

## Optimal learning in Experimental Design Using the Knowledge Gradient Policy with Application to Characterizing Nanoemulsion Stability

Si Chen<sup>†¶</sup>, Kristofer-Roy G. Reyes<sup>‡¶</sup>, Maneesh K. Gupta<sup>§||</sup>, Michael C. McAlpine<sup>§</sup>, and Warren B. Powell<sup>‡¶</sup>

**Abstract.** We present a technique for adaptively choosing a sequence of experiments for materials design and optimization. Specifically, we consider problem of identifying the choice of experimental control variables that optimize the kinetic stability of a nanoemulsion, which we formulate as a ranking and selection problem. We introduce an optimization algorithm called the Knowledge Gradient with Discrete Priors (KGDP) that sequentially and adaptively selects experiments that maximizes the rate of learning the optimal control variables. This is done through a combination of a physical, kinetic model of nanoemulsion stability, Bayesian inference and a decision policy. Prior knowledge from domain experts are incorporated into the algorithm as well. Through numerical experiments, we show that the KGDP algorithm outperforms both the policies of random exploration (in which an experiment is selected at uniformly random among all potential experiments) and exploitation (which selects the experiment that appears to be the best, given a current state of Bayesian knowledge).

**Key words.** Bayesian analysis, optimal learning, nanoemulsion, knowledge gradient, sequential decisions making, material science

**AMS subject classifications.** 62F07, 62F15, 62K05, 49M25, 62L05, 93E35

**1. Introduction.** Controlled release is the deliberate triggering and delivery of payload molecules into solution through an active mechanism [5]. Controlled release of payload has applications in chemical sensing [26], and the induction of spatial and temporal concentration gradients of the molecules in solution [25]. Payloads include reactive or catalytic species, bio molecules [9, 29], fluorescent markers and whole cells (e.g. bacteria, yeast). One technique for controlled payload delivery uses water-oil-water (W/O/W) double nanoemulsions [11, 13], which is comprised of the dispersion of two immiscible liquids (referred to throughout this paper as oil and water) wherein water droplets containing the payload molecules are dispersed inside oil droplets, which are subsequently dispersed inside an external aqueous phase. The dispersed droplets of both phases have diameters in the nanometer and micrometer length scales.

In this paper, we focus on the stability of an emulsion, whose release is triggered by the excitation of gold nanoparticles which have been functionalized onto the oil droplets' surface.

---

<sup>†</sup>Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 ([sichen@princeton.edu](mailto:sichen@princeton.edu))

<sup>‡</sup>Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544 ([kreyes@princeton.com](mailto:kreyes@princeton.com), [powell@princeton.edu](mailto:powell@princeton.edu))

<sup>§</sup>Department of Mechanical and Aerospace Engineering, Princeton, NJ 08544 ([maneeshg@princeton.edu](mailto:maneeshg@princeton.edu), [mcm@princeton.edu](mailto:mcm@princeton.edu)). This author's research was partially supported by the Air Force Office of Scientific Research (No. FA9550-12-1-0367).

<sup>¶</sup>This author was supported in part by the Air Force Office of Scientific Research (No. FA9550-12-1-0200) for Natural Materials, Systems and Extremophiles.

<sup>||</sup>This author was supported in part by an Intelligence Community Postdoctoral Fellowship (No. 2013-13070300004).

As we shall see, the experimental design of such a study is extremely difficult. To address this issue, we introduce a procedure for sequential design of experiments that maximizes the rate of learning, using an optimal learning technique known as the Knowledge Gradient (KG) [8], which quantifies the informational value of an experiment. We build on a model that captures the dynamics of the destabilization process, parameterized by a family of unknown kinetic coefficients. Our method can be applied to any linear or non-linear parametric model, and can deal with models with no closed analytic form, which is the case in the emulsion problem.

The kinetics of payload delivery involve several coupled processes that depend on control variables such as droplet sizes, water/oil volume fractions and droplet diameters, as well as uncontrollable parameters such as kinetic coefficients. In determining which set of controllable, tunable parameters optimize some aspect of the emulsion (e.g. the stability of the emulsion), a scientist must often deal with ambiguity in the experiment on several fronts. First is inexact knowledge of the uncontrollable parameters. For example, when using a new material as the oil phase, parameters such as payload diffusivities through this new material may not be well understood. Second is the large number of potential experiments to run, which increases exponentially with respect to the number of control variables to be considered. Lastly, experiments are expensive and noisy. For example, emulsion stability is often characterized over a time scale of hours, days and sometimes even longer. Measurements of the amount of payload delivered to the external solution are made through secondary processing of solution samples, and reported values are not exactly the same between samples, leading to measurement noise. Together, the problems stated above express a need for a systematic technique in deciding an effective sequence of experiments that will lead us to the optimal set of control parameters for nanoemulsion stability. We address these problems using research drawn from the field of optimal learning, which offers a framework for guiding the process of collecting information when collecting information is time consuming and expensive.

In this study, we model the selection of optimal control parameters as a sequential ranking and selection problem (see [8] and the references cited there). Each experiment is a choice of several control variables, collectively denoted as  $\mathbf{x}$  and called an alternative. For example,  $\mathbf{x}$  might consist of the initial volume fraction of water droplets in the oil phase, the diameter of the oil droplets, and the initial volume fraction of water droplets in the oil phase. The goal in our experiments is to find the control variables  $\mathbf{x}$  that maximizes some measure of emulsion stability. Let  $\mu_{\mathbf{x}}$  denote this measure of stability using the configuration  $\mathbf{x}$ . This quantity is *a priori* unknown to us, and hence the challenge is to find the variables  $\mathbf{x}$  that maximize  $\mu_{\mathbf{x}}$ , using experiments that are both time-consuming and noisy.

In the literature on ranking and selection problems, there are two major approaches: the frequentist approach, which is based entirely on observed data [14, 16], and the Bayesian one, which assumes we have a prior distribution about the behavior of the experiment as we vary the control variables. The Knowledge Gradient is an example of a Bayesian approach, and was introduced in [8] for the case where alternatives are considered independent (i.e.  $\mu_{\mathbf{x}}$  does not correlate with  $\mu_{\mathbf{x}'}$  when  $\mathbf{x} \neq \mathbf{x}'$ ) and then generalized to problems with correlated alternatives [7]. Subsequent work on the KG technique focuses on the case when the measurement can be parameterized  $\mu_{\mathbf{x}} = f(\mathbf{x}; \boldsymbol{\kappa}^*)$ , wherein the uncertainty on the  $\mu_{\mathbf{x}}$  is transferred onto the *a priori* unknown parameter values  $\boldsymbol{\kappa}^*$ . For example, [20] describes the KG algorithm when the parameterization  $f(\mathbf{x}; \boldsymbol{\kappa})$  is linear in the indeterminate  $\boldsymbol{\kappa}$ .

In this paper, we present an extension of KG in the setting where  $f$  is non-linear in  $\kappa$  called the Knowledge Gradient with Discrete Priors (KGDP). With KGDP, we assume that the true function  $f(\mathbf{x}; \kappa^*)$  may be well approximated as a convex combination of the form

$$\bar{f}(\mathbf{x}) = \sum_{i=1}^L f(\mathbf{x}; \kappa_i) p_i,$$

where the  $p_i$  is a discrete probability distribution and the  $\kappa_i$  are sampled according to some prior distribution. This assumption has two major advantages. First, the convex combinations lead to simple Bayesian update and easy KG calculations. Second is that this technique does not restrict the function  $f$  to be linear in  $\kappa$  nor necessarily have an analytic closed-form. Such is the case when modeling kinetic processes, in which case  $f$  is a solution to a system of ordinary differential equations (ODEs), which often must be solved numerically. Here we present a numerical study of the performance of KGDP in the context of the nanoemulsion optimization problem, and show that it can significantly decrease the number of sequential experiments necessary in order to achieve a level of optimality when compared to other policies.

The paper is organized as follows. In Section 2, we describe the ranking and selection problem, discuss various KG policies and introduce our novel KGDP. In Section 3 we present the kinetic system to be studied throughout the paper and formulate it as a ranking and selection problem. In Section 4, we explain how to apply KGDP to the nanoemulsion problem and present empirical results. We conclude in Section 5.

**2. Knowledge Gradient with Discrete Priors.** We formulate the problem class in optimal learning as a ranking and selection selection problem. In the following section, we first discuss the formulation of the ranking and selection problem, followed by the review of one of the Bayesian approaches (Knowledge Gradient) to this problem and our extension, Knowledge Gradient with Discrete Prior (KGDP).

**2.1. Ranking and selection problem.** Ranking and selection problems in general consider a discrete set of  $M$  alternatives, which we denote as  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_M\}$ . We let the set  $\mathcal{I} = 1, \dots, M$  to be the index set of the alternatives and for any index  $i \in \mathcal{I}$ ,  $\mathbf{x}_i \in \mathcal{X}$ . Each alternative  $\mathbf{x}_i \in \mathcal{X}$  with  $i \in \mathcal{I}$  is assigned a true utility value  $\mu_i$ , which measures the performance of  $\mathbf{x}_i$ . This true utility value is presumed unknown to us and we can only estimate it using  $\theta_i$ . Our goal is to determine the experiment  $\mathbf{x} \in \mathcal{X}$  with largest assigned value  $\mu_i$  by querying these values through  $N$  sequential, noisy measurements. We wish to design an adaptive decision rule that suggests which alternative to query next, given our current knowledge about the values  $\mu_i$ , so that we are well equipped to make the final decision on the optimal alternative after we have exhausted the measurement budget  $N$ .

Under the Bayesian setting, we assume we have a prior belief on the unknown true utility value  $\boldsymbol{\mu}$ . We write  $\boldsymbol{\mu}$  to indicate the column vector  $(\mu_1, \dots, \mu_M)'$ . We define  $\Omega_1$  and  $\Omega_2$  to be the sample spaces on which the true utility  $\boldsymbol{\mu}$  and the measurement noise  $W$  are defined respectively. We then consider the sample space  $\Omega := \Omega_1 \times \Omega_2$ . The filtration  $\mathcal{F}^n$  is defined to be the  $\sigma$ -algebra generated by  $\mathbf{x}^0, \hat{y}^1, \dots, \mathbf{x}^{n-1}, \hat{y}^n$ . We write  $\mathbf{x}^n \in \mathcal{F}^n$  to imply the fact that we allow the experimentalist to make decisions sequentially, i.e. the decision  $\mathbf{x}^n$  depends only on measurements observed by time  $n$ . Note that we have chosen our indexing so that random

variables measurable with respect to the filtration at time  $n$  are indexed by the superscript  $n$ . We write  $\mathbb{E}^n$  to indicate  $\mathbb{E}[\cdot | \mathcal{F}^n]$ .

For the  $n$ -th experiment, we choose  $\mathbf{x}^n \in \mathcal{X}$  according to some decision making rule. We assume that the sample measurements  $\hat{y}_{\mathbf{x}^n}^{n+1}$  for alternative  $\mathbf{x}^n = \mathbf{x}_i$  are normally distributed with unknown mean  $\mu_i$  and known variance  $\sigma_i^2$ , and are of the form

$$\hat{y}_{\mathbf{x}^n}^{n+1} = \mu_i + W^{n+1},$$

where  $W^{n+1} \sim \mathcal{N}(0, \sigma_i^2)$  is the inherent noise of an experiment,  $\sigma_i^2$  is its variance, and  $\mu_i$  is the unknown true value of running the experiment using alternative  $\mathbf{x}^n = \mathbf{x}_i$  at time  $n$ . The decision  $\mathbf{x}$  is indexed by  $n$  and the measurement  $\hat{y}$  is indexed by  $n+1$  in order to emphasize the fact that  $\hat{y}$  is an unknown value when we make the decision at time  $n$ , i.e.  $\hat{y}^{n+1} \notin \mathcal{F}^n$ . The measurement will only be deterministic at time  $n+1$  after the time  $n$  experiment is performed, i.e.  $\hat{y}^{n+1} \in \mathcal{F}^{n+1}$ . Throughout this paper, we use bold letters to indicate vectors, superscripts to index time and subscripts to index the element of a vector or different elements in a set.

In offline learning, our goal is to select the alternative with the highest posterior mean after the budget of  $N$  measurement. In other words, we do not care about how well our choices perform during the process of collecting information. Instead, we are only concerned with how well our final choice performs. We define  $\Pi$  to be the set of all possible measurement policies that satisfies our sequential requirements; that is  $\Pi := \{(\mathbf{x}^0, \dots, \mathbf{x}^{N-1}) : \mathbf{x}^n \in \mathcal{F}^n\}$ . We use  $\mathbb{E}^\pi$  to indicate the expectation for a generic policy  $\pi \in \Pi$ . The process of choosing a measurement policy maximizing the expected reward can be written as

$$\sup_{\pi \in \Pi} \mathbb{E}^\pi \mu_{j^N},$$

where  $j^N = \arg \max_i \theta_i^N$  and  $\mathbf{x}^N = \mathbf{x}_{j^N}$  is the decision at time  $N$ .

There are two main approaches to the ranking and selection problems: the frequentist approach and the Bayesian approach. In this study, we focus on a Bayesian approach known as optimal learning with knowledge gradient, which selects alternatives that maximizes the expected value of information. Like other Bayesian approaches, the knowledge gradient uses subjective prior beliefs on the utility values of the parameter choices. This prior captures the expert knowledge of the scientists familiar with the problem. We briefly review the knowledge gradient approach here. For frequentist approaches and other Bayesian approaches, such as optimal computing budget allocation, one can refer to [3, 27, 16] for a thorough review. In this section, we describe two variations of the knowledge gradient. The first is the knowledge gradient for correlated beliefs using a lookup table belief model; the second uses a model that is linear in a low dimensional parameter vector  $\kappa$ . We then introduce an extension of KG, the Knowledge Gradient with Discrete Priors (KGDP), which handles belief models that are nonlinear in the parameter vector  $\kappa$ .

**2.2. Knowledge Gradient with Correlated Beliefs.** The knowledge gradient with correlated beliefs (KGCB) was first introduced in [7], which treats the function values as a random vector  $\mu = (\mu_i)_{i \in \mathcal{I}}$ . It assumes the true value  $\mu$  is distributed according to a multivariate normal prior distribution with mean  $\theta^0$  and covariance matrix  $\Sigma^0$ , i.e.  $\mu \sim \mathcal{N}(\theta^0, \Sigma^0)$ . An element of the matrix  $\Sigma^0$  is  $\text{Cov}(\mu_i, \mu_{i'})$ , which captures the relationship between alternatives

$\mathbf{x}_i$  and  $\mathbf{x}_{i'}$ . If  $\text{Cov}(\mu_i, \mu_{i'})$  is large and our belief about  $\mu_i$  is higher than expected (for example), then we will raise our belief about  $\mu_{i'}$ . Such a non-trivial covariance structure can arise in real world application such as correlations in beliefs about experimental results that use similar tunable parameter values. For example,  $\mathbf{x}_i$  and  $\mathbf{x}_{i'}$  might be two experiments with relatively similar tunable parameters, or two catalysts with similar properties, and we may expect their experimental outcomes are similar to each other.

We define the state variable  $S^n := (\boldsymbol{\kappa}^n, \Sigma^n)$  to be the state of knowledge at time  $n$ , and the value of information (or the reward) of state  $S^n$  to be  $V^n(S^n) = \max_{i'} \theta_{i'}^n$ . At time  $n$ , the prior mean  $\boldsymbol{\kappa}^n$  is our best estimate of true  $\boldsymbol{\mu}$  with the uncertainty captured by  $\Sigma^n$ . The knowledge gradient at  $\mathbf{x}$  represents the expected incremental value of information obtained from measuring a particular alternative  $\mathbf{x}$ . It is defined as

$$\begin{aligned}\nu^{KG,n}(\mathbf{x}) &= \mathbb{E}^n[V^{n+1}(S^{n+1}(\mathbf{x})) - V^n(S^n)|S^n] \\ &= \mathbb{E}^n[\max_{i'} \theta_{i'}^{n+1}|S^n = s, \mathbf{x}^n = \mathbf{x}] - \max_{i'} \theta_{i'}^n,\end{aligned}$$

where  $s$  is the sample value of  $S^n$ . In this KGCB definition,  $\boldsymbol{\theta}^{n+1}$  is the Bayesian posterior estimate of the function values given the observation  $\hat{y}_{\mathbf{x}^n}^{n+1}$  of alternative  $\mathbf{x}^n = \mathbf{x}$  at time  $n+1$ . This estimate is a random variable at time  $n$  as it depends on the actual experiment outcome, which is random at time  $n$  (the same reason why we index the measurement by  $n+1$ ), hence we need to take the expectation over all the possible experimental outcomes. At time  $n$ , KGCB makes the sampling decision by maximizing the knowledge gradient, which is given by

$$\mathbf{x}^n \in \arg \max_{\mathbf{x} \in \mathcal{X}} \nu^{KG,n}(\mathbf{x}).$$

After every experiment, we update our distribution based on the sampled value of the alternative that we decide to measure. Since the multivariate normal distribution is a natural conjugate family when the sample observations are normally distributed, the Bayesian posterior is also multivariate normal. The updating equation is given in [12] as

$$(2.1) \quad \boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n + \frac{\hat{y}_{\mathbf{x}^n}^{n+1} - \boldsymbol{\theta}_i^n}{\sigma_i^2 + \Sigma_{ii}^n} \Sigma^n e_i,$$

$$(2.2) \quad \Sigma^{n+1} = \Sigma^n - \frac{\Sigma^n e_i e_i' \Sigma^n}{\sigma_i^2 + \Sigma_{ii}^n},$$

where  $\hat{y}_{\mathbf{x}^n}^{n+1}$  is the outcome of the experiment run using  $\mathbf{x}^n = \mathbf{x}_i$ ,  $(\boldsymbol{\theta}^n, \Sigma^n)$  is the corresponding prior distribution at time  $n$ , and  $e_i$  is the  $M$ -column vector with 1 at the  $i$ -th index and the rest 0s.

The knowledge gradient policy is optimal by construction if the budget is  $N=1$ , and is asymptotically optimal [7], and is the only stationary policy with these properties (and no tunable parameters). The computation of the knowledge gradient with correlated beliefs grows with the square of  $M$ , due to storage and manipulation of the covariance matrix. The computation becomes problematic when the number of potential experimental combinations exceeds 1,000. To address this issue, [20] seeks a low dimensional parameterization of the function, and considers all uncertainty on the function values as arising from uncertainty in the parameter values.

**2.3. Knowledge Gradient for a Linear Belief Model.** The knowledge gradient for a linear belief model (KGLin) assumes the true function value  $\mu$  can be represented linearly in the unknown parameters. For example,  $\mu_i = \kappa_1 x_{i,1} + \kappa_2 x_{i,2} + \dots + \kappa_m x_{i,m}$ , where  $m$  is the dimension of the alternatives and  $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,m})^T \in \mathcal{X}$  is an alternative. Let  $\boldsymbol{\kappa}$  be the column vector  $(\kappa_1, \dots, \kappa_m)$  and  $X = (\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_M)^T$  be the alternative matrix. Instead of assuming the distribution of  $\mu$ , KGLin assumes the unknown parameters  $\boldsymbol{\kappa}$  is multivariate normal distributed with mean  $\boldsymbol{\kappa}^0$  and variance  $\Sigma^{\boldsymbol{\kappa},0}$ , i.e.  $\boldsymbol{\kappa} \sim \mathcal{N}(\boldsymbol{\kappa}^0, \Sigma^{\boldsymbol{\kappa},0})$ . Then the belief induced on the function value is

$$\mu \sim \mathcal{N}(X\boldsymbol{\kappa}^0, X\Sigma^{\boldsymbol{\kappa},0}X^T).$$

At time  $n$ , the true utility  $\mu$  is best estimated by the prior mean  $X\boldsymbol{\kappa}^n$ . The state variable is defined as  $S^n = (\boldsymbol{\kappa}^n, \Sigma^{\boldsymbol{\kappa},n})$ . The knowledge gradient for alternative  $\mathbf{x}$  at state  $S^n = s$  is now defined as

$$\begin{aligned}\nu^{KG,n}(\mathbf{x}) &= \mathbb{E}^n[V^{n+1}(S^{n+1}(\mathbf{x})) - V^n(S^n)|S^n] \\ &= \mathbb{E}^n[\max_{i'} \theta_{i'}^{n+1} | S^n = s, \mathbf{x}^n = \mathbf{x}] - \max_{i'} \theta_{i'}^n \\ &= \mathbb{E}^n[\max_{i'} (X\boldsymbol{\kappa}^{n+1})_{i'} | S^n = s, \mathbf{x}^n = \mathbf{x}] - \max_{i'} (X\boldsymbol{\kappa}^n)_{i'}.\end{aligned}$$

The following updating equations may be derived from Equation (2.1) and (2.2) through standard expressions for normal sampling of linear combinations of alternatives (see, e.g., [24]),

$$\begin{aligned}\boldsymbol{\kappa}^{n+1} &= \boldsymbol{\kappa}^n + \frac{\hat{y}_{\mathbf{x}^n}^{n+1} - (\boldsymbol{\kappa}^n)^T \mathbf{x}_i}{\sigma_i^2 + \mathbf{x}_i^T \Sigma^{\boldsymbol{\kappa},n} \mathbf{x}_i}, \\ \Sigma^{\boldsymbol{\kappa},n+1} &= \Sigma^{\boldsymbol{\kappa},n} - \frac{1}{\sigma_i^2 + \mathbf{x}_i^T \Sigma^{\boldsymbol{\kappa},n} \mathbf{x}_i} (\Sigma^{\boldsymbol{\kappa},n} \mathbf{x}_i \mathbf{x}_i^T \Sigma^{\boldsymbol{\kappa},n}),\end{aligned}$$

where  $\hat{y}_{\mathbf{x}^n}^{n+1}$  is the measurement of alternative  $\mathbf{x}^n = \mathbf{x}_i$ .

KGLin solves the computational problem associated with large numbers of alternatives by using a linear parametric model. Instead of storing and manipulating a covariance matrix of all the alternatives with size  $M^2$ , KGLin only maintains the covariance matrix of the linear parameters with size  $m^2$  where  $m \ll M$ . KGLin has been shown to outperform other policies by simulation [20] and the optimality proofs for KGCB can also be extended to KGLin. However, a linear model is not always accurate in practice, especially in problems that involve kinetic models, such as the nanoemulsion problem introduced in the paper, which models are often solution to a system of ODEs that are highly non-linear and hence cannot be solved by KGLin.

**2.4. Knowledge Gradient with Discrete Priors.** The Knowledge Gradient with Discrete Priors (KGDP) assumes that we have  $L$  candidate truths (or candidates), denoted as  $f_1(\mathbf{x}), \dots, f_L(\mathbf{x})$  for  $\mathbf{x} \in \mathcal{X}$ . As in the linear model, we assume that the truth  $\mu_j$  can be parameterized with parameter  $\boldsymbol{\kappa}^*$  as  $\mu_j = f(\mathbf{x}_j; \boldsymbol{\kappa}^*)$ , and that the candidate truths differ by the choice of parameter values  $f_i(\mathbf{x}) = f(\mathbf{x}; \boldsymbol{\kappa}_i)$ . From now on, we use  $f(\mathbf{x}_j; \boldsymbol{\kappa}^*)$  instead of  $\mu_j$  to

denote the truth to emphasize parameterization. Our proximity assumption of KGDP is that the truth is equal to or near one of these  $L$  candidate truths. As we shall see momentarily, this assumption leads to easy computation of the Bayesian update of the prior and KG calculations. Under this assumption, the truth is denoted as  $f(\mathbf{x}; \boldsymbol{\kappa}^*) = f_i(\mathbf{x})$  for some  $1 \leq i \leq L$ . The probability of  $f_i$  being the truth at time  $n$  is defined as  $p_i^n = \mathbb{P}(f(\mathbf{x}, \boldsymbol{\kappa}^*) = f_i(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X} | \mathcal{F}^n)$  and abbreviated as  $p_i^n = \mathbb{P}(f^* = f_i | \mathcal{F}^n)$ . Denote  $\mathbf{p}^n = (p_1^n, p_2^n, \dots, p_L^n)$  as the weight vector where  $\mathbf{p}^n$  is  $\mathcal{F}^n$ -measurable. We define the  $\mathcal{F}^n$ -measurable utility function estimation of  $f(\mathbf{x}, \boldsymbol{\kappa}^*)$  as a weighted sum of all the candidate truths, i.e.

$$\bar{f}^n(\mathbf{x}) = \sum_{i=1}^L f_i(\mathbf{x}) p_i^n.$$

We assume that the sample measurements  $\hat{y}_{\mathbf{x}^n}^{n+1}$  of alternative  $\mathbf{x}^n$  are normally distributed with unknown mean  $f(\mathbf{x}^n; \boldsymbol{\kappa}^*)$  and known variance  $\sigma^2$ . The sample observation can be written as  $\hat{y}_{\mathbf{x}^n}^{n+1} | \mathbf{x}^n = f(\mathbf{x}^n; \boldsymbol{\kappa}^*) + W^{n+1}$  where  $W^{n+1}$  is a random variable and  $W^{n+1} \sim \mathcal{N}(0, \sigma^2)$ . Note that the noise  $W^1, \dots, W^{n+1}$  are independent and identically distributed and the current measurement is independent of the past history once the current decision is made. We assume uniform variance across all experiments (homoscedasticity) and believe this is a reasonable modeling approximation for most applications. By Bayes' rule, the posterior probability is proportional to the prior probability and the likelihood function,

$$\begin{aligned} p_i^{n+1} &= \mathbb{P}(f^* = f_i | \mathcal{F}^{n+1}) \\ &= \mathbb{P}(f^* = f_i | \hat{y}_{\mathbf{x}^n}^{n+1}, \mathbf{x}^n, \dots, \hat{y}_{\mathbf{x}^0}^1, \mathbf{x}^0) \\ &\propto g_Y(\hat{y}_{\mathbf{x}^n}^{n+1} | f^* = f_i, \mathbf{x}^n, \hat{y}_{\mathbf{x}^{n-1}}^n, \mathbf{x}^{n-1}, \dots, \hat{y}_{\mathbf{x}^0}^1, \mathbf{x}^0) \times \mathbb{P}(f^* = f_i | \mathbf{x}^n, \hat{y}_{\mathbf{x}^{n-1}}^n, \mathbf{x}^{n-1}, \dots, \hat{y}_{\mathbf{x}^0}^1, \mathbf{x}^0) \\ &= g_Y(\hat{y}_{\mathbf{x}^n}^{n+1} | f^* = f_i, \mathbf{x}^n) \times \mathbb{P}(f^* = f_i | \mathbf{x}^n, \mathcal{F}^n) \\ &= g_Y(\hat{y}_{\mathbf{x}^n}^{n+1} | f^* = f_i, \mathbf{x}^n) \times \mathbb{P}(f^* = f_i | \mathcal{F}^n) \\ &= g_Y(\hat{y}_{\mathbf{x}^n}^{n+1} | f^* = f_i, \mathbf{x}^n) p_i^n, \end{aligned}$$

where  $g_Y$  is the density function of  $\hat{y}^{n+1}$  and we assume that the observations  $\hat{y}^{n+1}, \dots, \hat{y}^1$  are conditionally independent given  $f^* = f_i$  and the decision  $x^n, \dots, x^0$  (i.e. that an measurement has an independent noise added to its corresponding truth value on each experiment, once a decision is made). When the noise is normally distributed, the likelihood is given by

$$g_Y(\hat{y}_{\mathbf{x}^n}^{n+1} | f^* = f_i, \mathbf{x}^n) = \exp \left[ -\frac{(\hat{y}_{\mathbf{x}^n}^{n+1} - f_i(\mathbf{x}^n))^2}{2\sigma^2} \right].$$

We update  $\mathbf{p}^n$  using

$$(2.3) \quad p_i^{n+1} = \frac{\exp \left[ -\frac{(\hat{y}_{\mathbf{x}^n}^{n+1} - f_i(\mathbf{x}^n))^2}{2\sigma^2} \right]}{\sum_{j=1}^L \exp \left[ -\frac{(\hat{y}_{\mathbf{x}^n}^{n+1} - f_j(\mathbf{x}^n))^2}{2\sigma^2} \right]} p_i^n.$$

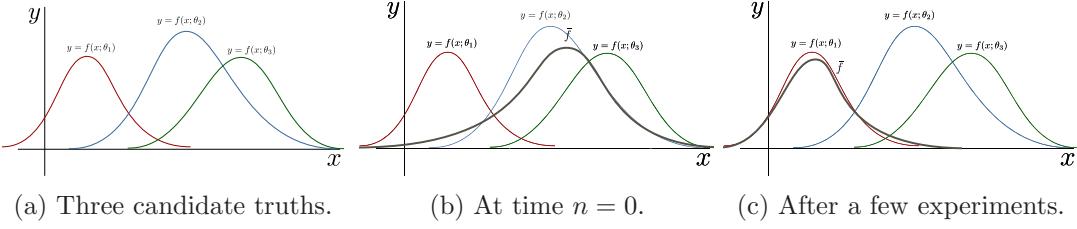


Figure 1: Illustration of KGDP updates.

To illustrate this method graphically, suppose we start with three different candidate truths,  $f_i(\mathbf{x})$  for  $i = 1, 2, 3$  as in Figure 1a. At time  $n = 0$ , we assume that these three candidates have equal probability of being the truth, i.e.  $p_i^0 = \frac{1}{3}$  for  $i = 1, 2, 3$ . Then the utility function at time  $n = 0$  is shown in Figure 1b. If  $f(\mathbf{x}, \kappa^*) = f_1(\mathbf{x})$  is the real truth, then after a few experiments and updates, the utility function will be much closer to  $f_1$ , as in Figure 1c.

We define the state variable to be  $S^n = \mathbf{p}^n$ . As before, the KGDP value of an alternative  $\mathbf{x}$  is defined to be the marginal value of information if we make a measurement on the alternative  $\mathbf{x}$ . At time  $n$ , the KGDP value for alternative  $\mathbf{x}$  at state  $S^n = s$  is now defined as:

$$\begin{aligned}
\nu^{\text{KGDP}, n}(x) &= \mathbb{E}^n[\max_{i'} \theta_{i'}^{n+1} | S^n = s, \mathbf{x}^n = x] - \max_{i'} \theta_{i'}^n \\
&= \mathbb{E}^n[\max_{x'} \bar{f}^{n+1}(x') | S^n = s, \mathbf{x}^n = x] - \max_{x'} \bar{f}^n(x') \\
&= \mathbb{E}^n[\max_{x'} \sum_{i=1}^L f_i(x') p_i^{n+1} | S^n = s, \mathbf{x}^n = x] - \max_{x'} \sum_{i=1}^L f_i(x') p_i^n \\
&= \sum_{j=1}^L \int_{w \in \Omega_2} \max_{x'} \frac{1}{c_j} \sum_{i=1}^L f_i(x') \exp \left[ -\frac{(f_j(x) - f_i(x) + w)^2}{2\sigma^2} \right] p_i^n g(w) dw p_j^n \\
(2.4) \quad &\quad - \max_{x'} \sum_{i=1}^L f_i(x') p_i^n,
\end{aligned}$$

with  $g(w)$  being the Gaussian density of  $W^{n+1}$  and  $c_j = \sum_{i=1}^L \exp \left[ -\frac{(f_j(x) - f_i(x) + w)^2}{2\sigma^2} \right] p_i^n$  being the normalizing constant.

The integral in Equation (2.4) can be approximated by numerical integration methods, such as Monte Carlo simulation and the midpoint rule. We use the midpoint rule since it produces smaller errors when used for evaluating a low-dimensional integral. To use the midpoint rule, we assume that the function is piecewise constant over some small interval. Within each interval, we approximate the function value by its value at the midpoint of the interval. The approximation of the integral is then given by the weighted sum of these values. This numerical integral is performed once for every candidate truth, and for every alternative, which means that it is calculated  $L \times M$  times at each time step. We note that this calculation

can be parallelized across alternatives. A satisfactory running time can be achieved by keeping  $L$  relatively small.

The KGDP algorithm offers several advantages. First, it can handle any nonlinear belief model; the derivation of KGDP does not make any assumptions of the general form of the model. In the nanoemulsion stability example, the prior comes from the solution of a system of nonlinear ODEs, which does not necessarily have a closed form. Second, optimal learning with KGDP achieves two goals simultaneously: optimization and learning. The KGDP policy is designed to quickly find the parameter settings which maximize the utility function. Through this process, we infer the discrete distribution on the different candidate parameter vectors, which allows us to approximate the true function. Through this approximation and knowledge of the discrete distribution, the inverse problem of determining the unknown parameters  $\kappa^*$  that yield the true function values is simplified, as we describe in Section 4.3. Hence, through KGDP, we also obtain an approximation of the underlying unknown parameters. Learning the unknown parameters gives a scientist a better understanding of the underlying destabilization kinetics, even in the face of coupled processes.

When the measurement noise  $\sigma^2$  is negligible and the set of possible experiments is small, both of these two problems may be solved with a relatively small budget. However, when the experimental noise and the number of possible experiments is large (as is usually the case), the problems are inherently more difficult. With all these difficulties, the KGDP still accelerates the rate at which we find the best set of control variables, sometimes cutting the experimental budget in half to achieve the same results when compared to pure exploration and exploitation. In fact, in the case of independent beliefs [8] and correlated beliefs with linear models in drug discovery [20], the KG policy performs well compared to other state-of-the-art policies. In the following sections, we describe the nanoemulsion stability problem in details and how to apply KGDP to solve the problem.

**3. The Model.** The goal of our nanoemulsion study is to construct a nano system that has the desired controlled release properties through a set of tunable control parameters. This goal and problem setting coincide with those of ranking and selection problems. In this section, we start with presenting the nanoemulsion optimization problem in detail in Section 3.1, which describes the kinetics of nanoemulsion stability. We then discuss how to model such a problem as a ranking and selection problem in Section 3.2.

**3.1. A Kinetic Model for Nanoemulsion Stability.** Throughout this paper, we address the problem of controlled payload delivery using double emulsions. While a full derivation of the kinetic model describing the stability of such an emulsion is beyond the scope of this paper, in this section we briefly present the model, and give expressions for the rate at which the payload is delivered. A water-oil-water (W/O/W) double emulsion consists of internal water droplets dispersed inside oil globules, which are subsequently dispersed inside an external water phase. Contained inside the internal water droplets are payload molecules, which are to be delivered to the external water phase. This delivery is facilitated by the functionalization of the oil globules by gold nanoparticles (NPs). Figure 2 illustrates the typical structure of such a W/O/W emulsion.

The delivery of the payload molecules from the internal water droplets to the external phase is performed via laser excitation, wherein a laser with appropriate wavelength induces

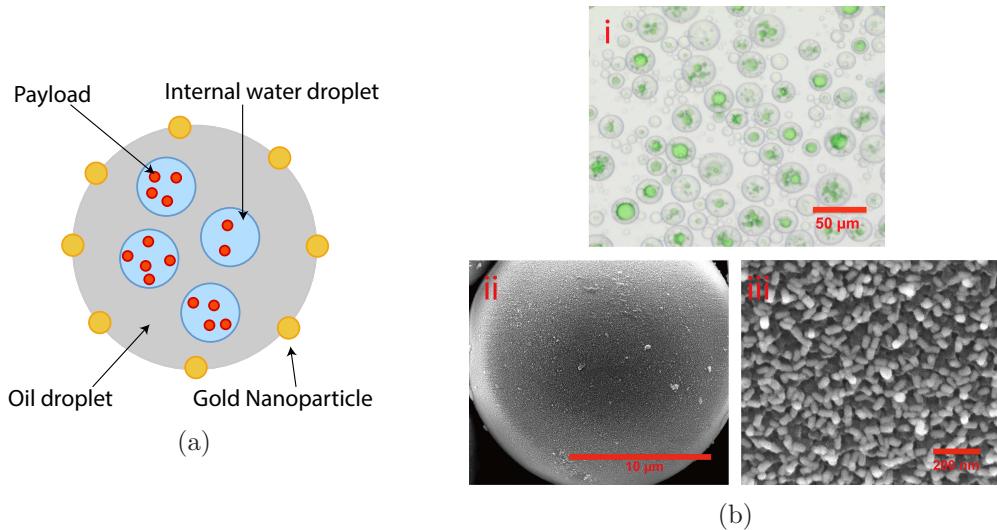


Figure 2: The structure of a W/O/W double emulsion. (a) A schematic highlighting the main constituents of a W/O/W emulsion. (b-i) An experimental image of a double emulsion. (b-ii) A single oil droplet whose surface is functionalized with gold NPs. (b-iii) A close-up image of the gold NPs.

surface plasmon resonance of the NPs and a subsequent temperature increase [18]. Throughout the literature, two main mechanisms of delivery are proposed. First is compositional ripening, in which the payload molecules diffuse and permeate the oil/water interfaces [22]. The second mechanism consists of the adsorption of an internal water droplet to the oil/water interface and its subsequent coalescence with the external phase, resulting in the delivery of the payload molecules contained inside [22, 10]. This mechanism is mediated by the activity of the internal water droplets, which is described by the secondary processes of droplet aggregation, droplet-droplet coalescence and droplet adsorption onto the oil/water interface. The two mechanisms are illustrated in Figure 3.

The resulting kinetics of the above processes is described by a coupled system of ordinary differential equations describing the amount  $N_i(t)$  (unit mols) of payload inside the internal water droplets at time  $t$ . The processes in compositional ripening are modeled as concentration-driven diffusion and thermally activated permeation. [22] provides a phenomenological model for this:

$$(3.1) \quad \partial_t^{\text{ripe}} N_i = k_{\text{ripe}} S_o \left( \frac{N_i}{V_i} - \frac{N_i(0) - N_i}{V_e} \right),$$

where  $\partial_t^{\text{ripe}}$  is the partial derivative with respect to time, corresponding to the change in payload via the ripening process only,  $k_{\text{ripe}}$  is the kinetic rate coefficient (units  $\mu\text{m s}^{-1}$ ) for ripening,  $S_o$  is the total surface area across all oil droplets. The volumes  $V_i, V_e$  are the total internal water droplet and external phase volume, respectively. The kinetic coefficient is given

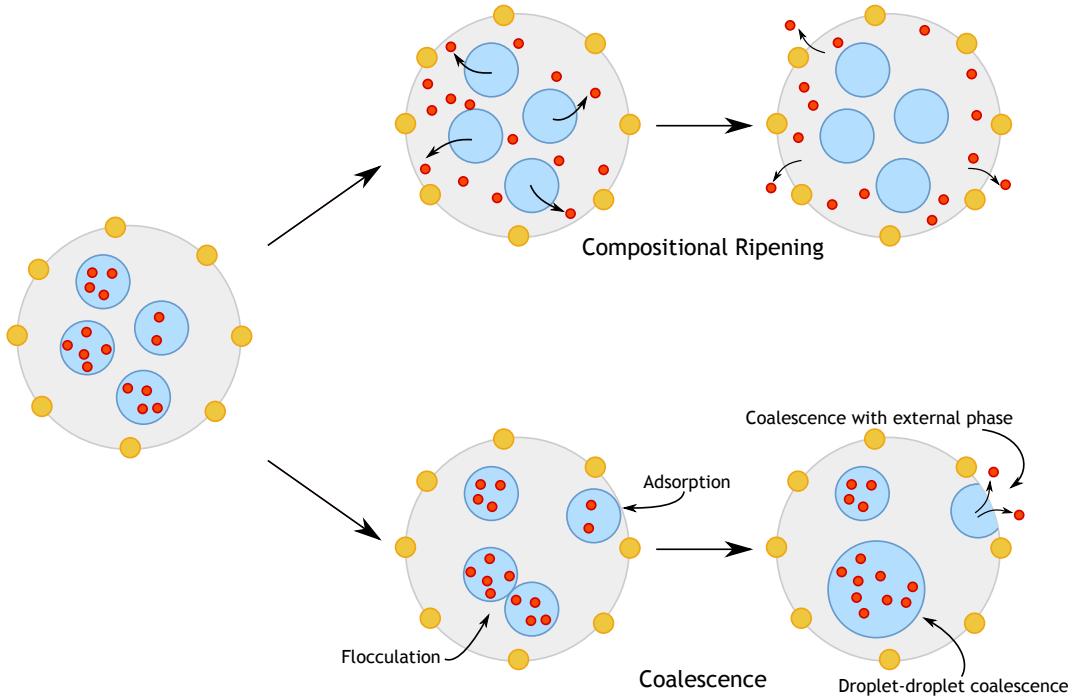


Figure 3: Mechanisms for payload delivery and associated processes.

by the Arrhenius Law

$$(3.2) \quad k_{\text{ripe}} = k_{\text{ripe}}^0 d_o \frac{\phi_w(1 - \phi_o)}{(1 - \phi_o) + \phi_w \phi_o} \exp \left[ -\frac{E_{\text{ripe}}}{k_B T} \right],$$

where  $d_o$  (units  $\mu\text{m}$ ) is the oil droplet diameter,  $k_{\text{ripe}}^0$  (units  $\text{time}^{-1}$ ) is a temperature independent rate prefactor and  $E_{\text{ripe}}$  (units eV) is the associated activation energy barrier for diffusion/permeation. The quantities  $\phi_w$  and  $\phi_o$  are time-dependent volume fractions whose initial values are controllable by an experimenter. The quantity  $\phi_w$  is the volume fraction between the internal water droplets and the oil droplets, while  $\phi_o$  is the volume fraction between oil droplets and external water phase.

The kinetics of the coalescence mechanism incorporate several models for activity of the internal water droplets, including the formation of aggregates of internal droplets, their adsorption onto the oil/water interface, and both droplet-droplet coalescence within aggregates and the coalescence of adsorbed droplets into the external water phase. In addition to  $V_i, V_e$  and  $N_i$  defined above, the quantities affected by these processes include  $\eta_k$ , the density (per unit oil droplet volume) of internal water droplet aggregates of size  $k$ ,  $\nu$ , the density (per unit oil droplet volume) of water droplets inside an oil droplet and  $d_w$ , the mean inner water droplet diameter. Droplet aggregation is modeled similar to Becker-Doering [23, 21] and von

Smoluchowski theories [28, 17], yielding the evolution equation for the  $\eta_k$

$$(3.3) \quad \partial_t \eta_k = \begin{cases} -k_{\text{coal}} \nu_a - k_{\text{floc}} \left( 2\eta_1^2 + \sum_{j=2}^{n_i^0-1} \eta_1 \eta_j \right) & k = 1, \\ -k_{\text{coal}} (k \eta_k - (k+1) \eta_{k+1}) + k_{\text{floc}} (\eta_1 \eta_{k-1} - \eta_1 \eta_k) & k = 2, \dots, n_i^0 - 1, \\ -k_{\text{coal}} n_i^0 \eta_{n_i^0} + k_{\text{floc}} \eta_1 \eta_{n_i^0-1} & k = n_i^0, \end{cases}$$

where  $n_i^0$  is the maximum (also initial) number of inner water droplets per oil droplet and  $\nu_a$  is the number density of droplets adsorbed onto the oil/water interface. The kinetic rate constants  $k_{\text{coal}}$  (units  $\text{s}^{-1}$ ) and  $k_{\text{floc}}$  (units  $\mu\text{m}^3 \text{s}^{-1}$ ) are given by Arrhenius law

$$(3.4) \quad k_{\text{coal}} = k_{\text{coal}}^0 \exp \left[ -\frac{E_{\text{coal}}}{k_B T} \right],$$

$$(3.5) \quad k_{\text{floc}} = k_{\text{floc}}^0 \exp \left[ -\frac{E_{\text{floc}}}{k_B T} \right],$$

where  $E_{\text{coal}}$  and  $E_{\text{floc}}$  are the activation energy barriers for the coalescence and flocculation process, respectively. The quantity  $\nu_a$  is given by the Langmuir adsorption model:

$$\frac{\nu_a}{\nu_s} = \frac{C_{\text{lang}} (V_o \eta_1 - S_o \nu_a)}{1 + C_{\text{lang}} (V_o \eta_1 - S_o \nu_a)},$$

where  $\nu_s$  is the density (per unit oil droplet surface area) of adsorption sites (obtained by geometric construction),  $V_o$  is the volume of the oil droplet,  $S_o$  is the surface area of the oil droplet and

$$C_{\text{lang}} = \exp \left[ \frac{\Delta E_{\text{sorp}}}{kT} \right],$$

with  $\Delta E_{\text{sorp}}$  being the difference in energy barriers of adsorption and desorption of a droplet onto and from the oil/water interface. Through the above evolution equation, the rates for all other time-dependent quantities can be obtained. In particular, the amount of payload delivered via the coalescence mechanism has rate

$$(3.6) \quad \partial_t^{\text{coal}} N_i = -k_{\text{coal}} \frac{\nu_a}{\nu} N_i,$$

where  $\partial_t^{\text{coal}}$  is the partial derivative with respect to time and the coalescence process only,  $\nu$  is the density (per unit oil droplet volume) of water droplets inside an oil droplet as defined earlier. By numerically solving the above system, we may determine the percentage of payload delivered  $N(\mathbf{x}, \boldsymbol{\kappa}; \tau, T)$  up to time  $\tau$  at a temperature  $T$  for tunable parameters

$$\mathbf{x} = (\phi_w^0, \phi_o^0, d_w^0, d_o, V_e^0)$$

and kinetic parameters

$$\boldsymbol{\kappa} = (k_{\text{ripe}}^0, E_{\text{ripe}}, k_{\text{floc}}^0, E_{\text{floc}}, k_{\text{coal}}^0, E_{\text{coal}}, \Delta E_{\text{sorp}}).$$

The tunable and kinetic parameters are summarized in Table 1. Note that  $\phi_w$ ,  $\phi_o$  and  $d_w$  are quantities that change over time and superscript zero above indicates the controllable variables

Table 1: A summary of the tunable and kinetic parameters in the model.

Tunable Parameter	Description
$\phi_w^0$	The initial volume fraction of water droplets in the oil phase
$\phi_o^0$	The initial volume fraction of the oil droplets in the external water phase
$d_w^0$	The initial diameter (units $\mu\text{m}$ ) of the water droplets
$d_o$	Diameter (units $\mu\text{m}$ ) of the oil droplets.
$V_e^0$	The initial volume (units $\mu\text{m}^3$ ) of the external phase
Unknown Parameter	Description
$k_{\text{ripe}}^0$	Rate prefactor (units $\text{s}^{-1}$ ) for compositional ripening
$E_{\text{ripe}}$	Energy barrier (units eV) for compositional ripening
$k_{\text{floc}}^0$	Rate prefactor (units $\mu\text{m}^3 \text{s}^{-1}$ ) for flocculation
$E_{\text{floc}}$	Energy barrier (units eV) for flocculation
$k_{\text{coal}}^0$	Rate prefactor (units $\text{s}^{-1}$ ) for coalescence
$E_{\text{coal}}$	Energy barrier (units eV) for coalescence
$\Delta E_{\text{sorp}}$	Droplet adsorption/desorption energy barrier difference (units eV)

are initial values of these quantities. Even though  $d_w^0$  does not appear in any of the above equations, it determines the geometric quantities  $\nu_s$ ,  $\nu_a$ , and  $\nu$ . In the controlled payload delivery context, we desire that the emulsion is stable under normal, unexcited conditions (characterized by a base temperature  $T_0$ ) over a long time-scale  $\tau_0$ , but unstable over a short time-scale  $\tau_f < \tau_0$  in the excited state (characterized by an elevated temperature  $T_f > T_0$ ). To model this trade-off in stability, we consider the utility function (unit %)

$$f(\mathbf{x}; \boldsymbol{\kappa}) = \alpha N(\mathbf{x}; \boldsymbol{\kappa}, \tau_f, T_f) - (1 - \alpha)N(\mathbf{x}; \boldsymbol{\kappa}, \tau_0, T_0),$$

where  $\alpha \in [0, 1]$  is an adjustable parameter. By maximizing  $f$  over the space of tunable parameters  $\mathbf{x}$ , we determine the initial conditions that yield the best trade-off in stability. If  $\alpha = 0$ , the optimization yields conditions that minimize the amount of payload delivered in the normal state, while  $\alpha = 1$  results in the maximization of the payload delivered.

**3.2. Nanoemulsion optimization as a ranking and selection problem.** In the nanoemulsion stability problem, we consider the choice of the five continuous, controllable parameters as our set of alternatives. In order to frame this problem as an RS problem, we must discretize this five dimensional space of continuous parameters. Throughout, the discretization used is a uniform grid in which the continuous intervals describing each parameter is discretized into a certain number of equally spaced points throughout that interval. Through this, we obtain  $M$  distinct 5-dimensional vectors which we think of as discrete choices of the tunable controllable parameters such as initial volume fraction, oil droplet diameter, etc. In the language of ranking and selection problems in general, these  $M$  vectors represent  $M$  alternatives  $x \in \mathcal{X}$ . Note that by performing such discretization, we are ignoring the fact that the control variables are indeed continuous. However, as the sampling becomes denser, we obtain a better approximation of the continuous smooth space. In experimental science, control variables are often

viewed as discrete given experimental precision. For example, the resolution of a thermometer is usually 0.1K and calibrated to within 0.3K[4].

For the nanoemulsion study, our goal is now to find out the best alternative or the best parameter choices that could allow us to construct the system with desired properties. If we knew the correct value for  $\kappa$ , we could find the best value of  $\mathbf{x}$  by using our numerical model discussed in the last section. However, we do not know  $\kappa$ , so we need to design experiments to learn the most likely value. We assume that the sample measurements  $\hat{y}_{\mathbf{x}}^{n+1}$  for alternative  $\mathbf{x}^n = \mathbf{x}_i$  are of the form

$$\hat{y}_{\mathbf{x}^n}^{n+1} = \mu_i + W^{n+1} = f(\mathbf{x}_i; \kappa^*) + W^{n+1},$$

where  $W^{n+1} \sim \mathcal{N}(0, \sigma^2)$  is the noise with known variance  $\sigma^2$ , and  $\mu_i = f(\mathbf{x}_i; \kappa^*)$  is the unknown true value of running the experiment using alternative  $\mathbf{x}_i$  at time  $n$ . For our nanoemulsion example, the true utility values is  $\mu_{\mathbf{x}} = f(\mathbf{x}; \kappa^*)$ , the process of choosing a measurement policy maximizing the expected reward is now given by

$$\sup_{\pi \in \Pi} \mathbb{E}^\pi f(\mathbf{x}^N; \kappa^*).$$

**4. KGDP in Nanoemulsion stability.** In this section, we discuss the application of KGDP to the nanoemulsion problem and illustrate how to use KGDP to guide the experiments. We describe how we obtain our priors in Section 4.1, and illustrate our simulation procedures in Section 4.2. In Section 4.3, we present the empirical results on the performance of KGDP in the nanoemulsion problem setting. We discuss the proximity assumption in Section 4.4.

**4.1. Prior Generation.** Prior knowledge often comes from domain experts or from simulation results of mathematical models. In this study, we obtain our prior knowledge about the values of unknown parameters and tunable parameters by reviewing similar systems in the literature, obtaining estimates for the relevant kinetic parameters to within a few orders of magnitude. [22] studies three double emulsion systems that use dodecane as oil phase and different lipophilic and hydrophilic surfactants, such as Arlacel P135 and Synperonic PE/F68, Span 80 and SDS. [19] focuses water-in-shale oil single emulsion with 1.5% of Span 60/Tween 60, while [2] analyzes soybean oil-in-water emulsion which stabilized by 0.3% (w/w) ethoxylated monoglyceride (EOM) or 0.01% (w/w) bovine serum albumin (BSA). Several different systems with various particles dispersed in aqueous solution have been examined in [6]. A summary of the possible values in literature are shown in Table 2.

For visualization, we consider all but two of the tunable parameters fixed. We vary the initial water droplet diameter  $d_w^0$  and the oil droplet diameter  $d_o$ . The remaining parameters are fixed as follows:  $\phi_w^0 = 0.2$ ,  $\phi_o^0 = 0.12$ , and  $V_e^0 = 10000\mu\text{m}^3$ . The droplet diameters  $d_w^0$  and  $d_o$  were varied between 0.3 to 1μm and 5 to 10μm, respectively. The range of  $d_w^0$  (and  $d_o$ ) is discretized in to 21 (and 8) equally spaced points for calculation. The various values of unknown parameters are also summarized in Table 2. While the literature provides an estimate of the kinetic parameters for similar systems, the obtained ranges for some of these parameters were adjusted to reflect domain experts' prior knowledge of the specific systems of interest. To generate a sample  $\kappa_i$  of kinetic parameters, we uniformly sample energy barriers  $E_{\text{ripe}}$ ,  $E_{\text{coal}}$ ,  $E_{\text{floc}}$  and  $\Delta E_{\text{sorp}}$  in their respective ranges. The same was done for the

Table 2: A survey of parameters in different systems.

Parameters	Unit	Values in Literature ( $T = 298K$ )	Values in Experiments ( $T = 298K$ )
$\phi_w^0$	n/a	0.2 [22]	0.2
$\phi_o^0$	n/a	0.18 [22], 0.12 [22]	0.12
$d_w^0$	$\mu m$	0.36 [22]	0.3 - 1
$d_o$	$\mu m$	4 [22], 3.6[22]	5 - 10
$V_e^0$	$\mu m^3$	n/a	10000 (nominally large)
$E_{ripe}$	eV	0.51 [22]	0.5 - 0.6
$E_{floc}$	eV	0.29 [19]	0.1 - 0.5
$k_{floc}$	$\mu m^3 s^{-1}$	$1.67 \times 10^0$ [19], $4.26 \times 10^2$ [2], $1.67 \times 10^2$ [2], 6.5 - 12.6 [6], 4 - 7.5 [6]	1 - 100
$k_{coal}^0$	$s^{-1}$	$1.13 \times 10^9$ [22], $1.23 \times 10^{-2}$ [19]	†
$E_{coal}$	eV	0.76 [22], 0.11 [19]	0.8 - 0.9
$k_{coal}$	$s^{-1}$	$1.49 \times 10^{-4}$ [22], $1.68 \times 10^{-4}$ [19], $7.28 \times 10^{-6}$ [2], $2.5 \times 10^{-4}$ [2]	$10^{-2.5} - 10^{-2}$
$\Delta E_{sorp}$	$s^{-1}$	n/a	-0.1 - 0.1

†We sample the rates and energy barriers and calculate the prefactor values by Equation (3.2), (3.4) and (3.5).

kinetic rates constants  $k_{ripe}$ ,  $k_{coal}$  and  $k_{floc}$ . From this, the rate prefactors  $k_{ripe}^0$ ,  $k_{coal}^0$  and  $k_{floc}^0$  were obtained using Equations (3.2), (3.4) and (3.5). Setting  $T = 298K$ , we sample rate constants rather than prefactor because such rate constants were more frequently reported in the literature. Through this procedure, we obtain a sample  $\kappa_i$  of kinetic parameters which we use to define a candidate truth  $f_i = f(x, \kappa_i)$ .

In our study, we select  $L = 50$ , i.e. a set of 50 candidate truths of the utility function. Each of these candidates corresponds to a different unknown parameter vector  $\kappa_i$  sampled according to the above method. By fixing  $\alpha$  to be 0.5 and solving the ODEs given in Section 3.1 (e.g. Equation (3.1) to (3.6)), each candidate has the form of

$$f(\mathbf{x}; \kappa) = 0.5N(\mathbf{x}; \kappa, T_f, \tau_f) - 0.5N(\mathbf{x}; \kappa, T_0, \tau_0),$$

where  $T_f = 325K$ ,  $\tau_f = 1800s$ ,  $T_0 = 298K$ , and  $\tau_0 = 18000s$  reflecting normal and excited states. In practice, instead of measuring  $f(\mathbf{x}; \kappa)$  exactly, a scientist can only measure the percentages of payload delivered  $N(\mathbf{x}; \kappa, T_f, \tau_f)$  and  $N(\mathbf{x}; \kappa, T_0, \tau_0)$  up to some Gaussian noise with variance  $\epsilon^2$ . This includes a measurement noise on the utility function  $f$  with corresponding variance

$$\sigma^2 = \alpha^2 \epsilon^2 + (1 - \alpha)^2 \epsilon^2.$$

In our case,  $\sigma^2 = 0.5\epsilon^2$ .

By using the sampled unknown parameter  $\kappa_i$  and numerically solving the system described in Section 3.1, we get a candidate truth  $f(\mathbf{x}; \kappa_i)$  corresponding to  $\kappa_i$ . Figure 4 illustrates four candidate examples of the utility function. Each of these candidates corresponds to a different set of unknown kinetic parameters, which determines the location of the maximum utility. A

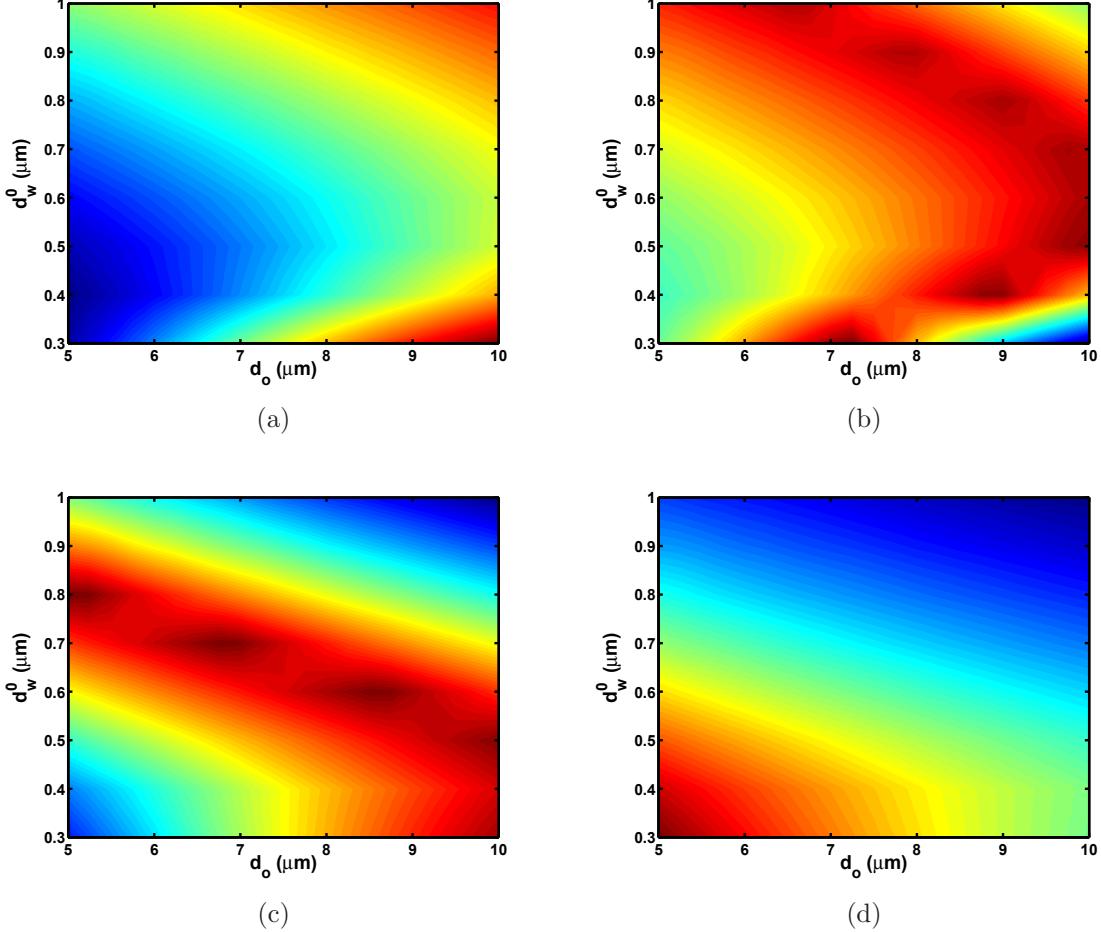


Figure 4: Example candidate truths with different  $\kappa$  values. The x- and y-axis correspond to the initial water droplet diameter  $d_w^0 \in [0.3, 1]\mu\text{m}$  and the oil droplet diameter  $d_o \in [5, 10]\mu\text{m}$ . The color represents the relative values of the utility function. The color red represents a high utility values while blue indicated a relatively low utility.

change of unknown kinetic parameters can move the maximum utility across the entire domain. For example, the maximum occurs near the lower right corner ( $d_w^0 = 0.3\mu\text{m}$  and  $d_o = 10\mu\text{m}$ ) in Figure 4a, while it appears in the lower left corner ( $d_w^0 = 0.3\mu\text{m}$  and  $d_o = 5\mu\text{m}$ ) in Figure 4d. Since the location maximum is determined by the unknown kinetic parameter, correctly identifying the unknown parameters can help us find out the maximum utility location.

**4.2. Simulation Procedure.** In this section, we discuss how KGDP can be used to guide the nanoemulsion experiments and how we conduct our simulations. In the simulation, we first assume the truth values of the unknown parameters is  $\kappa^*$ , which determine the true values of utility function  $f(\mathbf{x}; \kappa^*)$ . This truth is unknown to the measurement policy, which

**Algorithm 1** Algorithm for simulating KGDP

**Require:** Inputs the budget of  $N$  measurements, measurement noise  $\sigma$ ,  $L$  candidate truths  $f_1(\mathbf{x}), \dots, f_L(\mathbf{x})$  and truth kinetic parameters  $\kappa^*$ .

**for**  $n = 0$  to  $N - 1$  **do**

    Calculate KG values for all alternatives  $\mathbf{x} \in \mathcal{X}$  according to

$$\begin{aligned}\nu^{\text{KGDP},n}(\mathbf{x}) &= \sum_{j=1}^L \int_{w \in \Omega_2} \max_{\mathbf{x}'} \frac{1}{c_j} \sum_{i=1}^L f_i(\mathbf{x}') \exp \left[ -\frac{(f_j(\mathbf{x}) - f_i(\mathbf{x}) + w)^2}{2\sigma^2} \right] p_i^n g(w) dw p_j^n \\ &\quad - \max_{\mathbf{x}'} \sum_{i=1}^L \bar{f}_i(\mathbf{x}') p_i^n\end{aligned}$$

    Select the alternative  $\mathbf{x}^n \in \arg \max_{\mathbf{x} \in \mathcal{X}} \nu^{KG,n}(\mathbf{x})$

    Make a noisy measurement on  $\mathbf{x}^n = \mathbf{x}$  and obtain  $y_{\mathbf{x}}^{n+1} = f(\mathbf{x}, \kappa^*) + W^{n+1}$ , where  $W^{n+1} \sim N(0, \sigma^2)$

    Update the weights using

$$p_i^{n+1} = \frac{\exp \left[ -\frac{(\hat{y}_{\mathbf{x}}^{n+1} - f_i(\mathbf{x}))^2}{2\sigma^2} \right]}{\sum_{j=1}^L \exp \left[ -\frac{(\hat{y}_{\mathbf{x}}^{n+1} - f_j(\mathbf{x}))^2}{2\sigma^2} \right]} p_j^n,$$

**end for**

needs to discover the truth by making sequential measurements  $y_{\mathbf{x}^n}^{n+1} = f(\mathbf{x}^n; \kappa^*) + W^{n+1}$  for  $n = 0, \dots, N - 1$ .

In our simulations, the prior mean estimation of the truth is given by the sum of the 50 equal weighted candidates obtained from the sampling procedure outlined in the previous section. That is, the discrete distribution is  $p_i = \frac{1}{L}$  for all  $i$ . One example of such a prior mean is shown in Figure 5a. The simulation procedure is summarized in Algorithm 1. We start with the  $L$  candidate truths, the known measurement noise and a budget of  $N$  measurements. Then we calculate the KGDP values for each alternative according to Equation (2.4). An experiment is selected by choosing the one that maximizes the KGDP values. We generate a noisy observation of the selected alternative by adding a random noise to the true function value. The prior is then updated accordingly. By repeating this process, we expect the prior mean estimation to converge to the truth asymptotically. In the KGDP calculation step, we use numerical integration to approximate the integral over the noise  $W^{n+1}$ . Since the noise is a scalar random variable, this evaluation is straightforward.

**4.3. Empirical Results.** In this section, we consider the simulation results using the KGDP algorithm described in Algorithm 1. First, we examine how well the KGDP estimates the true function after a small number of measurements. We then study the KGDP performance by analyzing the effect of different measurement noises.

An example of a simulation of the KGDP policy is illustrated in Figure 5 graphically. The figures in the left column of Figure 5 are the prior mean estimates at time  $n$  while

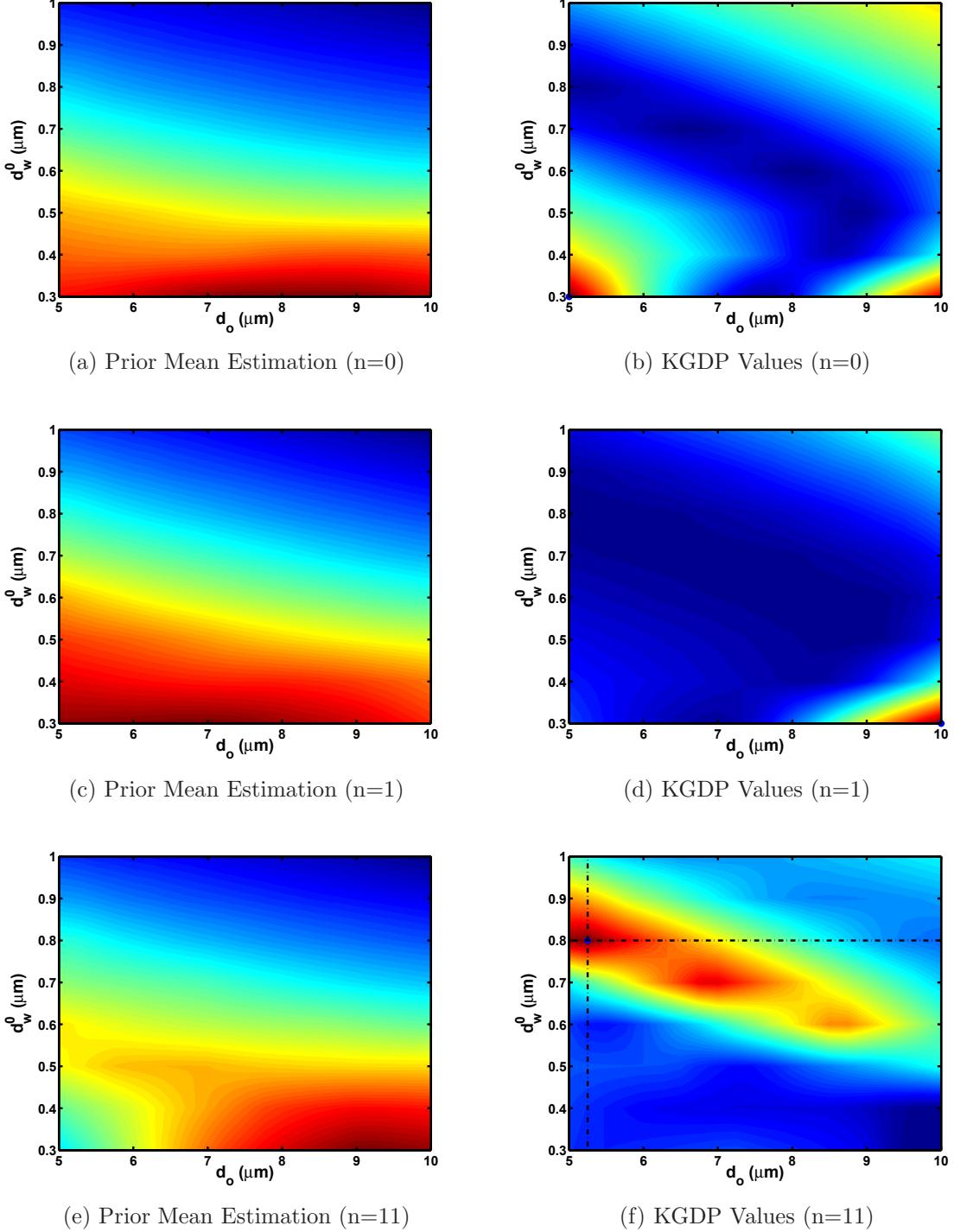


Figure 5: Prior mean estimation and KGDP Values at  $n = 0, 1, 10$ . The figures in left column are the prior mean estimates at time  $n$  while those in the right column are the corresponding KGDP values. The x- and y-axis correspond to the initial water droplet diameter  $d_w^0 \in [0.3, 1]\mu\text{m}$  and the oil droplet diameter  $d_o \in [5, 10]\mu\text{m}$ . The color indicates the relative value of the utility, red indicates a high utility value and blue corresponds to low utility value. The KGDP value plots are in a similar manner. The blue circles in the KGDP plots are the maximum KGDP value in the corresponding domain.

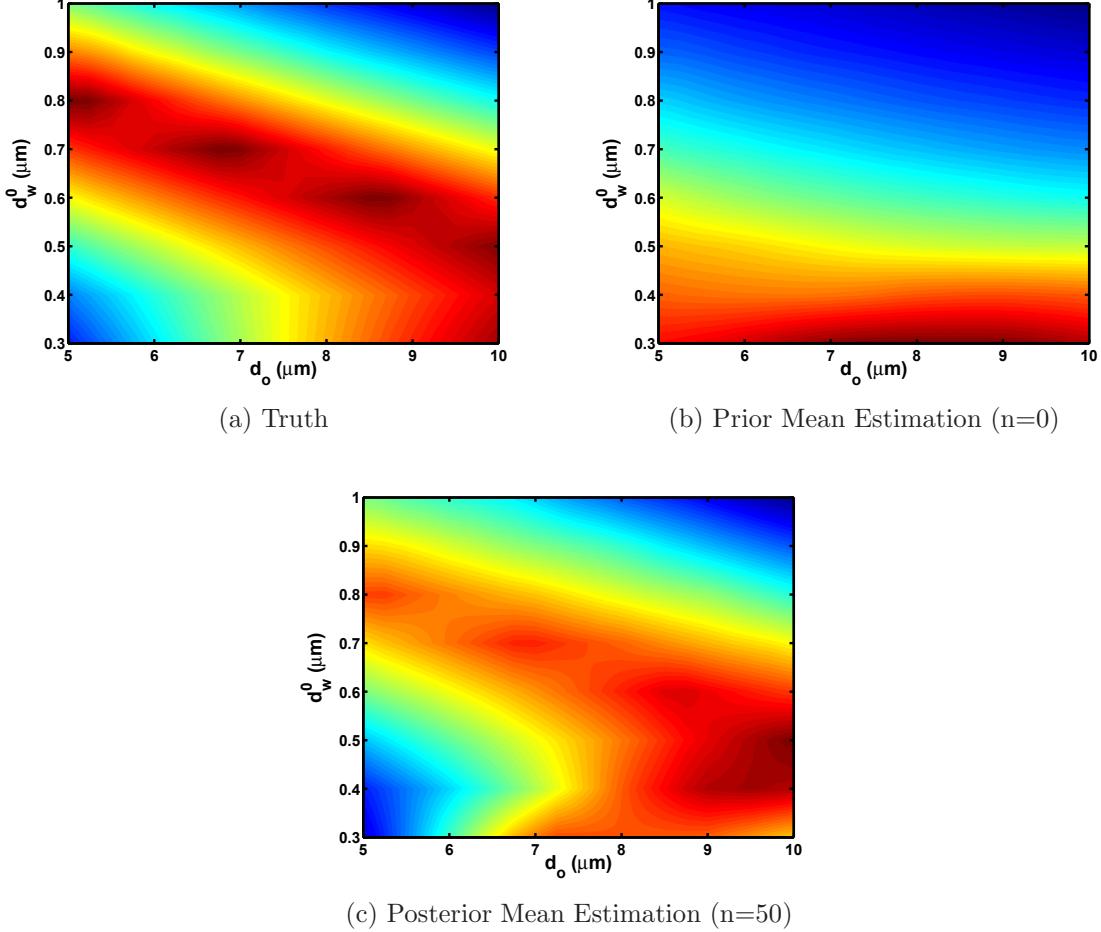


Figure 6: Comparison among the truth, prior mean estimate and posterior mean estimate after 50 measurements. The x- and y-axis correspond to the initial water droplet diameter  $d_w^0$  and the oil droplet diameter  $d_o$ . Starting with the given prior, the posterior mean converges to the truth within 50 measurements in a high noise case ( $\epsilon = 0.2$ ).

those in the right column are the corresponding KGDP values. The prior mean estimation,  $\bar{f}^n(x) = \sum_1^L f(x; \kappa_i)p_i^n$ , is plotted for  $\mathbf{x} = (d_w^0, d_o)$ , where  $d_w^0 \in [0.3, 1]\mu\text{m}$  and  $d_o \in [5, 10]\mu\text{m}$ . The color indicates the relative value of the utility, red indicates a high utility value and blue corresponds to low utility value. The KGDP value plots are in a similar manner.

In the simulation, we start with a prior mean and its corresponding KGDP values at time  $n = 0$ , Figure 5a and 5b. According to Figure 5b, KGDP suggests that we try the alternative with  $d_w^0 = 0.3\mu\text{m}$  and  $d_o = 5\mu\text{m}$ , the alternative with the highest KGDP value. In this figure, we note that the region in which our prior mean predicts to have high utility values (the broad red band) corresponds to the region of small KGDP values. This suggests that an exploitation

strategy, in which we simply pick the experiment that is predicted to do well according to the prior mean, does not gain much information from a single measurement. That is, if we pick the apparent best alternative, and are wrong about this guess, we have not learned much about the truth. This highlights the advantage of KGDP over other policies that do not consider the amount of information gained from an experiment.

While the KGDP values indicate those experiments that yield a high value of information, in practice such experiments may be undesirable to perform. The benefit of KGDP is that all potential experiments are scored, allowing a scientist to pick a more feasible experiment that results in a moderate (albeit non-optimal) amount of information gained. In this way, the KGDP values serves as a roadmap to the scientists, and hence the KGDP algorithm can be incorporated into the iterative work flow of a scientist.

After making the measurement decision, we generate the measurement by adding noise to the truth and perform a Bayesian update on our discrete distribution using Equation (2.3). For example, measuring  $\mathbf{x} = (0.3, 5)\mu\text{m}$  allows us to update our prior mean of time  $n = 0$  from Figure 5a to the corresponding posterior mean estimate, Figure 5c. We then iterate by considering this posterior mean estimate as the prior mean estimate at time  $n = 1$ . The corresponding KGDP plot is now Figure 5d. By repeating this process 10 times, we make 10 measurements, perform 10 Bayesian updates of the prior. The resulting posterior mean estimation at time  $n = 10$  is shown in Figure 5e. Figure 6 plots the truth, the prior mean estimate at  $n = 0$ , and the posterior mean estimate after 50 measurements. Although the prior mean we start with does not match the truth at all, we find that the posterior mean looks very similar to the truth after only 50 measurements. This shows that the posterior mean converges to the truth with a reasonable speed and a small amount of experiments, which is desirable in real-world applications like the nanoemulsion study.

To judge how well KGDP does in performing the optimization and in learning about the underlying kinetics, we consider two criteria, opportunity cost and kinetic rate error. Opportunity cost (OC) shows a policy's ability to find the optimal utility while rate error indicates its ability to learn the underlying kinetic model. In the following section, we demonstrate the effectiveness of KGDP by simulating its performance and that of other policies, and using our ability within controlled simulations to assess the ability of each policy to find the best set of experimental parameters against the optimal design (using our simulated known truth). It is important to emphasize that due to the small and limited measurement budget (as is motivated by the real world experimental setting), the initial performance of a policy is far more important than its asymptotic behavior.

**4.3.1. Performance Based on Opportunity Cost.** Opportunity cost (OC) is defined as the difference between the value of the alternative that is actually best and the true value of the alternative that is best according to the policy's posterior belief distribution, i.e.

$$\text{OC}(n) = \max_{\mathbf{x}} f(\mathbf{x}; \boldsymbol{\kappa}^*) - f(\arg \max_{\mathbf{x}} \bar{f}(\mathbf{x}); \boldsymbol{\kappa}^*).$$

When the opportunity cost is zero, the policy has found the best alternative. This quantity may only be calculated via simulations in which  $\boldsymbol{\kappa}^*$  is known. For illustrative purposes, we

compare the percentage OC with respect to the optimal value,

$$\text{OC\%}(n) = \frac{\max_{\mathbf{x}} f(\mathbf{x}; \boldsymbol{\kappa}^*) - f(\arg \max_{\mathbf{x}} \bar{f}(\mathbf{x}); \boldsymbol{\kappa}^*)}{\max_{\mathbf{x}} f(\mathbf{x}; \boldsymbol{\kappa}^*)}.$$

This normalization gives us a unit-free representation of a policy's performance. By taking the average percentage OC over several simulations, we can estimate the policy's average performance in practice.

Figure 7 plots the average percentage OC over 50 simulations as a function of the number of measurements using two values for the measurement noise on  $N(\mathbf{x}, \boldsymbol{\kappa}, T, \tau)$  ( $\epsilon = 10\%$  and  $\epsilon = 20\%$ ). We compare KGDP with the pure exploration policy, which selects a random experiment to run, and the pure exploitation policy, which chooses the experiments with the maximum estimated utility value. As both figures show, KGDP outperforms both exploration and exploitation, with lower average opportunity costs most of the time. In particular, when the budget is small and the noise is large (e.g.  $\epsilon = 20\%$ ), KGDP results in a significant reduction in the opportunity cost in comparison to the other policies. It takes about 20 experiments for KGDP to get less than 5% from the optimal, while the pure exploitation and pure exploration policies are approximately 10% and 13% from the optimal after 20 experiments. With a medium noise level (e.g.  $\epsilon = 10\%$ ), KGDP still does better than exploration and exploitation by requiring only one third of the experiments that others need to get to 5% from the optimal. Note that in Figure 7b pure exploitation outperforms KGDP by about 1% when the experimental budget becomes very large. All policies we compare here are updated using Equation (2.3), which involves the evaluation of the likelihood function. As the number of data points increases, the likelihood function performs better in discriminating the actual truth from other candidate truths, resulting in a better estimation for all policies. However, since our measurement budget is always small in practice, the initial performance of a policy is more important to us than its asymptotic behavior. KGDP outperforms the other two policies regarding each initial performance. When noise is significantly smaller (e.g.  $\epsilon = 1\%$ ), the OCs for all policies drop to nearly zero within 5 experiments. This case is not depicted in Figure 7.

**4.3.2. Performance Based on Kinetic Rate Error.** The other performance metric we use is the kinetic rate error, which measures the difference between the kinetic rates of key processes as determined by the underlying true kinetic parameters and those determined by the time- $n$  estimate of the kinetic parameters, as derived from the posterior discrete distribution  $\mathbf{p}^n$ . Specifically, we consider the three errors

$$\begin{aligned} Err_{\text{ripe}} &= |\log k_{\text{ripe}}^* - \log k_{\text{ripe}}^n|, \\ Err_{\text{coal}} &= |\log k_{\text{coal}}^* - \log k_{\text{coal}}^n|, \\ Err_{\text{floc}} &= |\log k_{\text{floc}}^* - \log k_{\text{floc}}^n|, \end{aligned}$$

where  $k_{\text{ripe}}^*$ ,  $k_{\text{coal}}^*$ ,  $k_{\text{floc}}^*$  are the values of the kinetic rates defined in Equations (3.2), (3.4) and (3.5), respectively, using the true kinetic parameters  $\boldsymbol{\kappa}^*$ . The estimated values,  $k_{\text{ripe}}^n$ ,  $k_{\text{coal}}^n$ ,  $k_{\text{floc}}^n$  are obtained from the same equations, using the linear estimation for kinetic parameters

$$\boldsymbol{\kappa}^n = \sum_{i=1}^L \boldsymbol{\kappa}_i p_i^n.$$

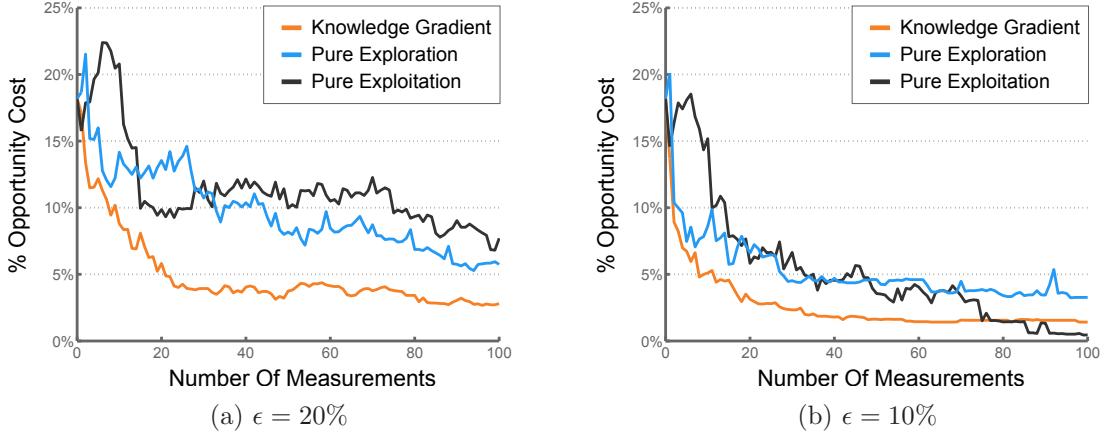


Figure 7: Average opportunity cost over 50 runs with 50 candidate truths and two different noise levels.

Due to the form of the above expressions, the reported numbers are statements about the order-of-magnitude of the error between the true and estimated values.

The rate error results are illustrated in Figure 8 with  $\epsilon = 20\%$ . The decreasing rate error indicates our ability to learn about the rates of coupled processes through scalar measurements. In general, KGDP continues to outperform pure exploration and exploitation by having a smaller rate error when the measurement noise is large. With the KGDP algorithm, the coalescence rate error is reduced by about 0.16 order-of-magnitude within 100 experiments, while pure exploration and exploitation can only reduce by 0.11 and 0.09 order-of-magnitude, respectively. Within the first 20 experiments, KGDP achieves an error of 0.21 order-of-magnitude away from the optimal while the exploration and exploitation can only achieve about 0.28 and 0.26 error on average. For discovering the flocculation rate, KGDP falls short at the beginning but catches up later and achieves a bigger error reduction within 100 experiments. And for the ripening rate, KGDP has a lower error at the beginning and working comparable to the pure exploration policy after 60 experiments.

Figure 9 compares the coalescence rate errors of different noise levels. KGDP performs better than both exploration and exploitation in reducing error. In the high noise case ( $\epsilon = 20\%$ ), the KGDP algorithm estimates the coalescence rate within 0.15 order-of-magnitude consistently after 70 experiments. The other two policies do not achieve this level of optimality within the scale of 100 experiments. In the medium noise case ( $\epsilon = 10\%$ ), the KGDP algorithm achieves the level of 0.15 order-of-magnitude from optimality within less than 20 experiments, while the exploration policy requires more than 50 experiments. Once again, exploitation cannot obtain this level of optimality. When the measurement noise is small ( $\epsilon = 1\%$ ), each measurement is fairly accurate and any measurement decision policies work equally well. Therefore KGDP performs similarly to pure exploration and exploitation, which is not included in Figure 9. From this, we see that KGDP can effectively learn about the

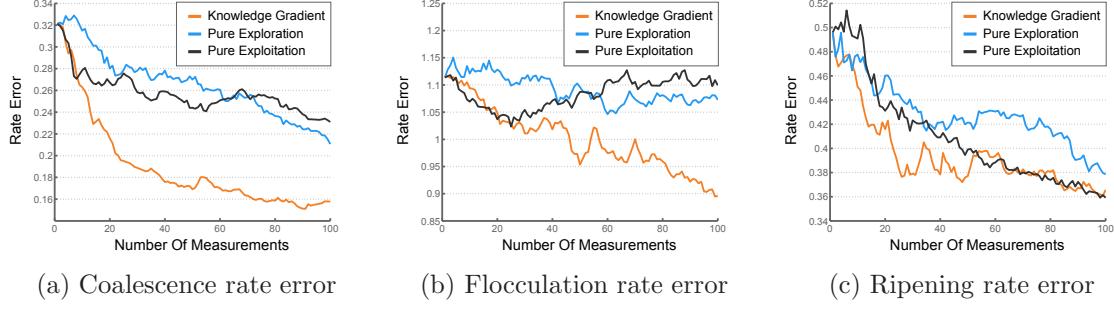


Figure 8: Average rate errors over 50 runs with 50 candidate truths and  $\epsilon = 20\%$ .

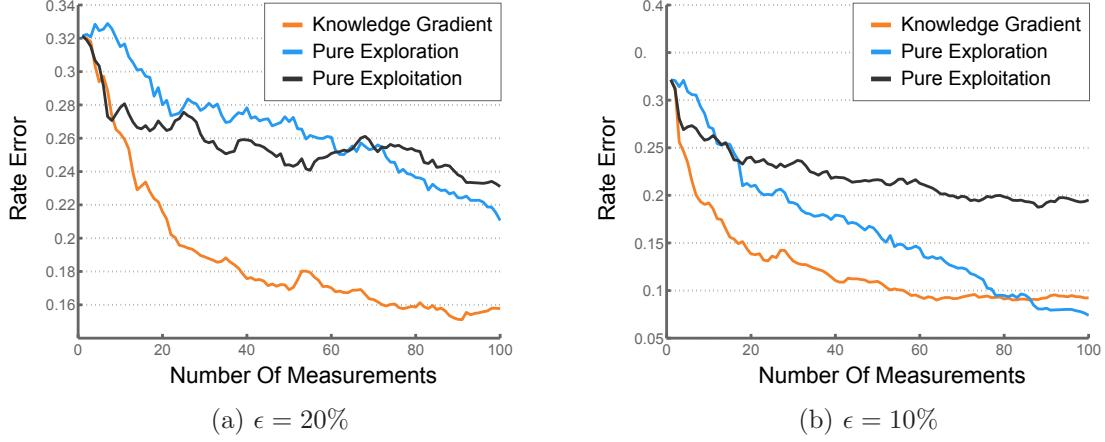


Figure 9: Average coalescence rate error over 50 runs with 50 candidate truths and two noise levels.

underlying kinetics while simultaneously performing the original optimization. This allows the experimentalists to gain scientific insight about the coupled kinetics of the material, while engineering an optimal configuration at the same time.

**4.4. Discussion on the Proximity Assumption.** Our proximity assumption of KGDP is that the truth is equal to or near one of the candidate truths. This assumption is similar to “ $M$ -closed framework” of [1], which is a general framework used in the model selection community [15] and assumes that one of the  $M$  candidate models is the truth. Note that  $M$  as used in this community is the same as our number of candidates, which we denote by  $L$ . The choice of the number of candidate truths,  $L$ , affects both the validity of the proximity assumption and the computational complexity of the KGDP. To assess the validity of the choice of  $L = 50$  in the above example application, we consider a sample of candidates

$C = \{\kappa_i, i = 1, \dots, L\}$  where  $\kappa_i \sim \mathcal{N}(\kappa^0, \Sigma^{\kappa,0})$  for  $i = 1, \dots, L$ . We then define the distance between a truth  $\kappa^*$  to a set of candidate parameters  $C$  to be

$$d(\kappa^*, C) = \min_{c \in C} \|\kappa_{\text{normalized}}^* - c_{\text{normalized}}\|_2,$$

where  $\kappa_{\text{normalized}}^* = (2\Sigma^{\kappa,0})^{-\frac{1}{2}}(\kappa^* - \kappa^0)$  and  $c_{\text{normalized}} = (2\Sigma^{\kappa,0})^{-\frac{1}{2}}(c - \kappa^0)$ . Though this normalization, the quantity  $\kappa_{\text{normalized}}^* - c_{\text{normalized}}$  is standard normally distributed, and is unitless, allowing for direct interpretation of the distance  $d(\kappa^*, C)$ . It is necessary to normalize the truths before comparison because of their different units and physical meanings cross dimensions. This minimum normalized distance gives us an idea of the difference between a set and a random truth.

We fix a set of  $L$  candidate truths  $C$ , and randomly generate 2000 truths according to the procedures discussed earlier in the section. From these sampled truths, we obtain the mean of  $d(\kappa^*, C)$  given  $L$ . We vary the size of the candidate truths, and plot the corresponding average of the minimum distance as a function of  $L$  in Figure 10. The mean minimum distance decreases as the number of candidates increases. And when  $L \rightarrow \infty$ , the minimum distance should approach zero. This provides evidence that our proximity assumption that one candidate  $c \in C$  is near or equal  $\kappa^*$  is true when we have infinitely many candidates. However, the computational difficulty of the KGDP calculation increases as the number of candidates increases. This is because the integral in KGDP cannot be calculated analytically and has to be approximated numerically, which will be repeated  $L$  times in the KGDP calculation for a single alternative. When  $L$  becomes large, this numerical procedure becomes time-consuming. And in fact, the minimum distance decreases at a slower rate as the number of candidates increases. When  $L = 50$ , the truth is within one standard deviation to the set of candidates, and hence provides a balance between the validity of our proximity assumption and the computational cost. Note that the actual distance depends on the structure of the covariance matrix, and is possible to be large even when its value is within one standard deviation.

The empirical results in this section are done under the proximity assumption that the actual truth is near one of the candidates. When the assumption is not satisfied, KGDP still outperforms the other policies we studied; however, KGDP performs best when one of the candidate truths is the actual truth. Further research is needed to handle the setting of high dimensional parameter spaces, where a small sample is unlikely to produce a candidate truth that is close to the actual truth.

**5. Conclusion.** Materials science challenges such as maximizing nanoemulsion stability involves expensive, time consuming experiments. Such problems can be described by nonlinear models that depend on several unknown parameters. We exploit this structure within an optimal learning framework, and use the knowledge gradient to maximize the value of information from each experiment. However, the knowledge gradient is computationally intractable when applied to problems with a belief model (the underlying kinetics) that is unknown in the nonlinear parameters.

We propose for the first time to use a sampled representation of the parameter space, and show that this allows us to represent the belief model in the form of a discrete set of

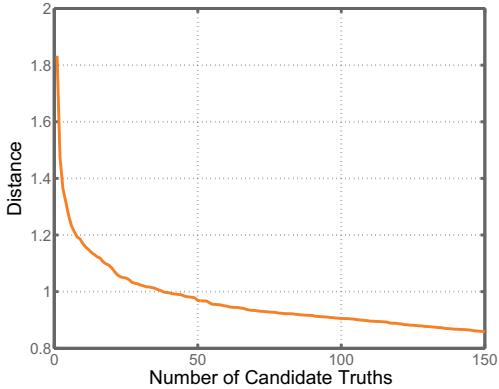


Figure 10: Average minimum distance as a function of number of candidate truths  $L$ . The distance from any truths to a set decreases as the size of the set increases. When  $L = 50$ , the distance is on average bounded by the unit ball center at zero, i.e. one standard deviation.

probabilities that each sampled parameter vector is correct. We then show that this allows the knowledge gradient to be computed quite easily. Controlled experiments generated around a known truth show that this method identifies the parameters much more quickly than standard exploration or exploitation heuristics. This method can be applied to general belief models with arbitrary structure.

## REFERENCES

- [1] JOS M. BERNARDO AND ADRIAN F. M. SMITH, *Bayesian Theory*, John Wiley & Sons, Sept. 2009.
- [2] RAJENDRA P. BORWANKAR, LLOYD A. LOBO, AND DARSH T. WASAN, *Emulsion stability kinetics of flocculation and coalescence*, Colloids and Surfaces, 69 (1992), pp. 135–146.
- [3] CHUN-HUNG CHEN, LIYI DAI, AND HSIAO-CHANG CHEN, *A gradient approach for smartly allocating computing budget for discrete event simulation*, in Simulation Conference, 1996. Proceedings. Winter, 1996, pp. 398–405.
- [4] FLUKE CORPORATION, *Fluke 51 & 52 series II thermometer user manual*.
- [5] FABIENNE DANHIER, EDUARDO ANSORENA, JOANA M SILVA, RGIS COCO, AUDE LE BRETON, AND VRONIQUE PRAT, *PLGA-based nanoparticles: an overview of biomedical applications*, Journal of controlled release: official journal of the Controlled Release Society, 161 (2012), pp. 505–522.
- [6] BOHUSLAV DOBIAS AND HANSJOACHIM STECHEMELLER, *Coagulation and Flocculation*, Second Edition, CRC Press, Dec. 2010.
- [7] PETER FRAZIER, WARREN POWELL, AND SAVAS DAYANIK, *The knowledge-gradient policy for correlated normal beliefs*, INFORMS Journal on Computing, 21 (2009), pp. 599–613.
- [8] PETER I. FRAZIER, WARREN B. POWELL, AND SAVAS DAYANIK, *A knowledge-gradient policy for sequential information collection*, SIAM J. on Control and Optimization, (2008).
- [9] SRINIVAS GANTA, HARIKRISHNA DEVALAPALLY, ALIASGAR SHAHIWALA, AND MANSOOR AMIJI, *A review of stimuli-responsive nanocarriers for drug and gene delivery*, Journal of controlled release: official journal of the Controlled Release Society, 126 (2008), pp. 187–204.
- [10] NISSIM GARTI, *Double emulsions scope, limitations and new achievements*, Colloids and Surfaces A: Physicochemical and Engineering Aspects, 123124 (1997), pp. 233–246.
- [11] N. GARTI, *New trends in double emulsions for controlled release*, in The Colloid Science of Lipids, B. Lind-

- man and B. W. Ninham, eds., no. 108 in Progress in Colloid & Polymer Science, Steinkopff, Jan. 1998, pp. 83–92.
- [12] ANDREW GELMAN, JOHN B. CARLIN, HAL S. STERN, AND DONALD B. RUBIN, *Bayesian Data Analysis, Second Edition*, CRC Press, July 2003.
- [13] JARROD A. HANSON, CONNIE B. CHANG, SARA M. GRAVES, ZHIBO LI, THOMAS G. MASON, AND TIMOTHY J. DEMING, *Nanoscale double emulsions stabilized by single-component block copolypeptides*, Nature, 455 (2008), pp. 85–88.
- [14] TREVOR J. HASTIE, ROBERT JOHN TIBSHIRANI, AND J. JEROME H. FRIEDMAN, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, Jan. 2001.
- [15] JOSEPH B KADANE AND NICOLE A LAZAR, *Methods and criteria for model selection*, Journal of the American Statistical Association, 99 (2004), pp. 279–290.
- [16] SEONG-HEE KIM AND BARRY L. NELSON, *Selecting the best system*, in Handbooks in Operations Research and Management Science, Shane G. Henderson and Barry L. Nelson, ed., vol. Volume 13 of Simulation, Elsevier, 2006, pp. 501–534.
- [17] MARKUS KREER AND OLIVER PENROSE, *Proof of dynamical scaling in smoluchowski's coagulation equation with constant kernel*, Journal of Statistical Physics, 75 (1994), pp. 389–407.
- [18] STEPHAN LINK AND MOSTAFA A. EL-SAYED, *Spectral properties and relaxation dynamics of surface plasmon electronic oscillations in gold and silver nanodots and nanorods*, The Journal of Physical Chemistry B, 103 (1999), pp. 8410–8426.
- [19] V. B. MENON AND D. T. WASAN, *Coalescence of water-in-shale oil emulsions*, Separation Science and Technology, 19 (1984), pp. 555–574.
- [20] DIANA M. NEGOESCU, PETER I. FRAZIER, AND WARREN B. POWELL, *The knowledge-gradient algorithm for sequencing experiments in drug discovery*, INFORMS Journal on Computing, 23 (2011), pp. 346–363.
- [21] BARBARA NIETHAMMER, *Macroscopic limits of the becker-dring equations*, Communications in Mathematical Sciences, 2 (2004), pp. 85–92. Mathematical Reviews number (MathSciNet) MR2119876.
- [22] K PAYS, J GIERMANSKA-KAHN, B POULIGNY, J BIBETTE, AND F LEAL-CALDERON, *Double emulsions: how does release occur?*, Journal of Controlled Release, 79 (2002), pp. 193–205.
- [23] O. PENROSE, *The becker-dring equations at large times and their connection with the LSW theory of coarsening*, Journal of Statistical Physics, 89 (1997), pp. 305–320.
- [24] WARREN B. POWELL, *Approximate Dynamic Programming: Solving the Curses of Dimensionality, 2nd Edition*, Wiley, Hoboken, N.J, 2 edition ed., Sept. 2011.
- [25] W. MARK SALTZMAN AND WILLIAM L. OLBRIGHT, *BUILDING DRUG DELIVERY INTO TISSUE ENGINEERING*, Nature Reviews Drug Discovery, 1 (2002), pp. 177–186.
- [26] MARTIEN A. COHEN STUART, WILHELM T. S. HUCK, JAN GENZER, MARCUS MLLER, CHRISTOPHER OBER, MANFRED STAMM, GLEB B. SUKHOUKOV, IGAL SZLEIFER, VLADIMIR V. TSUKRUK, MAREK URBAN, FRANOISE WINNIK, STEFAN ZAUSCHER, IGOR LUZINOV, AND SERGIY MINKO, *Emerging applications of stimuli-responsive polymer materials*, Nature Materials, 9 (2010), pp. 101–113.
- [27] JAMES R. SWISHER, SHELDON H. JACOBSON, AND ENVER YCESAN, *Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey*, ACM Trans. Model. Comput. Simul., 13 (2003), pp. 134–154.
- [28] M. VON SMOLUCHOWSKI, *Zur kinetischen theorie der brownschen molekularbewegung und der suspensio-nen*, Annalen der Physik, 326 (1906), pp. 756–780.
- [29] YU ZHANG, ZAIJIE WANG, AND RICHARD A GEMEINHART, *Progress in microRNA delivery*, Journal of controlled release: official journal of the Controlled Release Society, 172 (2013), pp. 962–974.