish to optimize the output current with respect to these two variables under the uncertainty of the regression coefficients c_i of this expansion. In the physical setting, preparing NPs of a particular size is expensive, while varying the density of a NP can be done easily and in parallel experiments. Therefore, we model the choice of experiment to perform as a nested-batch decision, and apply the NBKG policy toward finding the optimal choice of size and density.

8.1. Prior Generation. To generate a prior distribution on the values of the regression coefficients, we incorporate the following observations, which reflect a domain expert's prior knowledge about the role of NP size and density on output current. First, the experimental range of size was assumed to be between 550 nm to 1300 nm, while the range for NP density was assumed to be between 1 NP/cm^2 to 10^{15} NP/cm^2. When $d = 550$ nm and $\rho = 0$, the output current is simply the output current of the photoactive device non-functionalized with NPs. We presume that this current is scaled to 1 nanoamp (nA). For extreme values where $d = 1300$ nm and $\rho = 15$, we assume the current is a nominally small value 0.001 nA. Lastly, we presume that for

points away from the extremes, the current has moderate values between 1 and 20 nA.

Prior generation was performed by uniformly sampling values

$$d_1 = 550 \leq d_2 \leq d_3 \leq d_4 = 1300 \text{ nm},$$

and

$$\rho_1 = 0 \leq \rho_2 \leq \rho_3 \leq \rho_4 = 15.$$

Sixteen points of the form $(d_i, \rho_j, I(i) + I(j))$ were calculated, where

$$I(i) = \begin{cases} 0.5 & i = 1; \\ 1 & i = 2, 3; \\ 0.0005 & i = 4. \end{cases}$$

We then computed the least-squares fit of the polynomial model in Equation (2.1) to these points, and obtained an instance of the regression parameters c_i. This procedure was repeated several times, resulting in an empirical distribution on c, which we use as the regression parameters' prior distribution. From this, we obtain the induced prior distribution on function values that incorporate the domain expert's prior (albeit limited) knowledge about the behavior of the photocurrent with respect to NP size and density. Figure 1 plots several instances of $I(d, \rho)$ obtained in the above manner.

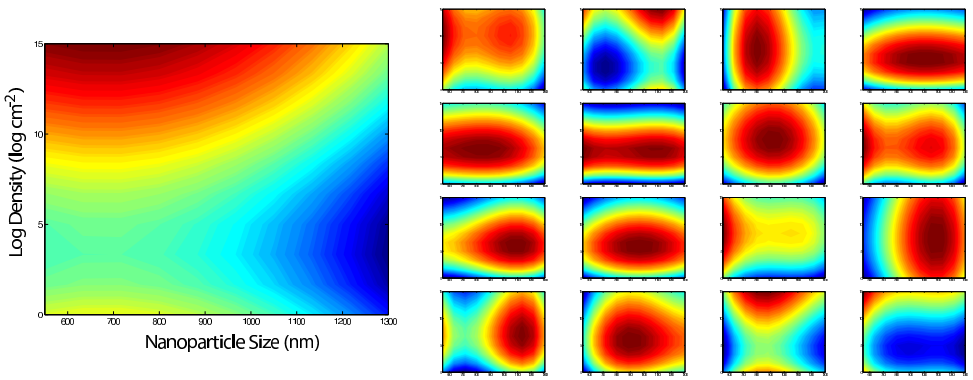
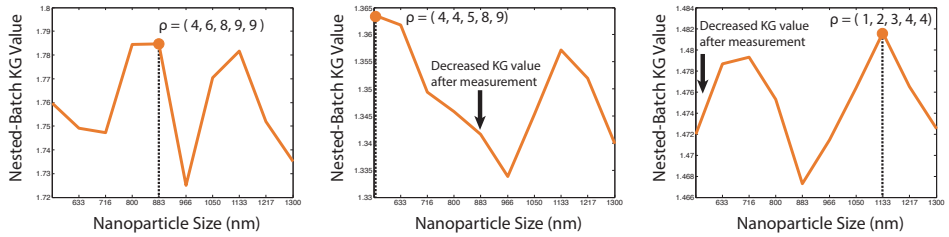


Fig. 1: Example plots of photocurrent $I(d, \rho)$ obtained from the procedure outlined above.

8.2. Performance of NBKG. In order to assess the performance of the N-BKG policy, we performed several numerical experiments in which the decision-measurement-update loop was simulated over several batch measurements and over several trials. For each simulation trial, a true value of the regression parameters (and hence a true response surface) was fixed, but unknown to the simulation.

8.2.1. Illustration on NBKG Policy. We first illustrate how NBKG works under a measurement noise of 30% of the function range. At each iteration, a NBKG value was calculated for each choice of NP size. Example NBKG values are depicted in Figure 2, which is an example of NBKG values after zero, one and two measurements,

respectively. The optimal NP size and corresponding batch of log density values are given at each step. The figure also illustrates a key feature of the KG policy, as shown by a marked decrease in the relative KG value of a NP size after it has been measured. Due to correlation, the values of measuring adjacent alternatives also drops since they roughly provide similar information. As shown in Figure 2, the KG value for NP size = 800nm drops after measuring NP size = 883nm. This gives the KG policy the ability to explore parameter space during the initial set of measurements. From the KG values, the optimal NP size and five corresponding NP densities were selected in the nested-batch method outlined above. After a noisy measurement is made from the true surface, the posterior distribution on μ is calculated according to Equations (3.1). This process is repeated until 15 batch measurements are made.



(a) 0 batch measurements (b) 1 batch measurement (c) 2 batch measurements

Fig. 2: NBKG values before and after 3 batch measurements. The optimal NP size at each step is indicated by the dashed line, and the corresponding optimal batch of densities are also shown. The arrows indicate the decrease in KG value for the NP size that was previously measured.

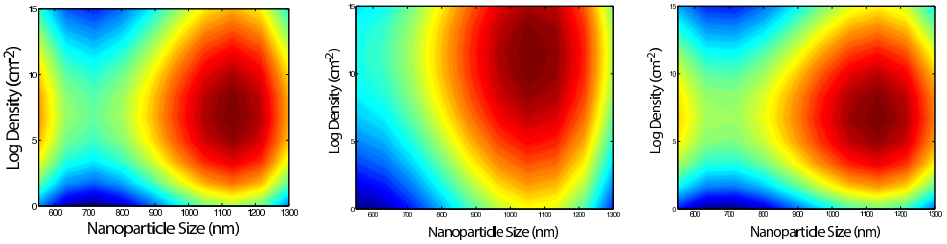
Figure 3 together with Figure 4 shows an example of the prior and posterior estimates of the true photocurrent function for a particular simulation. In Figure 3, the leftmost figure depicts the true photocurrent function values. The middle and rightmost figures demonstrate the prior and posterior estimates of the true function surface after 0 and 15 batch measurements, respectively, using the NBKG policy. Also depicted in Figure 4 is the residual error, which is the difference between the estimate and true function values. The residual errors are calculated after 0, 5, 10 or 15 batch measurements. By examining the residual error plot after 15 measurements, we see that the function value at the true maximum alternative is well approximated, while moderate error in the estimate is located away from this region of interest.

8.2.2. Computational Analysis. In this section, we analyze the performance of NBKG as parameters vary. As a more quantitative measure of the performance of NBKG, we consider the opportunity cost (OC) as a function of the number of batch measurements K:

$$\mathrm{OC}^K = \max_{(x,y)} \mu_{(x,y)} - \mu_{(x^K,y^K)}, \tag{8.1}$$

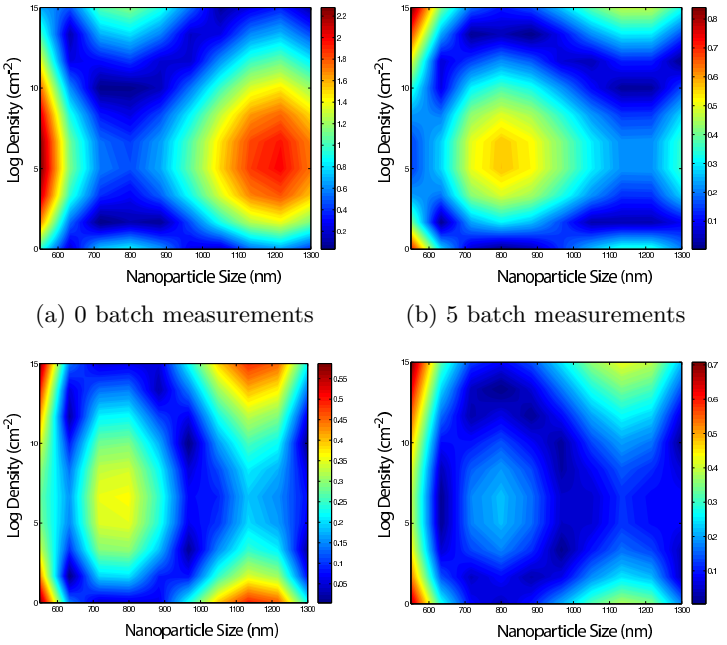
where $(x^K, y^K) = \arg\max_{(x,y)} \theta^K_{(x,y)}$.

Figure 5 shows the mean OC versus number of batch measurements, averaged over 500 simulation trials. In Figure 5a, we see this plot for the case when the measurement error is 30% of the true function's range (as before). We observe that the OC quickly decays as the number of measurements increases, showing that the NBKG value



(a) True function values (b) Prior estimation (c) Posterior estimation

Fig. 3: Prior and posterior estimates of the true function surface after 0 and 15 batch measurements, using the NBKG policy.

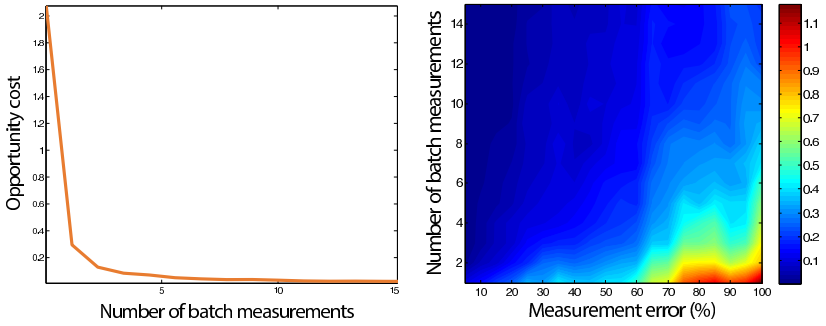


(a) 0 batch measurements (b) 5 batch measurements

(c) 10 batch measurements (d) 15 batch measurements

Fig. 4: Prior and posterior estimates of the true function surface after 0, 5, 10 and 15 batch measurements, using the NBKG policy. For each choice of number of measurements, the plot shows the residual error between this estimate and the true function.

rapidly finds the location of the maximal photocurrent. Figure 5b shows the mean OC versus the number of batch measurements and measurement error. We observe that the OC increases with increasing error, as expected. Such a plot is meaningful in experimental budgeting, and shows the requisite number of measurements needed to obtain a certain level of optimality for a particular level of noise. This plot can suggest to the experimenter the amount of measurement precision needed in order to achieve a desired level of optimality as measured by opportunity cost.



(a) OC under 30% measurement error

(b) OC vs. number of batch measurement and measurement error

Fig. 5: Opportunity cost

We then experiment with different Monte Carlo sampling sizes. At each time step (k, b), for different sampling sizes Q, we calculate the standard error $\frac{s_{mc}}{\sqrt{Q}}$ of all the densities, with s_{mc} denoting the sample standard deviation, and verify that the maximum of such values is below an acceptable tolerance. For example, the left two figures in Figure 6 depict the empirical means of NBKG values of a fixed NP size = 800nm and all densities under different sampling sizes. In the leftmost figure, the time step (k,b)=(0,2) is considered. When $Q = 1000$, the maximum standard error of all the densities is 0.0476 and the maximum difference between this case and $Q = 50000$ over all densities is 0.0158. When $Q = 10000$, the maximum standard error is 0.0129 and the maximum difference with $Q = 50000$ is 0.0051. $Q = 10000$ achieves the maximum NBKG value at the same point as $Q = 50000$ in this case. Similar performance is observed for $(k, b) = (0, 5)$, as plotted in the middle figure, and later time steps. We also plot the opportunity cost curves in log-log scale for different sampling sizes. We observe that the opportunity cost curve of $Q = 10000$ is similar to that of $Q = 50000$ with a mean difference of 0.0073, showing that $Q = 10000$ is sufficiently large to ensure accuracy as measured by opportunity cost.

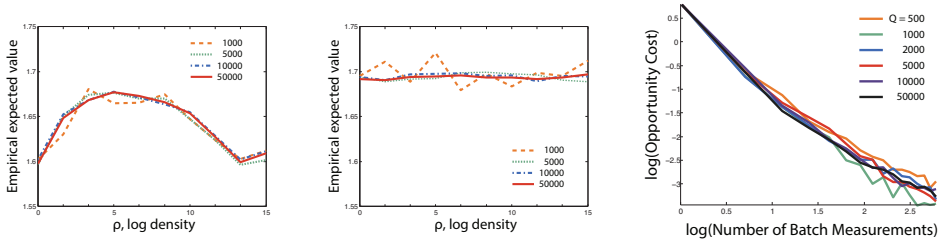


Fig. 6: Left two figures plot the empirical means of NBKG values of a fixed NP size and all densities under different Monte Carlo sampling sizes. Horizontal axis denotes 10 densities and vertical axis is the empirical mean. The left figure is calculated at time step $(k, b) = (0, 1)$ and the middle figure is calculated at time step $(k, b) = (0, 4)$. The right figure depicts the opportunity cost curves for different sampling sizes under different number of batch measurements.

We may also assess the performance of NGKB as the problem size increases. We experiment with different batch sizes $B = 1, 2, 3, 4, 5$ and report in Figure 7 the mean opportunity cost after each batch measurement ranging from 0 to 15, averaged over 500 runs. In order to make a fair comparison, all the observations are pre-generated and shared for simulations with different batch sizes. We observe that no matter which batch size it uses, the OC quickly decays as the number of batch measurements increases. Since a larger batch size means more measurements at each iteration, thus providing more information and yielding more precise estimation. This intuition is also verified in Proposition 6.2 (Benefits of measurement). We see from the figure that for any measurement budget K, larger batch sizes yield lower OC, as expected.

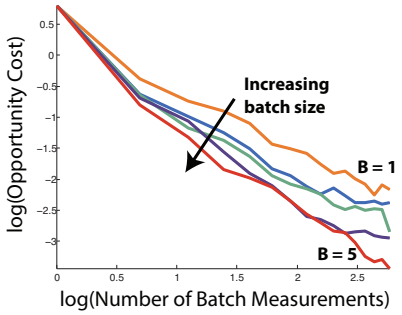


Fig. 7: Performance of NGKB as K, B changes. Horizontal axis denotes the logarithm of the number of batch measurement $K = 0, 1, ..., 15$. Vertical axis is the logarithm of mean opportunity cost. Lines with different colors correspond to different simulations with different batch sizes $B = 1, 2, ..., 5$.

8.3. Comparison with Other Policies. In this section, we consider the performance of NBKG in comparison to other policies. We consider the following policies:

1. **Nested-Batch KG**: The policy described in the paper.
2. **Sequential KG**: The basic, sequential KG policy as described in [11].
3. **Sequential Exploration**: The pure exploration policy, which chooses an alternative uniformly at random.
4. **Nested-Batch Exploration**: A random NP size is selected, and then B NP densities are selected in batch.
5. **Sequential Exploitation**: The pure exploitation policy, which chooses the alternative x^n corresponding to the maximum value, $\max_x \theta_x^n$.
6. **Nested-Batch Exploitation**: Select the batch of experiments

$$\{(d, \rho_1), \ldots, (d, \rho_B)\},$$

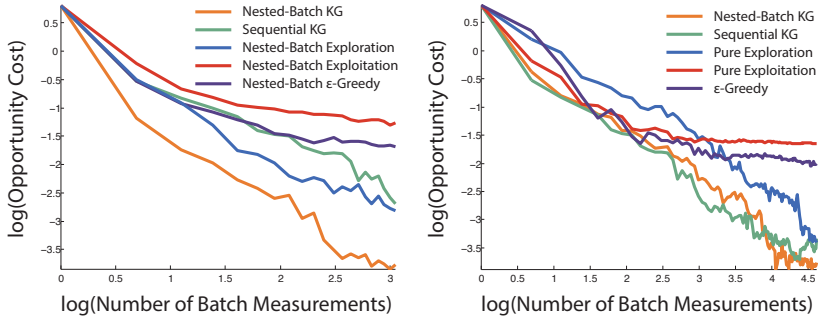
that maximizes

$$\mathcal{I}(d, \rho_1, \ldots, \rho_B) = \sum_{i=1}^{B} \theta^n_{(d,\rho_i)}.$$

7. **Sequential ϵ-Greedy**: A sequential policy that provides a mixture between the pure exploration and exploitation policy. The alternative x^n selected at time n is obtain by choosing between pure exploration with probability ϵ^n and pure exploitation with probability $(1 - \epsilon^n)$, where $\epsilon^n = 0.9/n$.

8. **Nested-Batch ϵ-Greedy**: Similar to the sequential ϵ-greedy policy, but chooses between the nested-batch versions of exploration and exploitation with probability ϵ^n and $(1-\epsilon^n)$, respectively.

Figure 8a plots the mean opportunity cost for the nested-batch policies as a function of the number of batch measurements, averaged over 200 independent simulations and plotted in log scale for clarity. We observe that NBKG outperforms all the nested-batch policies. Also included in the figure is the opportunity cost for the sequential KG policy. In the nested-batch setting, the sequential KG does not take advantage of batch experiments, opting instead of performing the single experiment with largest KG value, effectively using a batch size of $B = 1$. We note that NBKG outperforms the sequential KG policy, as illustrated in Figure 7. The comparison between NBKG and sequential KG exhibits the experimental savings to be gained in performing experiments in batch mode. Figure 8b compares NBKG versus the sequential policies in a sequential experiment setting. In this context, we equate one batch measurement performed using the NBKG policy with B sequentially measurements for comparison. The sequential policies are more adaptive than NBKG in this manner, as they can incorporate information obtained from experiments one at a time, while NBKG only updates the state of knowledge after B measurements. Nevertheless, we observe that for a large number of measurements, NBKG outperforms all sequential policies except for sequential KG. Between sequential and NBKG, we observe similar performance, hinting that while NBKG has a delay in updating information, the effect of this delay is minimal.



(a) Nested-batch experiments (b) Sequential experiments

Fig. 8: A comparison of policy performance. The graphs show mean opportunity cost versus the number of measurement for the policies outlined above. (a) Nested-batch experiments, in which a policy may perform several experiments in parallel, varying NP density, provided that the NP size is the same between the parallel experiments. Sequential policies use a batch size of $B = 1$. (b) Sequential experiments, in which experiments must be performed one at a time. Here we equate 1 batch measurement with B sequential measurements.

9. Conclusion. In this paper, motivated by several applications, we extended the sequential ranking and selection problem into a general framework for batch-mode learning and nested-batch-mode learning. By formulating the problem within a dynamic programming framework, we derived the Knowledge-Gradient variants to tackle both batch and nested-batch measurements. Since the Knowledge-Gradient

variants require computing expectations which may be intractable, a Monte Carlo sampling procedure was applied. We empirically demonstrate the effectiveness of the NBKG policy on the immobilized nanoparticles design problem. We see that NGKB is competitive with a fully sequential strategy and significantly outperforming pure exploration, pure exploitation and ϵ-greedy strategies for the model application presented in this paper.

REFERENCES

[1] R. Agrawal, M. Hegde, and D. Teneketzis, *Multi-armed bandit problems with multiple plays and switching cost*, Stochastics and Stochastic Reports, 29 (1990), pp. 437–459.

[2] J. Audibert, S. Bubeck, and G. Lugosi, *Regret in online combinatorial optimization*, Math. Oper. Res., (2013).

[3] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer, *Finite-time analysis of the multi-armed bandit problem*, Mach. Learn., 47 (2002), pp. 235–256.

[4] E. Barut and W. B. Powell, *Optimal learning for sequential sampling with non-parametric beliefs*, J. Global Optim., (2013), pp. 1–27.

[5] Sébastien Bubeck and Nicolò Cesa-Bianchi, *Regret analysis of stochastic and nonstochastic multi-armed bandit problems*, arXiv preprint arXiv:1204.5721, (2012).

[6] N. Cesa-Bianchi and G. Lugosi, *Combinatorial bandits*, J. Comput. System Sci., 78 (2012), pp. 1404–1422.

[7] Y. Chen and A. Krause, *Near-optimal batch mode active learning and adaptive submodular optimization*, in Proceedings of The 30th International Conference on Machine Learning, 2013, pp. 160–168.

[8] M. H. DeGroot, *Optimal Statistical Decisions*, McGraw-Hill, 1970.

[9] AleksandraB. Djurii and Yu Hang Leung, *Optical properties of zno nanostructures*, Small, 2 (2006), pp. 944–961.

[10] Franois Flory, Ludovic Escoubas, and Grard Berginc, *Optical properties of nanostructured materials: areview*, Journal of Nanophotonics, 5 (2011), pp. 052502–052502–20.

[11] P. I. Frazier, W. Powell, and S. Dayanik, *The knowledge-gradient policy for correlated normal beliefs*, INFORMS J. Comput., 21 (2009), pp. 599–613.

[12] P. I. Frazier, W. B. Powell, and S. Dayanik, *A knowledge-gradient policy for sequential information collection*, SIAM J. Control Optim., 47 (2008), pp. 2410–2439.

[13] Russell J. Gehr and Robert W. Boyd, *Optical properties of nanostructured optical materials*, Chemistry of Materials, 8 (1996), pp. 1807–1819.

[14] L. R. Giam, S. He, N. E. Horwitz, D. J. Eichelsdoerfer, J. Chai, Z. Zheng, D. Kim, W. Shim, and C. A. Mirkin, *Positionally defined, binary semiconductor nanoparticles synthesized by scanning probe block copolymer lithography*, Nano Lett., 12 (2012), pp. 1022–1025.

[15] Aditya Gopalan, Shie Mannor, and Yishay Mansour, *Thompson sampling for complex online problems*, in Proceedings of The 31st International Conference on Machine Learning, 2014, pp. 100–108.

[16] Donghai He, Stephen E Chick, and Chun-Hung Chen, *Opportunity cost and OCBA selection procedures in ordinal optimization for a fixed number of alternative systems*, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 37 (2007), pp. 951–961.

[17] S. Kale, L. Reyzin, and R. E. Schapire, *Non-stochastic bandit slate problems.*, in NIPS, 2010, pp. 1054–1062.

[18] M. R. Langille, M. L. Personick, J. Zhang, and C. A. Mirkin, *Defining rules for the shape evolution of gold nanoparticles*, J. Am. Chem. Soc., 134 (2012), pp. 14542–14554.

[19] M. R. Mes, W. B. Powell, and P. I. Frazier, *Hierarchical knowledge gradient for sequential sampling*, J. Mach. Learn. Res., 12 (2011), pp. 2931–2974.

[20] J. E. Millstone, S. J. Hurst, G. S. Mtraux, J. I. Cutler, and C. A. Mirkin, *Colloidal gold and silver triangular nanoprisms*, Small, 5 (2009), pp. 646–664.

[21] D. C. Montgomery, *Design and Analysis of Experiments*, John Wiley and Sons, 2008.

[22] D. M. Negoescu, P. I. Frazier, and W. B. Powell, *The knowledge-gradient algorithm for sequencing experiments in drug discovery*, INFORMS J. Comput., 23 (2011), pp. 346–363.

[23] S. Y. Park, A. K. Lytton-Jean, B. Lee, S. Weigand, G. C. Schatz, and C. A. Mirkin, *DNA-programmable nanoparticle crystallization*, Nat., 451 (2008), pp. 553–556.

[24] W. B. Powell and I. O. Ryzhov, *Optimal Learning*, John Wiley and Sons, 2012.

[25] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims, *Learning diverse rankings with multi-armed bandits*, in Proceedings of the 25th international conference on Machine learning, ACM, 2008, pp. 784–791.

[26] I. O. Ryzhov and W. Powell, *A Monte Carlo knowledge gradient method for learning abatement potential of emissions reduction technologies*, in Proceedings of the Winter Simulation Conference, IEEE, 2009, pp. 1492–1502.

[27] I. O. Ryzhov and W. B. Powell, *The knowledge gradient algorithm for online subset selection*, in IEEE SSCI ADPRL, Nashville, TN, 2009, pp. 137–144.

[28] A. J. Senesi, D. J. Eichelsdoerfer, R. J. Macfarlane, M. R. Jones, E. Auyeung, B. Lee, and C. A. Mirkin, *Stepwise evolution of DNA-programmable nanoparticle superlattices*, Angew. Chem. Int. Ed., 52 (2013), pp. 6624–6628.

[29] T. Uchiya, A. Nakamura, and M. Kudo, *Algorithms for adversarial bandit problems with multiple plays*, in Algorithmic Learning Theory, Springer, 2010, pp. 375–389.

[30] G. B. Wetherill and K. D. Glazebrook, *Sequential Methods in Statistics*, Chapman and Hall, 1986.