A Utile Function Optimizer

Christopher K. Monson, Kevin D. Seppi, and James L. Carroll

Abstract— **We recast the problem of unconstrained continuous evolutionary optimization as inference in a fixed graphical model. This approach allows us to address several pervasive issues in optimization, including the traditionally difficult problem of selecting an algorithm that is most appropriate for a given task. This is accomplished by placing a prior distribution over the expected class of functions, then employing inference and intuitively defined utilities and costs to transform the evolutionary optimization problem into one of active sampling. This allows us to pose an approach to optimization that is optimal for each expressly stated function class. The resulting solution methodology can optimally navigate exploration-exploitation tradeoffs using well-motivated decision theory, while providing the process with a natural stopping criterion. Finally, the model naturally accommodates the expression of dynamic and noisy functions, setting it apart from most existing algorithms that address these issues as an afterthought. We demonstrate the characteristics and advantages of this algorithm formally and with examples.**

I. Introduction

Continuous empirical optimization can be conceptualized as an inference process: given observed function values, conclusions are drawn about the hidden location of the global optimum. These conclusions are frequently drawn in the presence of uncertainty, where critical characteristics of the function are unknown and sampling produces sparse information.

This viewpoint suggests that the problem of optimization can be represented as a *graphical model* that describes its inherent information relationships [1], [2]. A graphical model is a probabilistic model that represents random variables with nodes and dependencies with edges. In this representation a problem's natural causality is easily represented and visualized. We refer to our model of evolutionary optimization as the Graphical Optimization Model (GOM).

Once defined, a GOM allows optimization to be conceptualized as a rational decision process: rather than assuming that sample locations are provided by some external source, an agent is created that selects them. In contrast to most other evolutionary algorithms based on probabilistic models, this rational agent will create populations based on both probabilistic models and well-defined utilities that reflect the goal of optimization: exploring the domain so as to acquire information about the global optimum.

The purpose of this work is to briefly introduce and explore this rich idea of inference-based evolutionary optimization, creating graphical models which describe the optimization process and developing them into a complete inference-based, evolutionary algorithm. The resulting algorithm is called the Utile Function Optimizer (UFO), and admits a principled analysis of optimization. We then provide a simple definition of utility and explore agents that use it to make rational sampling decisions, producing sophisticated behavior when provided with intuitive and straightforward declarations of practitioner intent.

This paper is organized as follows: In Section II we discuss related work. In Section III we review basic Bayesian techniques and graphical models. In Section IV we develop a graphical model of the evolutionary optimization problem and discuss the selection of its required prior distribution. In Section V we introduce optimal sampling in this model using utility theory. In Section VI we discuss how to implement this sampling technique and thus optimally select the membership of the next population using a specific utility model. In this section we also introduce the complete Utile Function Optimizer algorithm (UFO). In Section VII we summarize and clarify these concepts with a simple case study, and demonstrate how the various graphical model specifications can be produced. In Section VIII we discuss the problem of selecting an appropriate function class and compare it to the more traditional problem of selecting an appropriate algorithm. Finally, in Section IX we give conclusions and suggest future work.

II. Related Work

Optimization through Bayesian inference is not unique to this work: Stuckman [3] and Törn [4] survey a number of Bayesian optimization approaches. Mockus [5], [6] and Törn [4] also contribute important ideas to these methods. Many existing techniques rely on Gaussian Processes [7], including Kriging [8]. Particle Swarm Optimization has also been adapted to create and use distributions inferred through a Bayesian model [9], [10].

Other algorithms exist that specify a function class. In particular, meta models are used to tune an algorithm for a specified function class by reducing the need for expensive function evaluations [11]–[13].

Christopher K. Monson is with Google, Inc., 4720 Forbes Ave., Lower Level, Pittsburgh, PA 15213 (email: c@cs.byu.edu).

Kevin D. Seppi is with the Department of Computer Science, Brigham Young University, 3361 TMCB, Provo, UT 84602 (phone: 801-422-4619; email: k@cs.byu.edu).

James L. Carroll is with the Department of Computer Science, Brigham Young University, 3361 TMCB, Provo, UT 84602 (phone: 801-422-8717; email: jlcarroll@gmail.com).

1-4244-1340-0/07$25.00

Also related to this work is the field of function approximation. Linear, quadratic, Bayesian, and other forms of regression all relate to this work in the sense that data is used to draw conclusions about a function. In this work, however, the focus is limited to the discovery of the minimum, not the approximation of the entire function. In some cases a very good estimate for the location of the minimum can be obtained while leaving much about the rest of the function undiscovered.

Estimation of Distribution Algorithms (EDAs) generate distributions over likely minima [14]. While similar in spirit, EDAs and the UFO have a critical difference: EDAs rely on a *fixed* and usually *implicit* function class, encoded in the distribution representation and the algorithm used to obtain it; the Static Graphical Optimization Model relies on an *exchangeable* and *explicit* function class, encoded in the model distributions. This distinction will be discussed in greater detail later.

III. Bayesian Models

Given continuous random variables representing the state of the world x and a resulting observation y, Bayes' Law is used to compute the inverse relationship:

$$\rho_{x|y}(x|y) = \frac{\rho_{y|x}(y|x)\,\rho_x(x)}{\int \rho_{y|x}(y|x)\,\rho_x(x)\;dx}\;,$$

allowing conclusions to be drawn about x when observing only y. To do this, it is required that a *prior distribution* $\rho_x(x)$ and *likelihood* $\rho_{y|x}(y|x)$ be specified; the former represents what is believed about x in the absence of data, and the latter represents a belief about how the state of the world impacts observed data.

While it is possible to reason about these probabilities directly in the given notation, it is useful to depict the variables and relationships as a *graphical model*: a graph where nodes represent random variables and edges represent information dependencies [1]. In a directed graphical model (or Bayesian Network), the edges typically represent causal relationships, and every node in the model has an associated distribution: conditional if the node has parents, unconditional if not.

These models are typically acyclic when used for Bayesian inference. In this case, Bayes' Law and simple rules of probability can be combined to compute a distribution over any query variable given information about any number of evidence variables in the model [2]: $\rho(\text{query}|\text{evidence})$. Any variables that do not represent queries or evidence are integrated away in the inference process. This is a powerful tool for determining information about hidden state from available observations.

IV. Optimization as Inference

In the case of optimization, the development of a graphical model involves a declaration of available information and of the relationships that make inferring the location of the global optimum possible. These relationships can be extracted from the nature of the sampling process and straightforward notions of causality.

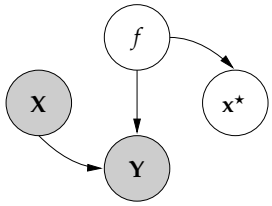


Fig. 1. The Static Graphical Optimization Model (SGOM)

The *sampling process* involves querying a function f at locations $\mathbf{X}$ to obtain corresponding values $\mathbf{Y}$. This process can be flexibly expressed as a conditional distribution $\rho_{\mathbf{Y}|f,\mathbf{X}}$. The *goal of optimization* is also important: to draw conclusions about the location of the global optimum $\mathbf{x}^\star$, a location that is defined by the function itself. This relationship may be described by the distribution $\rho_{\mathbf{x}^\star|f}$. An appeal to causality suggests that the direction of influence is correct: samples do not define the function, nor does its global optimum. All of these relationships are collected in the Static Graphical Optimization Model (SGOM) depicted in Figure 1, where the shading indicates the additional assumption that $\mathbf{X}$ and $\mathbf{Y}$ are observed while f and $\mathbf{x}^\star$ are hidden.

Using the SGOM, Bayes law, and other standard statistical techniques, optimization in the presence of sample information is simply the process of inferring the distribution over the location of the global optimum $\mathbf{x}^\star$, given the observed values of $\mathbf{X}$ and $\mathbf{Y}$:

$$\rho_{\mathbf{x}^\star|\mathbf{X},\mathbf{Y}}(\mathbf{x}^\star|\mathbf{X},\mathbf{Y}) = \frac{\int \rho_{\mathbf{x}^\star|f}(\mathbf{x}^\star|f)\,\rho_{\mathbf{Y}|f,\mathbf{X}}(\mathbf{Y}|f,\mathbf{X})\,\rho_{\mathbf{X}}(\mathbf{X})\,\rho_f(f)\,df}{\int \rho_{\mathbf{Y}|f,\mathbf{X}}(\mathbf{Y}|f,\mathbf{X})\,\rho_{\mathbf{X}}(\mathbf{X})\,\rho_f(f)\,df} \tag{1}$$

A. The Necessity of a Prior

Inference of $\rho_{\mathbf{x}^\star|\mathbf{X},\mathbf{Y}}$ requires the definition of the prior distribution ρ_f. This requirement is important because it indicates that useful conclusions about $\mathbf{x}^\star$ cannot be drawn without first specifying a distribution over all possible functions: a function class of interest. This necessity is unsurprising in light of No Free Lunch theorems (NFL) for optimization, which state that no (discrete) optimization algorithm has better average performance over all functions than random search [15]; if an algorithm *is* expected to perform better than random search, it must do so only for a limited class of functions. It will be demonstrated that in the SGOM, this class is specified precisely by ρ_f and $\rho_{\mathbf{Y}|f,\mathbf{X}}$.

The prior ρ_f may be expressed in many ways. It may be a discrete list of functions with associated probabilities, a distribution over the parameters of a fixed parametric representation (e.g., polynomial coefficients

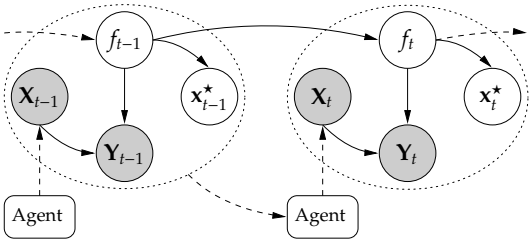


Fig. 2. The Graphical Optimization Model (GOM)

or artificial neural network weights [16]), a potentially unbounded data-based representation (e.g., a support vector machine [17]) or a hybrid of these. An example of one such representation will be given in Section VII.

V. Rational Sampling

Once a function class is defined, optimization is achieved through inference in the SGOM by solving (1). Happily, the resulting Bayesian posterior over global optima inherits the standard arguments for Bayesian optimality with respect to utility [18].

The optimality of the process itself is contingent upon the population of externally provided function samples. This raises the question of their origin. Therefore, the *decision process* involved plays as critical a role as does inference. In fact, it may be stated that a critical but underappreciated goal of optimization is to select sample locations $\mathbf{X}$ that will provide maximal information about $\mathbf{x}^\star$. This is the process of *active sampling*, also known as *active learning*, *active selection*, *experimental design*, or *optimal allocation of trials* [19]–[21].

Extending the SGOM to include an informed decision-making agent results in the complete Graphical Optimization Model (GOM) in Figure 2[1].

A rational optimizing agent will select samples that tend to benefit future generations. The utility of these sampling choices may be defined in a variety of ways, including simple definitions of the value of exploitation, measures based on Shannon information [20], [22], [23] and the Expected Value of Sample (or Partial) Information (EVSI) [24], [25]. Here we employ EVSI to assess the value of potential samples in the next generation; in applications where utility measures are available, samples based on EVSI will maximize expected utility. The next section gives an example of how to use EVSI to make rational decisions.

A. EVSI tutorial and example: Drilling For Oil

Consider an oil company that has an opportunity to purchase a plot of land [24] for \$1 million. If the site contains oil, it is worth \$10 million, that is, \$9 million net. Given this information, a utility function may be defined in terms of the boolean decision variable *Drill* and the boolean query variable *Oil*:

$$u_{Drill,Oil}(\mathrm{T},\mathrm{T}) = \$9\mathrm{M} \qquad u_{Drill,Oil}(\mathrm{T},\mathrm{F}) = -\$1\mathrm{M}$$
$$u_{Drill,Oil}(\mathrm{F},\mathrm{T}) = \$0 \qquad u_{Drill,Oil}(\mathrm{F},\mathrm{F}) = \$0 \ .$$

If the site's potential for containing oil is known in terms of the prior probabilities, e.g., $Pr_{Oil}(\mathrm{T}) = 0.6$, then the most rational action can be calculated using the Expected Utility (EU) of drilling ($d = \mathrm{T}$) and not drilling ($d = \mathrm{F}$):

$$\mathrm{E}_d\left[u_{Drill,Oil}(d,o)\right] = \sum_{o\in\{\mathrm{T},\mathrm{F}\}} u_{Drill,Oil}(d,o)\, Pr_{Oil}(o)$$
$$= \begin{cases} \$5\mathrm{M} & \text{if } d = \mathrm{T} \\ \$0 & \text{if } d = \mathrm{F} \end{cases} .$$

On average, drilling a plot like this one will produce a \$5 million profit, so the choice to purchase is rational. This result, however, says nothing about whether this *specific* plot should be purchased, only that the prior information suggests good return *on average*. Given such an isolated opportunity, a company would be unlikely to make a purchase based on such a result: it would gather more information first, perhaps in the form of a survey.

The presence of additional information improves the company's chances of making a profitable decision, so the availability of a geological survey ("test") transforms the problem from a purchase decision into a decision as to whether the cost of a test should be incurred. This is what EVSI computes. Let us suppose, then, that the test costs \$0.1 million and has the following characteristics:

$$Pr_{Test|Oil}(\mathrm{T}|\mathrm{T}) = 0.95 \qquad Pr_{Test|Oil}(\mathrm{T}|\mathrm{F}) = 0.20 \ .$$

The test is not very accurate, but may still provide useful information. The outcome of the test may affect the purchase decision, so expected utility is now conditional:

$$\mathrm{E}_{d|t}\left[u_{Drill,Oil}(d,o)\right] = \sum_{o\in\{\mathrm{T},\mathrm{F}\}} u_{Drill,Oil}(d,o)\, Pr_{Oil|Test}(o|t)$$

where $Pr_{Oil|Test}(o|t)$ is the Bayesian posterior

$$Pr_{Oil|Test}(o|t) = \frac{Pr_{Test|Oil}(t|o)\, Pr_{Oil}(o)}{Pr_{Test}(t)}$$

and the marginal on t is

$$Pr_{Test}(t) = \sum_{o\in\{\mathrm{T},\mathrm{F}\}} Pr_{Test|Oil}(t|o)\, Pr_{Oil}(o)$$
$$= 0.65$$

giving that

$$Pr_{Oil|Test}(\mathrm{T}|\mathrm{T}) \approx 0.88 \qquad Pr_{Oil|Test}(\mathrm{F}|\mathrm{T}) \approx 0.12$$
$$Pr_{Oil|Test}(\mathrm{T}|\mathrm{F}) \approx 0.09 \qquad Pr_{Oil|Test}(\mathrm{F}|\mathrm{F}) \approx 0.91$$

[1]The network has a structure similar to a Partially Observable Markov Decision Process (POMDP), but is fundamentally continuous, has non-stationary transitions, and is utilized quite differently than a traditional POMDP. Therefore, while noting the similarities, this work will not generally refer to the network as a POMDP.

and finally

$$\mathrm{E}_{d|t}\left[u_{Drill,Oil}(d,o)\right] = \begin{cases} \$7.8\mathrm{M} & \text{if } t = \mathrm{T}, d = \mathrm{T} \\ -\$0.1\mathrm{M} & \text{if } t = \mathrm{F}, d = \mathrm{T} \\ \$0 & \text{if } d = \mathrm{F} \end{cases} .$$

The presence of information changes the expected utility of drilling and therefore affects the purchase decision.

Knowledge of the test's outcome is desirable; it guides the decision process such that maximum expected utility (MEU) can be increased. The test itself costs \$0.1 million, however, making it useful to know the expected *improvement* in maximum utility given the test's outcome *before it is performed*. EVSI is the right way to calculate this while taking the accuracy of the test into account:

$$\begin{aligned} EVSI_{Test} &= \mathrm{E}\left[\max_d \mathrm{E}_{d|t}\left[u_{Oil,Drill}(o,d)\right]\right] \\ &\quad - \max_d \mathrm{E}_d\left[u_{Oil,Drill}(o,d)\right] \\ &= \mathrm{E}\left[\begin{cases} \$7.8\mathrm{M} & \text{if } t = \mathrm{T} \\ \$0 & \text{if } t = \mathrm{F} \end{cases}\right] - \$5\mathrm{M} \\ &= ((\$7.8\mathrm{M})\, Pr_{Test}(\mathrm{T}) + (\$0)\, Pr_{Test}(\mathrm{F})) - \$5\mathrm{M} \\ &= \$0.07\mathrm{M} \end{aligned}$$

Test results are therefore worth about \$0.07 million, but the test is not cost-effective because it costs \$0.1 million. Were the test more accurate, the expected value improvement might be higher; as it is, a lower price should be negotiated or a more accurate test utilized.

EVSI has answered an important question: whether or not to pay for this test before making a purchase decision. If two tests with distinct costs and accuracies are available, EVSI can also be used to choose between them based on their net expected utilities.

VI. A Utile Function Optimizer: EVSI in the GOM

EVSI is powerful and simple, indicating whether new information is expected to be worth its cost. In the process, it answers another supremely important question in function optimization: "How will immediate exploration affect future exploitation?" Optimization is fundamentally concerned with *finding* a location that, *if sampled*, will produce a good value, not with *actually sampling* that value. EVSI is a principled and meaningful way to accomplish this goal of active sampling for maximization of information about the global minimum, and it is only possible within a framework such as the GOM.

In the oil example, the question is whether to purchase (and drill) the land. The test (e.g., a geological survey) is distinct and has its own associated costs. It is conceivable, however, that the test could consist of purchasing and drilling the land; after all, doing so would yield excellent information. Function optimization is a similar situation; infinite sample locations (tests) are available, and the outcome of those tests will affect knowledge of $\mathbf{x}^\star$: the location of Maximum Expected Utility (MEU).

In optimization, the distinction between tests and decisions is the same as the distinction between exploration and exploitation: the test is performed in the hopes of obtaining more information, and the decision is made in order to obtain the actual value. EVSI computes the utility of *exploration*, and Expected Utility computes the utility of *exploitation*. Because they operate in the same domain, greater care must be taken with the notation.

The Utile Function Optimizer (UFO) employs EVSI in the context of the GOM, so the math of expectations will be used frequently throughout the rest of this work. We write the expectation thus:

$$\mathrm{E}_{y|x}\left[g(y,z)\right] = \int g(y,z)\rho_{Z|X}(z|x)\, dz \ , \tag{2}$$

where any variable *not* appearing in the subscript is integrated away. It is worth nothing that at least two popular opposing notations are used in practice, and that we have selected one that is consistent with the notations for probability and utility employed in this work. The reader who has prior experience with the opposing notation (where subscripts that *do* appear are integrated away) should therefore take care to perform the necessary mental reversals when examining the formulas that follow. Furthermore, we frequently employ a space-saving notational shortcut, denoting all evidence variables as $\mathbf{e}_t = (\mathbf{X}_{t-1}, \mathbf{Y}_{t-1})$.

A test variable and its outcome are distinguished by a superscript $^?$: the candidate population $\mathbf{X}_t^?$ represents a test, and its outcome is denoted $\mathbf{Y}_t^?$. The decision variable is also a population $\mathbf{X}_t$, representing locations that would be given high value by Expected Utility. The test is *concrete*, representing real samples, and the decision is *hypothetical* in the calculation of EVSI (as evidenced by the fact that decisions only appear in the context of the max operator). The query variable, representing desired information, must be one of the utility function parameters and is in this case f_t, from which $\mathbf{x}_t^\star$ is obtainable via the relationship $\rho_{\mathbf{x}^\star|f}$.

Assuming a candidate population of size 1, EVSI applied to the GOM is given as

$$\begin{aligned} EVSI_{\mathbf{x}_t^?|\mathbf{e}_t} &= \mathrm{E}_{|\mathbf{e}_t,\mathbf{x}_t^?}\left[\max_{\mathbf{x}_t} \mathrm{E}_{\mathbf{x}_t|\mathbf{e}_t,\mathbf{x}_t^?,\mathbf{y}_t^?}\left[u_{\mathbf{x}_t,f_t}(\mathbf{x}_t,f_t)\right]\right] \\ &\quad - \max_{\mathbf{x}_t} \mathrm{E}_{\mathbf{x}_t|\mathbf{e}_t}\left[u_{\mathbf{x}_t,f_t}(\mathbf{x}_t,f_t)\right] \ . \end{aligned} \tag{3}$$

Here EVSI uses evidence $\mathbf{e}_t = (\mathbf{X}_{t-1}, \mathbf{Y}_{t-1})$. Reorganizing to highlight required distributions yields

$$\begin{aligned} &\mathrm{E}_{\mathbf{x}_t|\mathbf{e}_t,\mathbf{x}_t^?,\mathbf{y}_t^?}\left[u_{\mathbf{x}_t,f_t}(\mathbf{x}_t,f_t)\right] \\ &\quad = \int u_{\mathbf{x}_t,f_t}(\mathbf{x}_t,f_t)\, \rho_{f_t|\mathbf{e}_t,\mathbf{x}_t^?,\mathbf{y}_t^?}\left(f_t|\mathbf{e}_t,\mathbf{x}_t^?,\mathbf{y}_t^?\right)\, df_t \end{aligned} \tag{4}$$

$$\mathrm{E}_{\mathbf{x}_t|\mathbf{e}_t}\left[u_{\mathbf{x}_t,f_t}(\mathbf{x}_t,f_t)\right] = \int u_{\mathbf{x}_t,f_t}(\mathbf{x}_t,f_t)\, \rho_{f_t|\mathbf{e}_t}(f_t|\mathbf{e}_t)\, df_t \ . \tag{5}$$

Additionally, computation of the outermost expectation in the first term of (3) requires specification of

Algorithm 1 Particle filter for the SGOM: produces an empirical $\rho_{f|\mathbf{X},\mathbf{Y}}$ for one instance of the static network

Given candidate population $\mathbf{X}$*, sample* $f^\star$ *to obtain results*
$\mathbf{Y} = (\mathbf{y}_1 \dots \mathbf{y}_{N_\mathbf{X}})^\top$, with $\mathbf{y}_i = f^\star(\mathbf{x}_i)$ for $\mathbf{x}_i \in \mathbf{X}$
Create particles and calculate the normalized likelihood for each
$\mathbf{P} = (f_1 \dots f_{N_\mathbf{P}})^\top$, with $f_i \sim \rho_{f_t}(f_t)$
$\mathbf{L} = \left(\mathcal{L}_{f_1|\mathbf{X},\mathbf{Y}} \dots \mathcal{L}_{f_{N_\mathbf{P}}|\mathbf{X},\mathbf{Y}}\right)^\top / \sum_{i=1}^{N_\mathbf{P}} \mathcal{L}_{f_i|\mathbf{X},\mathbf{Y}}$ with $\mathcal{L}_{f_i|\mathbf{X},\mathbf{Y}} = \prod_{j=1}^{N_\mathbf{X}} \rho_{\mathbf{y}_t|f_i,\mathbf{x}_t}\left(\mathbf{y}_j|f_i,\mathbf{x}_j\right)$
The result is a discrete representation of the function posterior:
$\rho_{f|\mathbf{X},\mathbf{Y}}(f|\mathbf{X},\mathbf{Y}) \coloneqq \mathbf{P},\mathbf{L}$

Algorithm 2 $EVSI_{\mathbf{x}^?|\mathbf{X}_{t-1},\mathbf{Y}_{t-1}}$ in the UFO using a particle filter ($\mathbf{x}^?$ is supplied and tested for information content)

Create empirical distributions as bags of values (chain rule)
$\mathbf{P} = (f_1 \dots f_{N_\mathbf{P}})^\top$, where $f_i \sim \rho_{f_t|\mathbf{e}_t}(\cdot|\mathbf{X}_{t-1},\mathbf{Y}_{t-1})$
$\mathbf{X} = (\mathbf{x}_1 \dots \mathbf{x}_{N_\mathbf{P}})$, where $\mathbf{x}_i \sim \rho_{\mathbf{x}_t^\star|f_t}(\cdot|f_i)$ for $f_i \in \mathbf{P}$
$\mathbf{Y}^? = \left(\mathbf{y}_1^? \dots \mathbf{y}_{N_\mathbf{P}}^?\right)$, where $\mathbf{y}_i^? \sim \rho_{\mathbf{y}_t|f_i,\mathbf{x}_t}\left(\cdot|f_i,\mathbf{x}^?\right)$ for $f_i \in \mathbf{P}$
Calculate EVSI
$T = \frac{1}{N_\mathbf{P}} \sum_{\mathbf{y}^? \in \mathbf{Y}^?} \max_{\mathbf{x} \in \mathbf{X}} \sum_{f \in \mathbf{P}} u_{\mathbf{x},f}(\mathbf{x},f)\, \rho_{\mathbf{y}_t|f_i,\mathbf{x}_t}\left(\mathbf{y}^?|f,\mathbf{x}^?\right)$
$M = \max_{\mathbf{x} \in \mathbf{X}} \sum_{f \in \mathbf{P}} u_{\mathbf{x},f}(\mathbf{x},f)\, \rho_{f_i|\mathbf{X}_{t-1},\mathbf{Y}_{t-1}}(f|\mathbf{X}_{t-1},\mathbf{Y}_{t-1})$
$EVSI_{\mathbf{x}^?|\mathbf{X}_{t-1},\mathbf{Y}_{t-1}} = T - M$

$\rho_{\mathbf{y}_i^?|\mathbf{X}_{t-1},\mathbf{Y}_{t-1},\mathbf{x}_i^?}$. All necessary distributions may be obtained from the network using empirical methods such as particle filters [2], the method of choice for this work (and the source of much of the computational complexity of this method). Computation of $\rho_{f|\mathbf{X},\mathbf{Y}}$ is performed as shown in Algorithm 1, and the method for computing EVSI in this context is supplied as Algorithm 2.

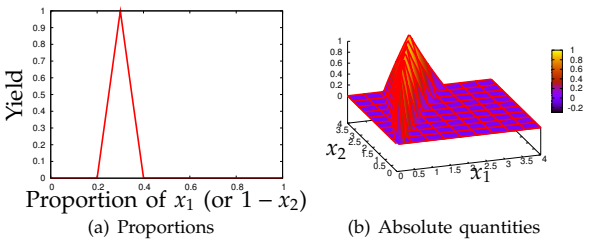
Application of this algorithm provides the practitioner with important information: where next to sample $f^\star$ to obtain maximal information about $\mathbf{x}^\star$.

The specification of function class and creation of a rational UFO that makes use of sampling cost and EVSI are best described with a simple example.

VII. Example: Optimal Chemistry

Consider a laboratory technician tasked with finding an optimal mixture of chemicals in a solution, where optimality is achieved by maximizing the percentage yield of a precipitate. The technician receives a commission based on achieved yield but must pay for all ingredients. Every experiment thus has a concrete cost (the cost in dollars of the ingredients) in the same units as the utility of its output (the dollar value of the commission).

This example will be used to describe each step of the setup procedure for the Graphical Optimization Model (GOM) and the creation of a corresponding Utile Function Optimizer (UFO), that is, representation of the function class, representation of the sample noise, representation of the goal of optimization, construction of a utility model for the output of the optimizer, and the construction of a cost model for sampling.



(a) Proportions (b) Absolute quantities

Fig. 3. The truncated cone function

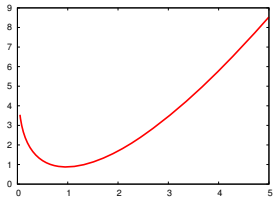


Fig. 4. $1 + \ln\Gamma(3x/2)$: Suitable for scaling $\sigma_\mathbf{y}$

A. Function Class

The technician's first task is to declare the function class. We will assume that this class contains *truncated cones* in the proportion space[2]:

$$f(\mathbf{x};\mathbf{a}) = \max\left\{0, 1 - \left\|10\left(\frac{\mathbf{x}}{\sum_{i=1}^{D} x_i} - \mathbf{a}\right)\right\|_2\right\} \quad (6)$$

$$\mathbf{a}_1 \sim N(0.5, 0.2) \text{ and } \mathbf{a}_2 = 1 - \mathbf{a}_1 \quad (7)$$

where $x_i, a_i > 0$ and $\sum_{i=1}^{D} a_i = 1$. The true function $f^\star$ is a member of this class with $\mathbf{a} = (0.3, 0.7)^\top$ as in Figure 3.

B. Sample Noise

An important aspect of the function class is sample noise, in this case due to measurement uncertainty. As ingredient quantities approach zero, accurate measurements become increasingly difficult to achieve. Uncertainty also increases as quantities become very large, since the error of measuring using available equipment compounds when multiple measurements must be taken to achieve larger quantities. As the measurements become very large, the experiments take on absurd characteristics: it is increasingly difficult to accurately measure vats, lakes, or oceans of constituent ingredient quantities.

Assuming that the sample noise is additive Gaussian, this implies that the standard deviation of noise changes with the absolute ingredient quantities. The size of that standard deviation might be conveniently described using a function like the shifted and scaled log-gamma shown in Figure 4, defined precisely within

[2]This model is purposely simplistic for pedagogical purposes; highly flexible models exist that, surprisingly, are only slightly more complex.

this specification of the likelihood:

$$\rho_{\mathbf{y}|f,\mathbf{x}}(\mathbf{y}|f,\mathbf{x}) = N\left(\mathbf{y} - f(\mathbf{x}), \sigma_{\mathbf{y}}\left(1 + \Gamma\left(\frac{3}{2}\frac{1}{D}\sum_{i=1}^{D} x_i\right)\right)\right) \quad . \quad (8)$$

This definition assumes that the most accurate measurement is scaled to 1 unit. It is also assumed that overall accuracy is a function of average quantities.

The distributions ρ_f (entirely represented by $\rho_{\mathbf{a}}$) and $\rho_{\mathbf{Y}|f,\mathbf{X}}$ are now defined, completing the definition of this "cone-like" function class: ρ_f dictates that the class contains cones, and $\rho_{\mathbf{Y}|f,\mathbf{X}}$ defines how far outside of that strict definition a sample value may be and still be considered consistent with the class.

C. Goal of Optimization

In this example, the goal of our optimization process is to find the maximum. Since the location of the maximum of any function in the truncated cone function class is $\mathbf{a}$, $\rho_{\mathbf{x}^\star|f}(\mathbf{x}^\star|f)$ should place maximal probability mass on the value $\mathbf{x}^\star = \mathbf{a}$, thus the goal distribution is simply:

$$\rho_{\mathbf{x}^\star|f}(\mathbf{x}^\star|f) \sim \delta(\mathbf{x}^\star - \mathbf{a}) \quad (9)$$

where δ represents Dirac's delta distribution, which places all of its density at the origin. That the maximum for any function in the class can be trivially extracted is not coincidental, but the product of a careful choice of function class representation. If the representation of ρ_f does not admit trivial extraction of the optimum from its parameters, the evaluation of ρ_f may hide a complex internal optimization problem. Addressing this directly is beyond the scope of this introductory material, but will be addressed in future work.

D. Output Utility

Compensation is a linear function of percentage yield: \$100 for an experiment producing 100% yield:

$$u_{f_t,\mathbf{x}_t}(f_t,\mathbf{x}_t) = \$100 f_t(\mathbf{x}_t) \quad . \quad (10)$$

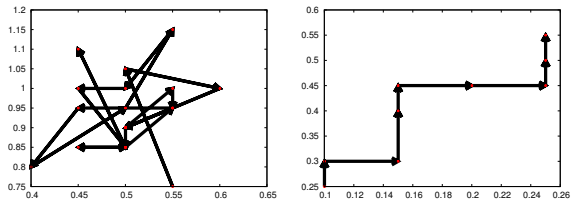
Even though the goal of the technician is to *maximize information* about regions of higher payoff, that goal is not directly reflected in the utility function. Instead, utility is defined as before, in terms of exploitation. It is unnecessary to explicitly define a utility of exploration. Instead, EVSI will be used to compute it.

E. Sample Cost

An obvious and trivial way to define sampling cost in this example is as the sum of the prices of the constituent ingredients that go into an experiment:

$$c = \sum_{i=1}^{D} c_i x_i \quad (11)$$

where c_i is the cost of a single unit of ingredient i. This is a straightforward definition of cost from the technician's perspective, who is required to buy his own ingredients.



(a) No cost specified (b) Cost specified

Fig. 5. Chemistry experiments with EVSI

When mixing a solution in order to achieve a precipitate, however, it is often possible to add ingredients *incrementally* to adjust the percentage yield, even after some amount of precipitate has been previously removed and measured. In other words, it is possible to perform a new experiment by *continuing an old one*, giving rise to a more interesting cost function:

$$c_t = \begin{cases} \sum_{i=1}^{D} c_i x_{t,i} & \text{if } \mathit{restart}_t \\ \sum_{i=1}^{D} c_i \left(x_{t,i} - x_{t-1,i}\right) & \text{otherwise} \end{cases} \quad (12)$$

where

$$\mathit{restart}_t = (\exists i.\ x_{t,i} < x_{t-1,i}) \vee (\mathbf{x}_t = \mathbf{x}_{t-1}) \quad . \quad (13)$$

If the technician desires to reduce an ingredient or to duplicate an experiment, the cost is calculated by totaling the cost of the constituents; the only way to repeat or reduce constituent quantities is to begin again. If, however, he wishes to adjust the balance by adding a small amount of an ingredient, the cost is the price of the additional material, not of the full solution.

F. Algorithm Behavior

The results of using EVSI for experiment selection in this section assume the following:

- Sample noise has the (scale) parameter $\sigma_{\mathbf{y}} = 0.1$,
- All ingredients cost \$10.00 per unit, and
- The best proportion is $\mathbf{a} = (0.3, 0.7)^\top$.

A number of interesting behaviors are observed when applying the UFO (EVSI in the GOM) to this problem.

Figure 5, for example, illustrates the path taken by EVSI through the space of absolute quantities: Figure 5(a) shows the path of experiments when EVSI is not aware of the sampling cost, and Figure 5(b) illustrates the path when costs are supplied. In the first case, it continues experimenting until a pre-specified maximum number of iterations has been reached (20 for this example) and does not appear to follow any particular pattern. In the second, it always adds ingredients incrementally and stops after 9 iterations. In both cases the resulting posteriors allocated more than 99% of the probability on solutions within 1% of the true optimal.

It is significant that the technician that does not supply a declaration of costs spends \$222.00 and uses all of

the 20 allowed experiments, and the technician that does supply cost spends $8.00 on ingredients before determining that 9 experiments are sufficient. Even if the first technician had stopped after 9 iterations, he would have spent $117.00 on ingredients.

The fact that the algorithm stopped the sampling process early is significant because the determination of a suitable stopping criterion is a common problem in empirical optimization. Generally it is assumed that one must put a cap on the number of samples taken for practical reasons, implying that it has an associated cost. The use of EVSI and the admission of an explicit cost specification makes the stopping criterion obvious: stop sampling (exploring) when its cost exceeds the expected improvement in utility. Then exploit the learned values.

It is also noteworthy that both sets of experiments choose absolute ingredient amounts that do not deviate too far from 1 unit of measurement. This is expected because ingredient quantities close to 0 and much higher than 1 have higher sample noise. Therefore, even though cost minimization is an important issue for this problem, taking measurements that are too small to admit much certainty does little to help the technician; the experimental process dictated by EVSI and the supplied declarations match this reasoning well.

One benefit of this approach is the ability to define various natural costs, including *opportunity cost*, defined as the expected loss due to lack of exploitation:

$$c_{\mathbf{x}_i^?} = \max_{\mathbf{x}_t} \mathrm{E}_{\mathbf{x}_t|\mathbf{e}_t}\left[u_{\mathbf{x}_t,f_t}(\mathbf{x}_t, f_t)\right] - \mathrm{E}_{\mathbf{x}_i^?|\mathbf{e}_t}\left[u_{\mathbf{x}_t,f_t}\left(\mathbf{x}_i^?, f_t\right)\right] \quad . \quad (14)$$

In the absence of other cost information, opportunity cost is a natural and easily-applied definition that at least takes into consideration the fact that exploration often precludes exploitation. For example, if the technician were adjusting controls on a production mixer, opportunity cost would be appropriate; exploration that does not produce immediately better output costs money because it keeps known better output from occurring. This idea is easily applied in the context of the GOM.

The sophisticated behaviors of the UFO, i.e., its tendency to minimize cost, stay away from noisy regions, and stop when sampling is no longer valuable, are a direct result of the application of standard principles of decision theory to a generalized model of optimization. This connection obviates the need for carefully crafted heuristics and makes powerful decision-theoretic tools available for solving optimization problems.

VIII. Function Class Specification

Setting up an optimization problem so that the UFO can be applied requires prior knowledge of the function class of interest and the ability to codify this knowledge. Of all the required specifications in the GOM, this is likely to be the most difficult. Sample noise and utility are often easily obtained, but the function class definition generally contains some of the information that is desired but not directly available in optimization.

The function class declaration can take many forms, some more general and interesting than others, but NFL dictates that *some* function class must be chosen by the practitioner, even when presented with a completely unknown function. This fact is hidden by common practice in optimization. Practitioners often have a toolbox of existing algorithms at their disposal, from which they select one to apply to a given problem. Such a toolbox may contain PSOs, GAs, EDAs, etc. Without knowledge of the function class, each must be applied with various parameter settings before one may be selected that is at least good enough for the given function.

The function class specification requirement imposed by the UFO initially appears to be even more onerous: if a practitioner has no prior knowledge about the function, how can he appropriately declare the function's class? The problem of function class specification is, in fact, the same as algorithm selection. Lacking any prior knowledge of the function, a practitioner will try various function classes in UFO to find one that works well. Once it is discovered that a particular problem performs well with a given function class, the practitioner gains specific information about the problem, namely that it belongs to a specific function class. No such information is gained when selecting opaque algorithms which do not explicitly indicate their intended function class. Furthermore, because a Bayesian framework is used to perform optimization, the various classes at a practitioner's disposal may be compared, contrasted, and even combined using principled statistical methods [22], [23].

IX. Conclusions and Future Work

The UFO is purely declarative, requiring the following:

- The function class (ρ_f),
- The nature of sampling noise ($\rho_{\mathbf{Y}|f,\mathbf{X}}$),
- The goal of optimization ($\rho_{\mathbf{x}^\star|f}$),
- The value of results ($u_{f_t,\mathbf{x}_t}$), and
- The cost of samples ($c_{\mathbf{x}_i^?}$).

With the possible exception of the function class, each of these is typically available and can be specified in the GOM in its natural form. The class specification, while requiring a different and non-traditional approach to optimization, has many benefits and is not as difficult as it may seem. Simple yet highly flexible function class definitions exist and will be described in future work.

One benefit not directly explored in this work is the fact that the SGOM admits the natural expression of uncertainty in function output due to real nondeterminism (noise), subjective uncertainty, or a combination of the two. Uncertainty is expressed in its native language: as the distribution $\rho_{\mathbf{Y}|f,\mathbf{X}}$. The GOM also admits a natural expression of functions that change over time, similarly expressed as the distribution $\rho_{f_t|f_{t-1}}$ shown in Figure 2

but assumed to be fixed and non-varying in this work. This expressive flexibility inherent in the GOM is fairly unique, since existing methods are frequently designed for the static, deterministic case; noise and dynamics are left to future research. That the addition of measurement noise in the chemistry example was so natural as to not merit specific mention is significant: it was simply folded naturally into the likelihood and then forgotten.

The iterative nature of the GOM places it in company with other Evolutionary algorithms based on probabilistic models. It bears strong resemblance to EDAs but uses expected utility rather than a random variable to induce exploration. It also may be related to Particle Swarm Optimization in that PSO has been shown to have surprisingly deep ties to Bayesian reasoning [10].

The GOM has been labeled as a model of optimization, but in reality it is a more general model of targeted search: the distribution $\rho_{\mathbf{x}^*|f}$ may be defined in arbitrary ways, allowing for the expression of multiple simultaneous target locations, none of which is required to be an optimum of the function. It can, in fact, direct search to *any region of interest* and is therefore simply a specification of practitioner intent. This flexibility lends credibility to the model because optimization is fundamentally a search problem with a very simple and specific definition of success.

The importance of the GOM and its associated UFO algorithm is best understood from the perspective of No Free Lunch: any optimizer that is more successful than random search must so be on a well-delimited function class [15], [26]. Discovery of that class is an implicit goal of most optimization research, generally approached using empirical approaches involving benchmarks [27]. The GOM changes the process of discovery into one of specification; because the function class is known to exist, it makes more sense to incorporate its specification directly and transparently into the algorithm design process. As a result, it becomes possible to specify more of the information that is available to a practitioner but that is often difficult to incorporate: utilities and costs.

This work has introduced an idea that is both interesting and rich enough to admit only a brief and rather dense introduction: evolutionary optimization in terms of Bayesian inference. The idea has been explored in two steps. First, the Graphical Optimization Model was introduced. This model describes the optimization process as inference, using samples to obtain useful information about the location of the optimum. Second, the Utile Function Optimizer algorithm was developed to address the source of those samples, a rational decision process based on available information and clear and explicit declaration of practitioner intent. The model and corresponding algorithm together make a powerful way of thinking about and working with optimization problems, obtaining sophisticated behavior from simple distributions and rules.

References

[1] F. V. Jensen, *Bayesian Networks and Decision Graphs*. New York: Springer Verlag, 2001.

[2] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Englewood Cliffs, New Jersey: Prentice Hall, 2003.

[3] B. E. Stuckman and E. E. Easom, "A comparison of bayesian/sampling global optimization techniques," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 5, pp. 1024–1032, 1992.

[4] A. Törn and A. Žilinskas, *Global Optimization*. Berlin: Springer, 1989.

[5] J. Mockus, *Bayesian Approach to Global Optimization*. New York: Kluwer Academic Publishers, 1989.

[6] J. Mockus, W. Eddy, A. Mockus, L. Mockus, and G. Reklaitis, *Bayesian Heuristic Approach to Discrete and Global Optimization*. Dordrecht, Netherlands: Kluwer Academic Publishers, 1996.

[7] D. J. MacKay, "Introduction to gaussian processes," in *Neural Networks and Machine Learning, vol. 168 of NATO Asi Series. Series F, Computer and Systems Sciences*, C. M. Bishop, Ed. Springer Verlag, 1998.

[8] D. G. Krige, "A statistical approach to some basic mine valuation problems on the witwatersrand," *Journal of the Chem., Metal. and Mining Soc. of South Africa*, vol. 52, no. 6, pp. 119–139, 1951.

[9] C. K. Monson and K. D. Seppi, "The Kalman swarm," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004)*, vol. 1, Seattle, Washington, 2004, pp. 140–150.

[10] ——, "Bayesian optimization models for particle swarms," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2005)*, vol. 1, Washington, D.C., 2005, pp. 193–200.

[11] D. Jones, M. Schonlau, and W. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[12] M. J. Sasena, "Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations," Ph.D. dissertation, University of Michigan, Department of Mchanical Engineering, 2002.

[13] M. Emmerich, A. Giotis, M. Ozdemir, T. Back, and K. Giannakoglou, "Metamodel-assisted evolution strategies," 2002.

[14] P. Larrañaga and J. A. Lozano, Eds., *Estimation of Distribution Algorithms : A New Tool for Evolutionary Computation*. Kluwer Academic Publishers Group, 2001, vol. 2.

[15] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, April 1997.

[16] J. de Freitas, M. Niranjan, A. Gee, and A. Doucet, "Sequential monte carlo methods for optimisation of neural network models," 1998.

[17] C. M. Bishop and M. E. Tippling, "Bayesian regression and classification," *Advances in Learning Theory: Methods, Models and Applications*, vol. 190, pp. 267–285, 2003.

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

[19] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 245–271, 1997.

[20] V. V. Fedorov, *Theory of Optimal Experiments*. Academic Press, 1972.

[21] J. Holland, "Genetic algorithms and the optimal allocation of trials," *SICOMP*, vol. 2, no. 2, pp. 88–105, 1973.

[22] D. J. C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, no. 3, pp. 415–447, 1992.

[23] ——, "Information-based objective functions for active data selection," *Neural Computation*, vol. 4, no. 4, pp. 590–604, 1992.

[24] H. Raiffa and R. Schlaifer, *Applied statistical decision theory*. Cambridge, Mass.: MIT Press, 1961.

[25] D. V. Lindley, *Making Decisions*, 2nd ed. John Wiley and Sons, 1985.

[26] W. G. Macready and D. H. Wolpert, "What makes an optimization problem hard?" *Complexity*, vol. 5, 1996.

[27] D. Whitley and J. P. Watson, "Complexity theory and the no free lunch theorem," in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, E. K. Burke and G. Kendall, Eds. Springer, 2006.