Journal of Global Optimization manuscript No.
(will be inserted by the editor)

A Framework for Optimization under Limited Information

Tansu Alpcan

the date of receipt and acceptance should be inserted later

Abstract In many real world problems, optimisation decisions have to be made with limited information. The decision maker may have no a priori or posteriori data about the often nonconvex objective function except from on a limited number of data points. The scarcity of data may be due to high cost of observation or fast-changing nature of the underlying system. This paper presents a "black-box" optimisation framework that takes into account the information collection, estimation, and optimisation aspects in a holistic and structured manner. Explicitly quantifying the observations at each optimisation step using the entropy measure from information theory, the -often nonconvex- objective function to be optimised is modelled and estimated by adopting a Bayesian approach and using Gaussian processes as a state-of-the-art regression method. The resulting iterative scheme allows the decision maker to address the problem by expressing preferences for each aspect quantitatively and concurrently.

1 Introduction

In many real world problems, optimisation decisions have to be made with limited information. Whether it is a static optimisation or dynamic control problem, obtaining detailed and accurate information about the problem or system can often be a costly and time consuming process. In some cases, acquiring extensive information on system characteristics may be simply infeasible due to prohibitive costs or hard constraints. In others, the observed system may be so nonstationary that by the time the information is obtained, it is already outdated due to system's fast-changing nature. Therefore, the

T. Alpcan
Department of Electical and Electronic Engineering
The University of Melbourne
E-mail: tansu.alpcan@unimelb.edu.au

only option left to the decision-maker is to develop a strategy for collecting information efficiently and choose a model to estimate the "missing portions" of the problem in order to solve it according to a given objective.

To make the discussion more concrete, consider the problem of maximising a Lipschitz continuous *nonconvex* objective function, which is unknown except from its value at only a small number of data points. The decision maker may have no a priori information about the function and start with zero data points. Furthermore, only a limited number of –possibly noisy– observations may be available before making a decision on the maximum value and its location. The function itself, however, remains unknown even after the decision is made. *What is the best strategy to address this problem*?

The decision making framework presented in this paper captures the posed "black-box" optimisation problem by taking into account the information collection, estimation, and optimisation aspects in a holistic and structured manner. Hence, the framework enables the decision maker to solve the problem by expressing preferences for each aspect quantitatively and concurrently. It explicitly incorporates many concepts that have been implicitly considered by other heuristic schemes, and builds upon many results from seemingly disjoint but relevant fields such as information, optimisation, and control theories, and pattern recognition (machine learning). Specifically, it combines concepts from these fields by

- explicitly quantifying the *information* acquired obtained from observations using the entropy measure from information theory,
- modelling and estimating the -nonconvex- function adopting a Bayesian approach and using Gaussian processes as a state-of-the-art regression method,
- using an iterative scheme for observation, learning, and optimisation steps,
- capturing all of these aspects under the umbrella of a multi-objective "meta" optimisation formulation.

Although methods and approaches from machine or statistical learning are heavily utilised in the presented framework, the problem at hand is quite different from many classical machine learning ones. In most classical application domains of machine learning such as data mining, computer vision, or voice recognition, the difficulty is often in handling significant amount of data in contrast to lack of it. However, the framework presented has to operate with limited data and is related more to "active learning" schemes in the literature [7]. The inherent information limitations prevent direct adoption of well-known methods such as Expectation-Maximisation (EM). Consequently, information theory plays here an important role in evaluating scarce data and in developing strategies for obtaining it. Interestingly, data scarcity can also be helpful. It removes the known disadvantages of some methods such as the computational problems faced by Gaussian processes when they are applied to large data sets.

It is worth noting that the class of problems described here are much more frequently encountered in practice than it may first seem. For example, the

class of black-box methods known as "kriging" [13] have been applied to such problems in geology and mining as well as to hydrology since mid-1960s. In addition, the solution framework proposed is applicable to a wide variety of fields due to its fundamental nature. One example is decentralised resource allocation decisions in networked and complex systems, e.g. wired and wireless networks, where parameters change quickly and global information on network characteristics are not available at the local decision-making nodes. Another example is security-related decisions where opponents spend a conscious effort to hide their actions. A related area is security and information technology risk management in large-scale organisations, where acquiring information on individual subsystems and processes can be very costly. Yet another example application is in biological systems where individual organisms or subsystems operate autonomously (even if they are part of a larger system) under limited local information.

2 Problem Analysis

A concrete definition of the motivating problem mentioned in the introduction section is helpful for describing the multiple aspects of the limited information decision making framework. Let $\mathcal{X} \subseteq \Psi \subset \mathbb{R}^d$ be a non-empty, convex, and compact subset of the original problem domain Ψ of d dimensions. The original domain Ψ does not have to be convex, compact, or even fully known. However, adopting a "divide and conquer" approach, it is possible to cover the original domain with a sequence of subsets $\mathcal{X}_i, \quad i = 1, \ldots, N$. Therefore, focusing on $\mathcal{X}$ provides a reasonable starting point.

Define next the original objective function to be maximised

$$f_o : \mathcal{X} \to \mathbb{R},$$

which is unknown except from on a finite number of points observed. As a special but broad case, let f_o belong to the L^p space, $1 \leq p < \infty$. Then, given $\epsilon > 0$, there is a continuous function f such that $||f_o - f|| < \epsilon$. It immediately follows from compactness of $\mathcal{X}$ that f is bounded and assumes its maximum and minimum [29]. Based on this assumption, we will focus on maximisation of the continuous real valued function f on the compact domain $\mathcal{X}$, which approximates f_o.

One of the main distinguishing characteristics of this problem is the limitations on set of observations

$$\Omega_n := \{x_1, \ldots, x_n \, : x_i \in \mathcal{X} \; \forall i, \; n \geq 1\},$$

due to cost of obtaining information or non-stationarity of the underlying system. In many cases these observations may also be noisy. Accordingly, a basic search problem is defined as follows:

Problem 1 (*Basic Search Problem*) *Consider a continuous objective function $f : \mathcal{X} \to \mathbb{R}$ on the d-dimensional non-empty, convex, and compact set*

$\mathcal{X} \subset \mathbb{R}^d$. The function is and will be unknown except from on a finite number of observed data points, Ω_M. What is the best search strategy

$$\Omega_N := \{x_1, \ldots, x_N \, : x_i \in \mathcal{X} \; \forall i, \; N \geq 1\}$$

to find $x^* = \arg\max_x f(x)$ such that $x^* \in \Omega_N \cup \Omega_M$, i.e. to maximise $f(x)$ using a set of limited observations?

The number of new observations, $N \geq 1$, in Problem 1 may be imposed by the nature of the specific application domain. Given a time window of a nonstationary problem, there may be opportunity for making only a certain number of new observations, N. Alternatively, the problem may be stationary but imposes c_o as a constant observation cost on obtaining each data point and the scalar C as the current "exploration budget". Then, N is determined by $\lfloor C/c_o \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function. However, for some problems the observation cost may not be uniform, and is captured by an observation cost function $c_o(x) : \mathcal{X} \to \mathbb{R}$. Then, the exploration budget constraint is $\sum_{x \in \Omega_n} c_o(x) \leq C$ and has to be explicitly taken into account when solving the problem. In certain cases, even this cost may be unknown except that it is obtained iteratively based on the distance from the previous observation, e.g. $c_o(x_n, x_{n-1})$. Hence, a location-based step-by-step search scheme has to be considered.

A special case of Problem 1 is when the new observations are done one by one, i.e. $N = 1$, corresponding to an iterative approach. Myopic or greedy optimisation techniques can be utilised to address this simplified variant. In the other direction, it is possible to generalise the problem to explicitly capture the trade-off between search cost and potential benefits. Subsequently, various optimisation techniques can be deployed.

Focusing on the maximisation of $f(x)$ in Problem 1, the simplest -both conceptually and computationally- strategy to solve it is random search on the domain $\mathcal{X}$. As such no attempt is made to "learn" the properties of the function f. Unless, f is "algorithmically random" [18], which is rarely the case, this strategy wastes the information collected on f. A slightly more complicated and very popular set of strategies combine random search with simple modelling of the function through gradient methods. In this case, the collected information is used to model f rudimentarily using derived gradients to "define slopes" in a heuristic manner. Then, these slopes of f are explored step-by-step in the upwards direction to find a local maximum, after which the search algorithm randomly jumps to another location. It is also possible to randomise the gradient climbing scheme for additional flexibility [30].

The framework presented in this paper takes one further step and **explicitly** models the entire objective function f on the set $\mathcal{X}$ using the information collected instead of heuristically describing only the slopes. The function $\hat{f}$, which models, approximates, and estimates f, belongs to a certain class functions such that $\hat{f} \in \mathcal{F}$. The selection and properties of this class is based on "a priori" information available and can be interpreted as the "world view"

of the decision maker. These properties can often be expressed using meta-parameters which are then updated based on the observations through a separate optimisation process. A slower time-scale process can be used for model selection if processing capabilities permit a multi-model approach. It is worth noting that the "no free lunch" theorem still applies [44].

This model-based search process, which lies at the center of the framework, is fundamentally a manifestation of the Bayesian approach [22]. It first imposes explicit and a priori modelling assumptions by choosing $\hat{f}$ from a certain class of functions, $\mathcal{F}$, and then infers (learns, updates) $\hat{f}$ in a structured manner as more information becomes available through observations. Acquiring new information (search or exploration) is an important objective of the framework since each new piece of information is used to ensure that $\hat{f}$ better approximates the original objective function f within the context of the optimisation problem. Otherwise, there is the danger that the solution of $\hat{f}$ differs significantly from that of the original function f.

From a computational point of view, the decision making framework with limited information lies at one end of the computation versus observation spectrum, while random search is at the opposite end. The framework tries to utilise each piece of information to the maximum possible extent almost regardless of the computational cost. The underlying assumption here is: **observation is very costly whereas computation is rather cheap**. This assumption is not only valid for a wide variety of problems from different fields ranging from networking and security to economics and risk management, but also inspired from biological systems. In many biological organisms, especially complex ones, operating close to this end of the computation-observation spectrum is more advantageous than doing random search.

When doing random search on the domain $\mathcal{X}$, at each stage i.e. given the previous observations, each remaining candidate data point provides equivalent amount of information. However, this is not the case when doing model-based search. Depending on the model adopted and previous information collected, different unexplored points provide different amount of information. This information can be exactly quantified using the definition of entropy and information from the field of ***Shannon* information theory**. A related issue is the reliability and possibly noisy nature of observations, which will be discussed in further detail in the next section.

An extension of Problem 1 that captures the aspects discussed above is defined next.

Problem 2 (*Model-based Search Problem*) *Let $f : \mathcal{X} \to \mathbb{R}$ be a continuous objective function on the d-dimensional non-empty compact set $\mathcal{X} \subset \mathbb{R}^d$, which is unknown except from on a finite number of observed data points, Ω_M. Further let $\hat{f}(x)$ be an estimate of the objective function obtained using an a priori model and observed data. What is the best search strategy $\Omega_N := \{x_1, \ldots, x_N : x_i \in \mathcal{X}\ \forall i,\ N \geq 1\}$ for solving the multi-objective problem with the following components?*

- *Objective 1:* $\max f(x)$ *given* $\hat{f}(x)$

– *Objective 2:* $\max \mathcal{I}(\hat{f}, \Omega_n)$

Here, the scalar quantity $\mathcal{I}$ is the aggregate information obtained from the set of observations Ω_N within the model represented by $\hat{f}$.

Since the original goal f is unknown, the *Objective 1* in Problem 1 has to be formulated as

$$\max_{\tilde{x} \in \mathcal{X}} \hat{f}(\tilde{x}),$$

where $\hat{f}$ is the best estimate obtained through GP regression (5) using the current data set $\mathcal{D}$ or state. Observation errors are modelled through additive Gaussian noise with variance σ as a first approximation.

The second objective in Problem 1 is to maximise the amount of information obtained with each new observation $\tilde{x}$, or

$$\max_{\tilde{x} \in \mathcal{X}} F_2(\tilde{x}) = \mathcal{I}(\tilde{x}, \hat{f}) = \sum_{x \in \Theta} \ln |C_q(x, \tilde{x})| dx,$$

given the best estimate of the original function, $\hat{f}$. This objective quantifies the exploration (search) aspect of the problem and will be discussed in Section 3.1 in detail.

A third objective is to minimise the difference between the estimated function $\hat{f}$ and observations f, which can only be computed at the observation points. For example, a quadratic risk or loss estimate function

$$R = \sum_{x \in \mathcal{D}} \left(\hat{f}(x) - f(x) \right)^2$$

can be chosen to quantify the difference. Note that, while the previous two objectives can be addressed by search policies, this one is a modelling problem. Fortunately, using GP regression for modelling automatically addresses this issue [Chap. 6.2] [27].

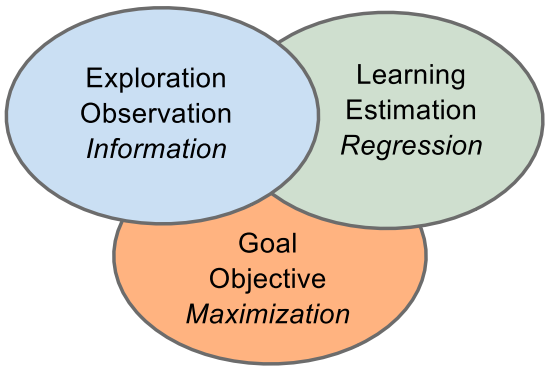

Fig. 1 The three fundamental aspects of decision making with limited information.

The three fundamental aspects of decision making with limited information are depicted in Figure 1. The *Objective 1* in Problem 2 defines the goal while *Objective 2* captures the exploration aspect. The learning dimension is handled by GP estimation of $\hat{f}$. The trade-offs resulting from these conflicting objectives are listed in Table 1. The first one, exploration versus exploitation, puts exploration or obtaining more observations against exploitation, i.e. *Objective 1*. Observation versus computation captures the trade-off between building sophisticated models using the available information to the fullest extend and making more observations. Robustness versus optimisation puts risk avoidance by means of widely applicable but suboptimal approaches against model-based optimisation.

Table 1 Fundamental Trade-offs

Exploration		Exploitation
Observation	versus	Computation
Learning		Model-based Optimisation

3 Model

The model adopted to build a framework that addresses the problem defined consists of three main parts: observation, update of GP for regression, and optimisation to determine next action. These three steps, shown in Figure 2 are taken iteratively to achieve the objectives in Problem 2.

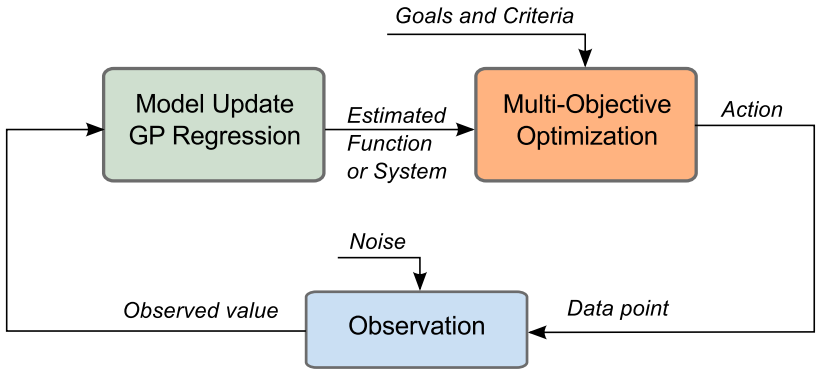

Fig. 2 The main parts of the underlying model of the decision making framework.

Observations, given that they are a scarce resource in the class of problems considered, play an important role in the model. Uncertainties in the observed

quantities can be modelled as additive noise. Likewise, properties (variance or bias) of additive noise can be used to model the reliability of (and bias in) the data points observed. **Gaussian Processes (GP)** provide a straightforward mathematical structure for incorporating these aspects to the model under some simplifying assumptions.

The set of observations collected provide the (supervised) training data for GP regression in order to estimate the characteristics of the function or system at hand. This process relies on the GP methods described in Subsection 3.2. Thus, at each iteration an up-to-date description of the function or system is obtained based on the latest observations. Specifically, $\hat{f}$ provides an estimate of the original function f.[1] Assuming an additive Gaussian noise model, the noise variance σ can be used to model uncertainties, e.g. older and noisy data resulting in higher σ values.

The final and most important part of the model provides a basis for determining the next action after an optimisation process that takes into account all three objectives in Problem 2. The information aspect of these objectives will be discussed in the next Subsection 3.1. An important issue here is the fact that there are infinitely many candidate points in this optimisation process, but in practice only a finite number of them can be evaluated.

3.1 Quantifying Information in Observations

Each observation made provides a data point for identifying and estimating the objective function f by constructing $\hat{f}$. Most works in the learning literature consider the "training" data used in regression available (all at once or sequentially) and do not discuss the possibility of the decision maker influencing or even optimising the data collection process. The *active learning* problem here requires, however, exactly addressing the question of "how to quantify information obtained and optimise the observation process?". Following the approach discussed in [20,22], the framework here provides a precise answer to this question.

Making any decision on the next (set of) observations in a principled manner necessitates first *measuring the information obtained from each observation within the adopted model*. It is important to note that the information measure here is dependent on the chosen model. For example, the same observation provides a different amount of information to a random search model than a GP one.

Shannon information theory readily provides the necessary mathematical framework for measuring the information content of a variable. Let p be a probability distribution over the set of possible values of a discrete random variable A. The **entropy** of the random variable is given by $H(A) = \sum_i p_i \log_2(1/p_i)$, which quantifies the amount of uncertainty. Then, the information obtained

[1] See [27, Chap 7.2] for a discussion on asymptotic analysis of GP regression. It should not be noted, however, that asymptotic properties are of little relevance to the problem at hand.

from an observation on the variable, i.e. reduction in uncertainty, can be quantified simply by taking the difference of its initial and final entropy,

$$\mathcal{I} = H_0 - H_1.$$

It is important here to avoid the common conceptual pitfall of equating entropy to information itself as it is sometimes done in communication theory literature.[2] Within this framework, (Shannon) *information is defined as a measure of the decrease of uncertainty after (each) observation (within a given model).* This can be best explained with the following simple example.

3.1.1 Example: Bisection

Choose a number between 1 and 64 randomly with uniform probability (prior). What is the best searching strategy for finding this number? Let the random variable A represent this number. In the beginning the entropy of A is

$$H_0(A) = \sum_{i=1}^{64} \frac{1}{64} \log_2 \left(\frac{1}{64} \right) = 6 \text{ (bits)}.$$

The information maximisation problem is defined as

$$\max \mathcal{I} = \max H_0 - H_1 = \min H_1,$$

since H_0, the entropy before the action (obtaining information) is constant. The entropy H_1 is the one after information is obtained, and hence is directly affected by the specific action chosen. Now, define the action as setting a threshold $1 < t < 64$ to check whether the chosen number is less or higher than this threshold t. To simplify the analysis, consider a continuous version of the problem by defining p as the probability of the chosen number being less than the threshold. Thus, in this uniform prior case, the problem simplifies to

$$\min_p H_1 = \min_p \; p \log(p) + (1-p) \log(1-p),$$

which has the derivative

$$\frac{dH_1}{dp} = \log(p) - \log(1-p).$$

Clearly, the threshold $p^* = 0.5$ is the global minimum, which roughly corresponds to $t = 32$ (ignoring quantisation and boundary effects). Thus, bisection from the middle is the optimal search strategy for the uniform prior. In this example, the number can be found in the worst-case in 6 steps, each providing one bit of information. Non-uniform probabilities (priors) can be handled in a similar way.

[2] Since this issue is not of great importance for the class of problems considered in communication theory, it is often ignored. However, the difference is of conceptual importance in this problem. See `http://www.ccrnp.ncifcrf.gov/~toms/information.is.not.uncertainty.html` for a detailed discussion.

If this search process (bisection) is repeatedly applied without any feedback, then it results in the optimal quantisation of the search space both in the uniform case above and for the non-uniform probabilities. If feedback is available, i.e. one learns after each bisection whether the number is larger or less than the boundary, then this is as shown the best search strategy.

3.2 Regression and Gaussian Processes (GP)

Problem 2 presented in the previous section involves inferring or learning the function f using the set of observed data points. This is known as the *regression* problem in machine learning and is a supervised learning method since the observed data constitutes at the same time the learning data set. This learning process involves selection of a "model", where the learned function $\hat{f}$ is, for example, expressed in terms of a set of parameters and specific basis functions, and at the same time minimisation of an error measure between the functions f and $\hat{f}$ on the learning data set. Gaussian processes (GP) provide a non-parametric alternative to this but follow in spirit the same idea.

The main goal of regression involves a trade-off. On the one hand, it tries to minimise the *observed* error between f and $\hat{f}$. On the other, it tries to infer the "real" shape of f and make good estimations using $\hat{f}$ even at unobserved points. If the former is overly emphasised, then one ends up with "over fitting", which means $\hat{f}$ follows f closely at observed points but has weak predictive value at unobserved ones. This delicate balance is usually achieved by balancing the prior "beliefs" on the nature of the function, captured by the model (basis functions), and fitting the model to the observed data.

This paper focuses on Gaussian Process [27] as the chosen regression method within the framework developed without loss of any generality. There are multiple reasons behind this preference. Firstly, GP provides an elegant mathematical method for easily combining many aspects of the framework. Secondly, being a non-parametric method GP eliminates any discussion on model degree. Thirdly, it is easy to implement and understand as it is based on well-known Gaussian probability concepts. Fourthly, noise in observations is immediately taken into account if it is modelled as Gaussian. Finally, one of the main drawbacks of GP namely being computational heavy, does not really apply to the problem at hand since the amount of data available is already very limited.

It is not possible to present here a comprehensive treatment of GP. Therefore, a very rudimentary overview is provided next within the context of the decision making problem. Consider a set of M data points

$$\mathcal{D} = \{x_1, \ldots, x_M\},$$

where each $x_i \in \mathcal{X}$ is a $d-$dimensional vector, and the corresponding vector of scalar values is $f(x_i),\ i = 1, \ldots, M$. Assume that the observations are distorted by a zero-mean Gaussian noise, n with variance $\sigma \sim \mathcal{N}(0, \sigma)$. Then, the resulting observations is a vector of Gaussian $y = f(x) + n \sim \mathcal{N}(f(x), \sigma)$.

A GP is formally defined as a collection of random variables, any finite number of which have a joint Gaussian distribution. It is completely specified by its mean function $m(x)$ and covariance function $C(x, \tilde{x})$, where

$$m(x) = E[\hat{f}(x)] \text{ and } C(x, \tilde{x}) = E[(\hat{f}(x) - m(x))(\hat{f}(\tilde{x}) - m(\tilde{x}))],\ \forall x, \tilde{x} \in \mathcal{D}.$$

Let us for simplicity choose $m(x) = 0$. Then, the GP is characterised entirely by its covariance function $C(x, \tilde{x})$. Since the noise in observation vector y is also Gaussian, the covariance function can be defined as the sum of a *kernel function* $Q(x, \tilde{x})$ and the diagonal noise variance

$$C(x, \tilde{x}) = Q(x, \tilde{x}) + \sigma I,\ \forall\, x, \tilde{x} \in \mathcal{D}, \tag{1}$$

where I is the identity matrix. While it is possible to choose here any (positive definite) kernel $Q(\cdot, \cdot)$, one classical choice is

$$Q(x, \tilde{x}) = \exp\left[-\frac{1}{2}\left\|x - \tilde{x}\right\|^2\right]. \tag{2}$$

Note that GP makes use of the well-known *kernel trick* here by representing an infinite dimensional continuous function using a (finite) set of continuous basis functions and associated vector of real parameters in accordance with the *representer theorem* [32].

The (noisy)[3] training set $(\mathcal{D}, y)$ is used to define the corresponding GP, $\mathcal{GP}(0, C(\mathcal{D}))$, through the $M \times M$ covariance function $C(\mathcal{D}) = Q + \sigma I$, where the conditional Gaussian distribution of any point outside the training set, $\bar{y} \in \mathcal{X}, \bar{y} \notin \mathcal{D}$, given the training data $(\mathcal{D}, t)$ can be computed as follows. Define the vector

$$k(\bar{x}) = [Q(x_1, \bar{x}), \ldots Q(x_M, \bar{x})] \tag{3}$$

and scalar

$$\kappa = Q(\bar{x}, \bar{x}) + \sigma. \tag{4}$$

Then, the conditional distribution $p(\bar{y}|y)$ that characterises the $\mathcal{GP}(0, C)$ is a Gaussian $\mathcal{N}(\hat{f}, v)$ with mean $\hat{f}$ and variance v,

$$\hat{f}(\bar{x}) = k^T C^{-1} y \text{ and } v(\bar{x}) = \kappa - k^T C^{-1} k. \tag{5}$$

This is a key result that defines GP regression as the mean function $\hat{f}(x)$ of the Gaussian distribution and provides a prediction of the objective function $f(x)$. At the same time, it belongs to the well-defined class $\hat{f} \in \mathcal{F}$, which is the set of all possible sample functions of the GP

$$\mathcal{F} := \{\hat{f}(x) : \mathcal{X} \to \mathbb{R} \text{ such that } \hat{f} \in \mathcal{GP}(0, C(\mathcal{D})),\ \forall \mathcal{D},\ C\},$$

where $C(\mathcal{D})$ is defined in (1) and $\mathcal{GP}$ through (3), (4), and (5), above. Furthermore, the variance function $v(x)$ can be used to measure the uncertainty level of the predictions provided by $\hat{f}$, which will be discussed in the next subsection.

[3] The special case of perfect observation without noise is handled the same way as long as the kernel function $Q(\cdot, \cdot)$ is positive definite

3.3 Quantifying Information using Gaussian Processes

The information measurement and GP approaches in the previous subsections are now directly combined. Let the zero-mean multivariate Gaussian (normal) probability distribution at a point $x \in \mathcal{X}$ be denoted as

$$p(x) = \frac{1}{\sqrt{2\pi|C_p(x)|}} \exp\left(-\frac{1}{2}[x-m]^T|C_p(x)|^{-1}[x-m]\right), \tag{6}$$

where $|\cdot|$ denotes determinant if the argument is a matrix and absolute value of it is scalar, m is the mean (vector) as defined in (5), and $C_p(x)$ is the covariance matrix given by

$$C_p(x) = \begin{bmatrix} C(\mathcal{D}) & k(x)^T \\ k(x) & \kappa \end{bmatrix}. \tag{7}$$

Here, the vector $k(x)$ is defined in (3) and κ in (4), respectively. The matrix $C(\mathcal{D})$ is the covariance matrix based on the training data $\mathcal{D}$ as defined in (1).

The entropy of the multivariate Gaussian distribution (6) is [2]

$$H(x) = \frac{d}{2} + \frac{d}{2}\ln(2\pi) + \frac{1}{2}\ln|C_p(x)|,$$

where d is the dimension. Note that, this is the entropy of the GP estimate at the point x based on the available data $\mathcal{D}$.

Now, in order to simplify the analysis, sample the region $\mathcal{X}$ to obtain a set of solution candidates

$$\Theta := \{x_1, \ldots, x_T : x_i \in \mathcal{X},\, x_i \notin \mathcal{D},\, \forall i\},$$

which constitutes a *discrete search space*. Due to the nature of the problem, it is not possible evaluate infinitely many points on $\mathcal{X}$ anyway. The sampling of the search space is discussed in detail in the next section.

The aggregate entropy of the function on the search space Θ is given by

$$H^{agg} := \frac{1}{2}\sum_{x\in\Theta}\ln|C_p(x)| + \frac{d}{2}\ln(2\pi e). \tag{8}$$

The problem of choosing the optimal new data point $\hat{x} \in \Theta$ such that the information obtained from it within the GP regression model is maximised can be formulated in a way similar to the one in the bisection example by computing

$$\hat{x} = \arg\max_{\tilde{x}} \mathcal{I} = \arg\max_{\tilde{x}} [H_0 - H_1(\tilde{x})] \tag{9}$$

While the uncertainty (entropy) before observation, H_0 is fixed, the uncertainty after the observation is a function of the observation $\tilde{x}$, and is given by

$$H_1 = \frac{1}{2}\sum_{x\in\Theta\setminus\tilde{x}}\ln|C_q(x,\tilde{x})| + \frac{d}{2}\ln(2\pi e),$$

where the covariance matrix $C_q(x,\tilde{x})$ is defined as

$$C_q(x,\tilde{x}) = \begin{bmatrix} C(\mathcal{D}) & k^T(\tilde{x}) & k^T(x) \\ k(\tilde{x}) & \tilde{\kappa} & Q(x,\tilde{x}) \\ k(x) & Q(x,\tilde{x}) & \kappa \end{bmatrix}, \tag{10}$$

and $\tilde{\kappa} = Q(\tilde{x},\tilde{x})+\sigma$. Here, $C(\mathcal{D})$ is a $M\times M$ matrix and C_q is a $(M+2)\times(M+2)$ one, whereas κ and $Q(x,\tilde{x})$ are scalars and k is a $M\times 1$ vector.

Note that [25]

$$|C_q(x,\tilde{x})| = |C_p(\tilde{x})|\,|\kappa(x) - k(x)C_p^{-1}(\tilde{x})k^T(x)|, \tag{11}$$

and

$$|C_p(\tilde{x})| = |C(\mathcal{D})|\,|\kappa(\tilde{x}) - k(\tilde{x})C^{-1}(\mathcal{D})k^T(\tilde{x})|, \tag{12}$$

where the matrix C_p is given by (7). The first term in (12 above, $|C(\mathcal{D})|$ is fixed and the second one,

$$v(\tilde{x}) = |\kappa(\tilde{x}) - k(\tilde{x})C^{-1}(\mathcal{D})k^T(\tilde{x})| \tag{13}$$

is the GP variance in (5).

Define next

$$\tilde{v}(x, C_p(\tilde{x})) := |\kappa(x) - k(x)C_p^{-1}(\tilde{x})k^T(x)|. \tag{14}$$

Then, the uncertainty after observation is

$$H_1 = \frac{1}{2}\sum_{x\in\Theta\setminus\tilde{x}} \ln v(\tilde{x}) + \frac{1}{2}\sum_{x\in\Theta\setminus\tilde{x}} \ln \tilde{v}(x, C_p(\tilde{x})) + \frac{1}{2}\sum_{x\in\Theta\setminus\tilde{x}} \ln|C(\mathcal{D})| + \frac{d}{2}\ln(2\pi e),$$

where the last two terms are constant. The results of the discussion above are summarised in the following proposition.

Proposition 1 *Let Θ denote a set of of candidates (search space) sampled from the domain $\mathcal{X}$ of an unknown function $f(x)$, which is estimated using a Gaussian Process regression model with covariance matrix $C(\mathcal{D})$ derived from a given data set $\mathcal{D}$ and a positive definite kernel. The observation $\hat{x}\in\Theta$, provides the maximum information within the model of a Gaussian Process regression, if chosen in such a way that*

$$\hat{x} = \arg\min_{\tilde{x}\in\Theta}\left(\mathrm{card}(\Theta\setminus\tilde{\mathrm{x}})\cdot\ln \mathrm{v}(\tilde{\mathrm{x}}) + \sum_{\mathrm{x}\in\Theta\setminus\tilde{\mathrm{x}}}\ln\tilde{\mathrm{v}}(\mathrm{x},\mathrm{C_p}(\tilde{\mathrm{x}}))\right), \tag{15}$$

where $=\mathrm{card}(\cdot)$ denotes the cardinality of a set, $v(\tilde{x})$ is defined in (13) and $\tilde{v}(x,C_p(\tilde{x}))$ in (14), respectively

Although the optimisation problem in Proposition 1 is not analytically tractable (see e.g. [17] for an interesting discussion), a rudimentary analysis reveals interesting insights. Let the Gaussian process kernel to be stationary [27], i.e. a function of the distance to the known data points as in the case of (2) . Then, both $v(\tilde{x})$ in (13) and $\tilde{v}(x, C_p(\tilde{x}))$ in (14) are positive and less than one. Furthermore, the variance (13) in the first term of (15) approaches one, its maximum value, when $\tilde{x}$ is farthest away from the data set $\mathcal{D}$, which immediately follows its definition. At the same time, however, the second variance decreases toward zero. To see this, note that if $\tilde{x}$ is chosen "away" from $\mathcal{D}$, then there are more points $x \in \Theta$ closer to the aggregate data set, and hence with less variance. Due to the logarithms in (15), the second term dominates the first one, as the first one approaches zero, especially when the data is more or less uniformly distributed over the search space.

Based on these observations, it is possible to devise a simple heuristic algorithm to *approximately* solve (15). If $\tilde{x}$ is chosen such that the variance in (13) is maximised, then this leads to a large (possibly largest) reduction in the second term of (15), and hence provides a rough approximate solution to the problem in Proposition 1. If there are multiple candidates with maximum variance, then a secondary goal should be to uniform coverage of the search space.

Proposition 2 *Given a Gaussian Process with a stationary covariance matrix $C(\mathcal{D})$ and a collection of candidate points Θ, an approximate solution to the maximum information data collection problem defined in Proposition 1 is to choose the sample points $\tilde{x}$ in such a way that they have the maximum variance and provide a uniform coverage of the search space Θ.*

The algorithm in Proposition 2 corresponds to the widely-known heuristics such as "maximum entropy" or "minimum variance" methods [34] and a variant has been discussed in [20]. It is also quite intuitive: the maximum amount of information is obtained if the search is conducted away from known data points in empty parts of the search space.

3.4 Sampling Search Space

When making a decision on the next action through multi-objective optimisation, there are infinitely many candidate points. A pragmatic solution to the problem of finding solution candidates is to sample the problem domain $\mathcal{X}$ to obtain the set

$$\Theta := \{x_1, \ldots, x_T : x_i \in \mathcal{X},\, x_i \notin \mathcal{D},\, \forall i\}$$

that does not overlap with known points. In low (one or two) dimensions, this can be easily achieved through grid sampling methods. In higher dimensions, Monte Carlo or Quasi Monte Carlo schemes can be utilised. For large problem domains, the current domain of interest $\mathcal{X}$ can be defined around the last or

most promising observation in such a way that such a sampling is computationally feasible. Likewise, multi-resolution and adaptive sampling schemes can be deployed to increase computational efficiency.

Although such a solution may seem restrictive at first glance, it is in spirit not very different from other schemes such as simulated annealing, which are widely used to address nonconvex optimisation problems. However, a major difference between this and other schemes is the fact that the candidate sampling and evaluation are done here "a priori" due to experimentation being costly while other methods rely on abundance of information.

A natural question that arises is: whether and under what conditions does such a sampling method give satisfactory results. The following result from [39,40] provides an answer to this question in terms of number of samples required.

Theorem 1 *[39,40] Define a multivariate function $f(x)$ on the convex, compact set $\mathcal{X}$, which admits the maximum $x^* = \arg\max_{x\in\mathcal{X}} f(x)$. Based on a set of N random samples $\Theta = \{x_1, \ldots, x_N : x_i \in \mathcal{X}\ \forall i\}$ from the entire set $\mathcal{X}$, let $\hat{x} := \arg\max_{x\in\Theta} f(x)$ be an estimate of the maximum x^*.*

Given an $\varepsilon > 0$ and $\delta > 0$, the minimum number of random samples N which guarantees that

$$Pr\left(Pr[f(x^*) > f(\hat{x})] \leq \varepsilon\right) \geq 1 - \delta,$$

i.e. the probability that 'the probability of the real maximum surpassing the estimated one being less than ε' is larger than $1-\delta$, is

$$N \geq \frac{\ln 1/\delta}{1/(1-\varepsilon)}.$$

Furthermore, this bound is tight if the function f is continuous on $\mathcal{X}$.

It is interesting and important to note that this bound is independent of the sampling distribution used (as long as it covers the whole set $\mathcal{X}$ with non-zero probability), the function f itself, as well as the properties and dimension of the set $\mathcal{X}$.

Thus, Theorem 1 provides a probabilistic upper bound on the results in Propositions 1 and 2. It can also be used as a way to "hedge the bet", in other words, as a way to balance the risk that the model-based search proposed fails to perform adequately due to modelling errors.

4 Search Algorithms

In the classical full information case, the problem $\max_x f(x)$ is a static one. Even though it is usually solved using iterative methods, the objective function remains constant and is completely known. The search problem in Section 2, however, differs from the classical one in terms of information available. In this case the objective function is not known a priori. It is only explored and

estimated using observations made over time. Hence, the optimisation of the objective in the limited information becomes a **dynamic problem**.

The dynamic nature of Problem 1 can be clarified by building a connection with dynamic programming [6,37]. The problem analysed in the previous sections can be formulated as

$$\begin{array}{ll} \max_{x_1,\ldots,x_N} & \hat{F}(s_{N+1}) \\ \text{such that} & s_{n+1} = \mathcal{T}(s_n, x_n) \\ \text{and} & x_n \in \Theta,\ s_n \in \mathcal{S},\ \ n = 1, \ldots, N. \end{array} \tag{16}$$

The (estimated) goal function $\hat{F}$ represents the objectives discussed in Section 2 and takes the states s in a state space S as arguments, $\mathcal{T}$ maps previous state and current action to the next state, and $x_1, \ldots, x_N$ denote the search/optimisation actions over time $n = 1, \ldots, N$. In the context of this paper, the state s simply corresponds to the current data set $\mathcal{D}$, and hence the mapping $\mathcal{T}$ is simply the set union operator, $\mathcal{T}(s_n, x_n) = s_n \cup x_n$. Thus, the Problem 1 corresponds to the most general formulation of a finite-horizon dynamic program. Furthermore, the principle of optimality naturally holds: "the optimal search policy has the property that the remaining search decisions must constitute an optimal policy given the state resulting from the first decision, regardless of the initial data set" [35].

As a result of its rich nature one can devise a variety of algorithms focusing on different aspects of this problem. As discussed in Section 2, the combination of exploration, learning, optimisation, and robustness dimensions creates a vast playing field for algorithm design. The methodology presented here differs from other heuristic approaches due to the explicit quantification and systematic combination of these objectives. As a starting point, three specific (classes of) algorithms are discussed next.

4.1 Greedy Algorithms

While the search Problem 1 is related to dynamic programming as discussed above, it has also fundamental differences. By the very nature of the problem, the objective F in (16) is time-varying and unknown except from on a set of observed data points. This lack of information prevents applying standard backward propagation type direct dynamic programming methods [36] to the problem. However, it is still possible to adopt a successive approximation approach, where an initial estimate of the objective function is constructed. This estimate is then iteratively updated as more data becomes available.

First, using the most common approach to multi-objective optimisation – the **weighted sum method** [23,11], the two objectives defined in Section 2 are combined to obtain a single objective with the respective weights $[w_1,\ w_2]$, $w_1 + w_2 = 1$, $0 \leq w_i \leq 1$, $i = 1, 2$. Hence, a specific weighted sum formulation is obtained:

$$F(x) = w_1 F_1 + w_2 F_2 = w_1 \hat{f}(x) + w_2 \mathcal{I}(x, \hat{f}), \tag{17}$$

Assuming a single data point is chosen from the solution candidates Θ at each step, a specific weighted sum formulation to address Problem 2 is obtained.

Proposition 3 *The solution, $\tilde{x} \in \Theta$, to the optimisation problem*

$$\max_{\tilde{x}\in\Theta} \hat{F}(\tilde{x}) = w_1 \hat{f}(\tilde{x}) + w_2 \mathcal{I}(\tilde{x}, \hat{f}), \tag{18}$$

provides a greedy search iteration for addressing this weighted sum formulation of Problem 2.

As discussed in Subsection 3.1, the information objective in (17) can be approximated by substituting it with GP variance $v(x)$ in (5) to decrease computational load. Hence, an approximation to (17) is:

$$F(x) = w_1 \hat{f}(x) + w_2 v(x), \tag{19}$$

where $v(x)$ is defined in (5). Thus, an approximation to the solution in Proposition 3 is obtained.

Proposition 4 *The solution, $\tilde{x} \in \Theta$, to the optimisation problem*

$$\max_{\tilde{x}\in\Theta} \hat{F}(x) = w_1 \hat{f}(x) + w_2 v(\tilde{x}), \tag{20}$$

where $v(\tilde{x})$ is defined in (5), approximates the greedy search iteration in Proposition 3.

The weighted greedy search scheme described is only meaningful if both objectives are of the same order of magnitude. Therefore, the individual objective terms have to be transformed or "normalised. There are many different approaches to perform such a transformation [23, 11]. The most common one, which coincidentally is known as normalisation, aims to map each objective function to a predefined interval, e.g. [0, 1]. To do this, estimate first an upper F_i^U and lower F_i^L bound on each individual objective $F_i(x)$. Then, the i^{th} normalised objective is

$$F_i^{norm}(x) = \frac{F_i(x) - F_i^L}{F_i^U - F_i^L}.$$

The main issue in normalisation is to determine the appropriate upper and lower bounds, which is a very problem-dependent one. Here, the estimated function $\hat{f}$ on the set Θ as well as the existing observations $\mathcal{D}$, can be utilised to obtain these values. The specific bounds for the respective objectives $F_1^U = \max_{x\in\Theta} \hat{f}(x)$, $F_1^L = \min_{x\in\Theta} \hat{f}(x)$, $F_2^U = \max_{x\in\Theta} \kappa(x)$, and $F_2^L = 0$ provide a suitable starting estimate and can be combined with a prior domain knowledge if necessary.[4] A normalised version of the formulation in (19) is:

$$F^{norm}(x) = \frac{w_1}{\Delta_1}\left(\hat{f}(x) - F_1^L\right) + \frac{w_2}{\Delta_2} v(\tilde{x}), \tag{21}$$

where $\Delta_i = F_i^U - F_i^L$ $i = 1, 2$. Thus, a normalised version of the formulation in Proposition 4 is obtained.

[4] These bounds depend on the specific kernel function used. Here, the specific kernel function in (2) is assumed.

Proposition 5 *The solution, $\tilde{x} \in \Theta$, to the optimisation problem*

$$\max_{\tilde{x}\in\Theta} \hat{F}(x) = \frac{w_1}{\Delta_1}\left(\hat{f}(x) - F_1^L\right) + \frac{w_2}{\Delta_3} v(\tilde{x}), \tag{22}$$

where $\Delta_i = F_i^U - F_i^L$ $i = 1, 2$, provides an approximation to the greedy search iteration for addressing the normalised weighted-sum formulation of Problem 2.

The estimated objective function $\hat{F}_n$ is iteratively updated at each step n after a new observation is made. GP regression model already provides the necessary framework for estimating the function as well as the expected information gain. Hence, successive approximation through iterative updates ensures that all of the available data is fully utilised within the model.

Finally, the next point for observation is selected in a greedy manner by solving $\arg\max_{x\in\Theta} F(x)$. Thus, the algorithm concurrently satisfies - according to the chosen weights- both the objective of exploration and maximisation. Furthermore, if the algorithm is interrupted at any step, the result is the best one obtained. The overall structure of the framework is visualised in Figure 3. An summary of the weighted-sum approach described above is provided by Algorithm 1.

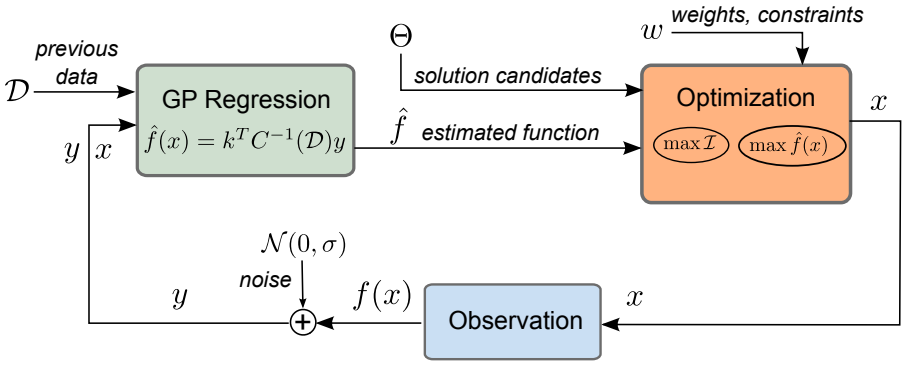

Fig. 3 The decision making framework for static optimisation with limited information.

It is worth noting that the class of algorithms described above is very similar to the famous *Dijkstra's algorithm*. This is maybe not very surprising considering the fact that Dijkstra's algorithm is itself a specific successive approximation or greedy algorithm [36]. However, building a connection can be illustrative.

Let us define a graph $\mathcal{G}$ using the subset of points on the search space Θ as individual nodes. The non-negative edge weights are defined as distances from each subset to the next one, which differ from each other by a single point in Θ, based on the objective function F described above. Then, given $\mathcal{D}$ as a starting point, the search problem becomes finding the shortest path to one of the subsets which contain the point $\arg\max_x f(x)$. At each step, the estimated distance to neighbouring nodes are computed and the best one is chosen for

Algorithm 1 Greedy Algorithm for Optimisation under Limited Information

1: **Input:** Function domain, $\mathcal{X}$, GP meta-parameters, objective weights $[w_1, w_2]$ initial data set $(\mathcal{D}, y)$.
2: Use GP with a Gaussian kernel and specific expected error variances for function $\hat{f}$ estimation.
3: **while** Search budget available, $1 \leq n \leq N_{max}$. **do**
4: Sample domain $\mathcal{X}$ to obtain $\Theta(n)$. In some cases, $\Theta(n) = \Theta \; \forall n$.
5: Estimate $\hat{f}$ based on observed data $(\mathcal{D}, y)$ on $\Theta(n)$ using GP regression.
6: Compute variance, $v(x)$, of $\hat{f}$ (5) on $\Theta(n)$ as an estimate of $\mathcal{I}(\hat{f})$.
7: Choose the point that maximises a normalised and weighted sum of objectives F_i^N as stated in Proposition 5.
8: Update the observed data $(\mathcal{D}, y)$.
9: **end while**

the next observation. From this point of view, the Algorithm 1 corresponds to Dijkstra's algorithm. However, there are couple of caveats. The edge weights from a node to its neighbours are assumed to be completely known in classical Dijkstra's algorithm. Likewise, the algorithm runs for as many steps as necessary to find the correct solution. Both of these assumptions are violated in Problem 2. Hence, a model-based view has to be necessarily adopted and the exploration costs are embedded to edge weights in Algorithm 1.

4.2 Hedging Algorithms

While greedy algorithms discussed above aggressively try to find a solution to Problem 2, they may under-perform significantly if the underlying model parameters do not match the problem at hand. Ultimately, this issue requires adopting an adaptive model selection scheme as an outer loop the framework in Figure 3. However, it is possible to consider a simpler temporary solution by allocating a portion of the search budget available to random exploration as an insurance or a way to counteract modelling errors to some extend. For example, a certain percentage such as %10 of the search budget N can be used to randomly choose points from Θ, especially from unexplored regions. One way to achieve this is to select these points according to a distribution inversely proportional to the $\hat{F}$ suggested by the model. Hence, the bet on the model is hedged by this scheme. The "insurance" thus bought or the minimum expected performance can be quantified by the result in Theorem 1. The Algorithm 2 provides an example of the approach described.

4.3 Meta-optimisation Algorithms

It is possible to change the search objectives dynamically while running the algorithm. For example, the weights w in the greedy Algorithm 2 can be adaptively changed according to some criterion or gradually similar to the approach

Algorithm 2 Hedging Algorithm for Optimisation under Limited Information

1: **Input:** Function domain, $\mathcal{X}$, GP meta-parameters, objective weights $[w_1, w_2]$ initial data set $(\mathcal{D}, y)$.
2: Use GP with a Gaussian kernel and specific expected error variances for function $\hat{f}$ estimation.
3: **while** Search budget available, $1 \leq n \leq N_{max}$. **do**
4: Sample domain $\mathcal{X}$ to obtain $\Theta(n)$. In some cases, $\Theta(n) = \Theta \; \forall n$.
5: Estimate $\hat{f}$ based on observed data $(\mathcal{D}, y)$ on $\Theta(n)$ using GP regression.
6: Compute variance, $v(x)$, of $\hat{f}$ (5) on $\Theta(n)$ as an estimate of $\mathcal{I}(\hat{f})$.
7: **if** Regular step **then**
8: Choose the point that maximises a normalised and weighted sum of objectives F_i^N as stated in Proposition 5.
9: **else if** Hedging step, e.g. once every k steps **then**
10: Next point is chosen randomly according to a distribution inversely proportional to the $\hat{F}$ suggested by the model.
11: **end if**
12: Update the observed data $(\mathcal{D}, y)$.
13: **end while**

in simulated annealing [30]. Alternatively, the criterion may be formally motivated by the multi-objective nature of the problem.

The **bounded objective function** method provides a suitable alternative to the weighted sum formulation above in addressing the multi-objective problem defined. The bounded objective function method minimises the single most important objective, in this case $\max \hat{f}(x)$, while the exploration objective $\max \mathcal{I}(x)$ is converted to form an additional constraint. Such constraints are in a sense similar to QoS ones that naturally exist in many real life problems [24,4,38]. As an advantage, in the bounded objective formulation there is no need for normalisation.

The bounded objective counterpart of the result in Proposition 4 is as follows.

Proposition 6 *The solution, $\tilde{x} \in \Theta$, to the constrained optimisation problem*

$$\max_{\tilde{x} \in \Theta} \hat{f}(x) \tag{23}$$
$$\text{such that } 0 \leq v(\tilde{x}) \leq b,$$

where b is a given (predetermined) scalar bound, provides an approximate best search iteration to a bounded-objective formulation of Problem 2.

The advantage of the bounded objective function method in Algorithm 3 is that it provides a bound on the information collection objective while maximising the estimated function. This leads in practice to an initial emphasis on information collection and correct estimation of the objective function. In that sense, the method is more "classical", i.e. follows the common method of learn first and maximise later. Furthermore, it does not require normalisation,

Algorithm 3 Bounded Objective Algorithm for Optimisation under Limited Information

1: **Input:** Function domain, $\mathcal{X}$, GP meta-parameters, bound b, initial data set $(\mathcal{D}, y)$.
2: Use GP with a Gaussian kernel and specific expected error variances for function $\hat{f}$ estimation.
3: **while** Search budget available, $1 \leq n \leq N_{max}$. **do**
4: Sample domain $\mathcal{X}$ to obtain $\Theta(n)$. In some cases, $\Theta(n) = \Theta \; \forall n$.
5: Estimate $\hat{f}$ based on observed data $(\mathcal{D}, y)$ on $\Theta(n)$ using GP regression.
6: Compute variance, $v(x)$, of $\hat{f}$ (5) on $\Theta(n)$ as an estimate of $\mathcal{I}(\hat{f})$.
7: **if** variance is larger than bound $v(x) > b$ **then**
8: Choose the point with maximum variance.
9: **else**
10: Choose the point that maximises $\hat{f}$ or heavily emphasises this objective.
11: **end if**
12: Update the observed data $(\mathcal{D}, y)$.
13: **end while**

i.e. it is easier to deploy. The method has, however, a significant disadvantage which makes its usage prohibitive. In large-scale or high-dimensional problems, the space to explore to satisfy any bound on information is simply immense. Therefore, one does not have the luxury of identifying the function first to maximise it later as it would take too many samples to do this. In such cases, it makes more sense to deploy the weighted sum method, possibly along with a cooling scheme to modify the weights as part of a cooling scheme to balance depth-first versus breadth-first search.

Until now, it has been (implicitly) assumed that the static optimisation problem at hand is stationary. However, in a variety of problems this is not the case and the function $f(x, t)$ changes with time. The decision making framework allows for modelling such systems in the following way. Let

$$\mathcal{O}(t) = \{f(x_i, t_i) + n(x_i, t_i) : x_i \in \mathcal{D}, t_i \leq t, \forall i\},$$

be the set of noisy or unreliable past observations until time t, where $n(x, t) \sim \mathcal{N}(0, \sigma(t))$ is the zero mean Gaussian "noise" term at time t. Now, the deterioration in the past information due to change in $f(x, t)$ can be captured by increasing the variance of the noise term, $\sigma(t)$, with time. For example, a simple linear dynamic can be defined as

$$\frac{d\sigma(t)}{dt} = \eta,$$

where $\eta > 0$ captures the level of stationarity, e.g. a large η indicates a rapidly changing system and function $f(x, t)$.

5 Numerical Analysis

The approach presented and the algorithms in Section 4 are illustrated next with multiple numerical examples. The goal here is to demonstrate the main ideas discussed rather than establishing competitive advantage of the algorithms over alternatives. It is worth reminding that the main contribution of this paper is establishment of underlying principles for solving optimisation problems with minimum data using active learning. The main assumption is that data is "expensive" while computation is "cheap" which is almost the opposite of the case in many other schemes in the literature.

Example 1

The first numerical example aims to visualise the presented framework and Algorithm 1. Hence, the chosen function is only one dimensional, $f(x) = sin(5x)/x$ on the interval $\mathcal{X} = [0.001, 6]$. The interval is linearly sampled to obtain a grid with a distance of 0.001 between points, i.e. $\Theta = \{x_i \in \mathcal{X} \, \forall i : x_1 = 0.001, x_2 = 0.002, \ldots, x_N = 6\}$. The Gaussian kernel in (2) with variance 0.1 is chosen for estimating both $\hat{f}$ and the weights are $w = [2, 1]$. The initial data consists of a single point, $x = 0.001$.

Figure 4 shows the results based on the normalised weighted-sum method in Proposition 5 after 14 iterations, or 15 samples in total, together with the initial data point. The variance here is $v(x)$ of the estimated function $\hat{f}$ using data points $\mathcal{D}$. The location of the functions peak is estimated correctly.

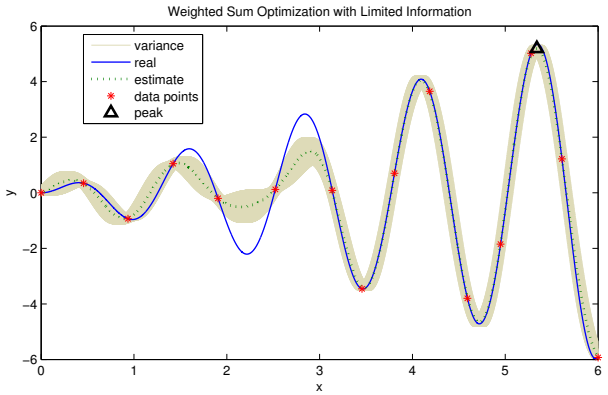

Fig. 4 Optimisation result using the weighted-sum method with 15 data points.

The amount of information obtained during the iterative optimisation is of particular interest. Figure 5 depicts the mean variance v and entropy $\mathcal{I}$ of the estimated function $\hat{f}$ on Θ at each iteration step. In this specific example, the two quantities are very well correlated. Note, however, that this correlation

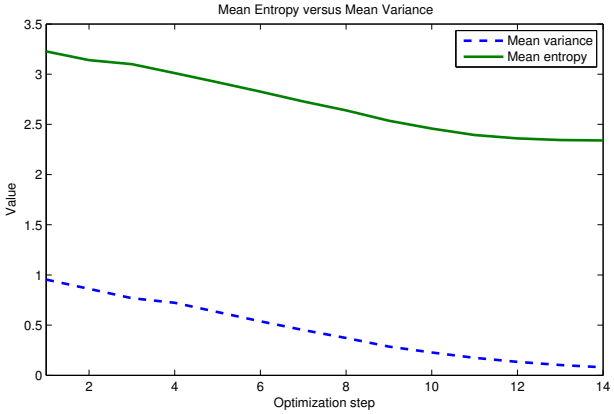

Fig. 5 Mean variance v and entropy $\mathcal{I}$ on Θ at each iteration step.

is a function of the relative weights between information collection and other objectives.

Example 2

The second example is based on the six-hump camel function [10] (see Figure 6) on the domain $\mathcal{X} = [-1, 1] \times [-2, 2]$, which is sampled uniformly with a 0.1 interval. The Gaussian kernel in (2) with variance 0.5 is chosen for estimating $\hat{f}$. The weighted-sum method is utilised in Algorithm 1 with the weights $w = [5, 1]$. The search budget is chosen as 40 before stopping the algorithm (for the search space of approx. 800 samples in the grid). The starting point is the $(-1, -2)$ corner of the function. Figure 7 shows the computed location of two optimums, the 40 data points, as well as the estimated function based on the data points. The optimum locations found are (0, 0.7) and (0, −0.6) with respective values of 1 and 0.98 (estimate), whereas the real locations are $(-0.09, 0.71)$ and $(0.09, -0.71)$ with the value 1.03.

Next, the same function is optimised with the bounded objective method using Algorithm 3. The bound b is chosen as 0.05, which is satisfied after 29 iterations. Subsequently, a weighed search is conducted with $w = [0.1, 1]$, i.e. a ten fold emphasis on maximisation. The optimum locations found are (−0.1, 0.6) and (0.1, 0.7) with respective values of 0.94 and 1.03. These results are depicted in Figure 8.

Example 3

Here, first the Ackley function [1,5] is optimised on the *ten dimensional hypercube* $[-2, 2]$. Due to curse of dimensionality, the hypercube is uniformly sampled with 2000 points to obtain the initial search space Θ. The Gaussian kernel in (2) with variance 100 is chosen for estimating $\hat{f}$. The weights are

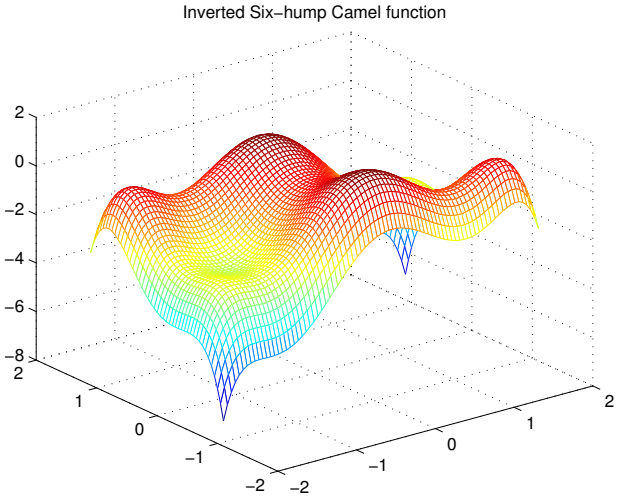

Fig. 6 The inverted six-hump camel function.

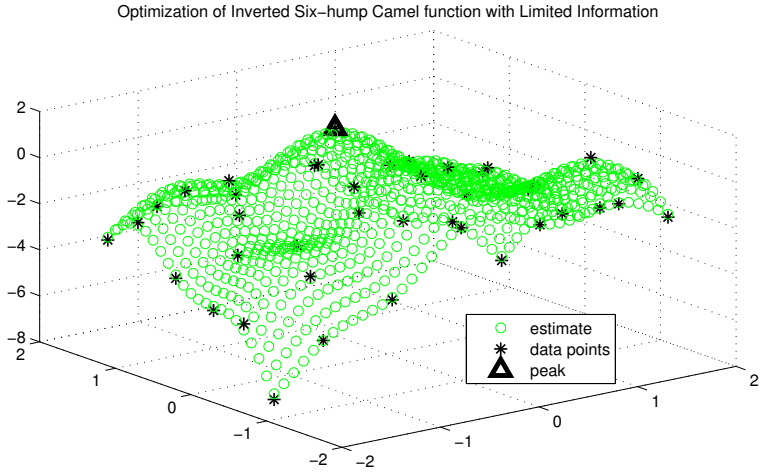

Fig. 7 Optimisation of the inverted six-hump camel function [10] using the weighted-sum method with 40 data points.

chosen as $w = [5, 1]$ in Algorithm 1 and %20 of the $N = 40$ search budget is allocated to hedging, i.e. $k = 5$ in Algorithm 2. Since the search space is very large due to high dimension, Algorithm 3 is modified. Instead of putting a bound on variance a modified meta optimisation scheme is implemented where the weight is changed to $w = [2, 1]$ after 20 steps. Concurrently, the search space Θ is redefined by sampling the unit hypercube around the maximum point at step 20. Hence, a simple multi-resolution approach is illustrated,

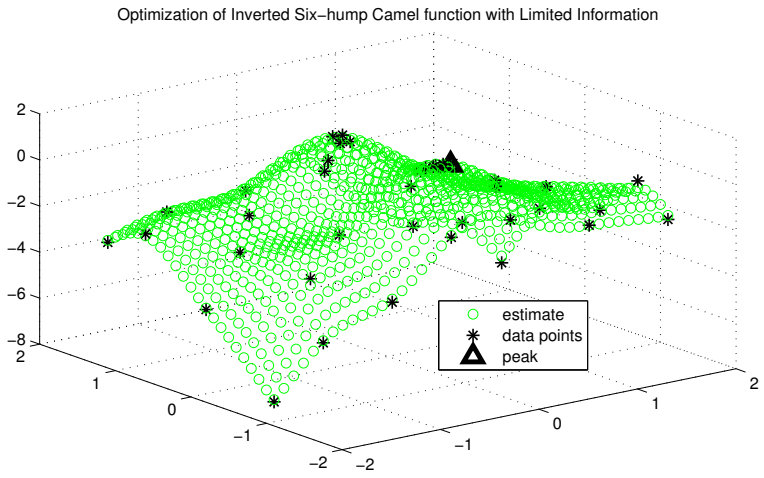

Fig. 8 Optimisation of the inverted six-hump camel function [10] using the bounded objective method with 40 data points.

which circumvents computational problems associated with sampling a high dimensional space. Due to the randomised nature of sampling, each algorithm is run 100 times.

The results are summarised in Table 2 indicate the limitations of a naive sampling of the search domain. The multi-resolution meta optimisation algorithm easily outperforms others. All algorithms perform better than uniform random sampling on average (except CMA-ES). The widely known CMA-ES [12] algorithm included in the table is limited to 40 function evaluations to match the information limitations in others. However, this is admittedly an "apples to oranges" comparison since clearly CMA-ES has not been designed to address the specific problem discussed in this paper. Therefore, it significantly under-performs compared to others.

Table 2 Ackley Function Optimised - 100 runs

Algorithm	Mean	Variance	Best in Θ
Weighed-sum	3.785	0.125	3.545
Hedged	3.782	0.152	3.487
Meta optimisation	2.750	0.150	2.318
Random	4.455	0.092	3.545
CMA-ES	5.590	0.188	x

Next, the same numerical analysis is conducted for the ten dimensional sphere function

$$y = \sum_{i=1}^{10} x_i^2,$$

on the same hypercube. The results summarised in Table 3 illustrate the value of the hedging algorithm over the weighted-sum one. This may indicate a modelling issue, e.g. a poor choice of kernel function or parameter. The remaining results are consistent with the previous case.

Table 3 Sphere Function Optimised - 100 runs

Algorithm	Mean	Variance	Best in Θ
Weighed-sum	3.751	1.191	2.670
Hedged	3.535	1.3345	2.569
Meta optimisation	1.570	0.458	0.658
Random	5.690	1.811	2.670
CMA-ES	11.455	13.375	x

6 Connections to Existing Literature

Decision making with limited information is related to search theory. The idea of using information (theory) in this context is hardly new as evidenced by the article "A New Look at the Relation Between Information Theory and Search Theory" from 1979 [26]. The subject is further studied in [14]. The topic of optimal search is more recently revisited by [45], which contains substantial historical notes and studies problems where the search target distribution in itself is unobservable.

The book [22] provides important and valuable insights into the relationship between information theory, inference, and learning. Measuring information content of experiments using Shannon information is explicitly mentioned and a slightly informal version of the bisection example in Subsection 3.1 is discussed. However, focusing mainly on more traditional coding, communication, and machine learning topics, the book does not discuss the type of decision making problems presented in this paper.

Learning plays an important role in the presented framework, especially *regression*, which is a classical machine (or statistical) learning method. A very good introduction to the subject can be found in [7]. A complementary and detailed discussion on kernel methods is in [32]. Another relevant topic is Bayesian inference [42, 22], which is in the foundation of the presented framework. In machine learning literature, Gaussian processes (GPs) are getting increasingly popular due to their various favourable characteristics. The book

[27] presents a comprehensive treatment of GPs. Additional relevant works on the subject include [22,32,21], which also discuss GP regression.

Convex optimisation [8] is a well-understood topic that is often easy to handle even if available information is limited. Optimising nonconvex functions, however, is still a research subject [15]. It is interesting to note that the method known as *kriging* in global optimisation is almost the same as GP regression in machine learning. The field *stochastic programming* focuses on optimisation under uncertainty but assumes a certain amount of prior knowledge on the problem at hand and models the uncertainty probabilistically [31]. The popular heuristic method *simulated annealing* [30] is essentially based on iterative random search. Another popular heuristic scheme particle swarm optimisation [16] is also based on random search but parallel in nature as a distinguishing characteristic rather than iterative.

Gaussian processes have been recently applied to the area of optimisation and regression [9] as well as system identification [41]. While the latter mentions active learning, neither work discusses explicit information quantification or builds a connection with Shannon information theory. The recent articles [19,43], which utilise GP regression for optimisation in a setting similar to the one in this paper and for state-space inference and learning, respectively, do not consider information-theoretic aspects of the problem, either. Likewise, the article [13] on stochastic black box optimisation, which considers a problem similar to the one here, does not take into account explicit measurement of information.

The area of active learning or experiment design focuses on data scarcity in machine learning and makes use of Shannon information theory among other criteria [34]. The paper [20] discusses objective functions which measure the expected informativeness of candidate measurements within a Bayesian learning framework. The subsequent study [33] investigates active learning for GP regression using variance as a (heuristic) confidence measure for test point rejection.

The recent paper [28] considers the shortest path problem on a graph with random edge weights, where side information about the edge weights becomes available as the graph is traversed. The performance is related to the total amount of side information received. That line of research is consistent with the discussion on the connection of greedy algorithms in Section 4 with successive approximation in general and Dijkstra's algorithm in particular.

7 Discussion

The foundation of the approach adopted in this paper is Bayesian inference, where the main idea is to choose an a priori model and update it with actual experimental data observed (see [22, Chap. 2] for a beautiful introductory discussion on the subject). As long as the a priori model is close to the reality (of the problem at hand), this inference methodology works very efficiently as indicated by the numerical examples in Section 5. In many cases this back-

ground information, which is sometimes referred to as "domain knowledge", is already available. However, in others one has to explore the model domain and learn model meta-parameters in a time scale naturally longer than the one of actual optimisation [21]. It is worth noting in this context that the "no free lunch" theorem still applies [44].

The GP regression adopted in the presented framework is only one method for function estimation and other, e.g. parametric, methods can easily replace GP for the regression part. In any case, the regression methodology here is consistent with the principle of "Occam's razor", more specifically its interpretation using Kolmogorov complexity [18]. A priori, the optimisation problems at hand are more probable to be simple rather than complex to describe in accordance with *universal distribution* [18]. Hence, given a data set it is reasonable to start describing it with the simplest explanation. GP regression already incorporates this line of thinking by relying on a kernel-based approach and making use of the representer theorem [27, Chap. 6.2].

This paper considers a class of problems where data is scarce and obtaining it is costly. Information theory plays an especially important role in devising optimal schemes for obtaining new data points (active learning). The entropy measure from Shannon information theory provides the necessary metric for this purpose, which quantifies the "exploration" aspect of the problem. Using a multi-objective optimisation formulation, the presented framework allows explicit weighting of *exploration* versus *exploitation* aspects. This trade-off is also very similar to one between the well-known depth-first versus breadth-first search algorithms in search theory.

The amount of information obtained from each data point is different here only because a specific a priori general model is utilised to explain the observed data (GP regression). Because of this the amount of information obtained is specific to the model. Otherwise, without this Bayesian approach, each data point would give the same information (inversely proportional to the total number of candidate points).

It is possible to combine the exploration and optimisation objectives in other ways. For example, if the maximum value of the objective function at the optimal point is known or can be estimated accurately, it is possible to develop a risk-based alternative multi-objective optimisation formulation as discussed in [3].

The numerical examples are provided to illustrate the approach rather than competing with existing approaches such as [13]. The classes of algorithms presented in Section 4 have significant room for improvement. However, the goal of this paper is not explore the huge algorithmic diversity in this domain but make a contribution by providing principles for developing various algorithms depending on the specific nature of the problems.

The optimisation approach presented here can also be interpreted from a biological perspective. If an analogy between the decision-maker and a biological organism is established, then the a-priori Bayesian model (meta parameters of the GP) that is refined over a long time scale corresponds to evolution of a species in an environment (problem domain). Each individual organism be-

longing to the species obtains new information to achieve its objective while preserving resources as much as possible. The existing evolutionary basis (GP model) gives them an advantage to find a solution much faster compared to random search. From the perspective of the species, it also makes sense for some of its members to explore the model (meta parameter) domain and further refine it through adaptation. Those with better meta parameters achieve then their objectives even more efficiently and obtain an evolutionary edge in natural selection (assuming competition).

8 Conclusion

The decision making framework presented in this paper addresses the problem of decision making under limited information by taking into account the information collection (observation), estimation (regression), and (multi-objective) optimisation aspects in a holistic and structured manner. The methodology is based on Gaussian processes and active learning. Various issues such as quantifying information content of new data points using information theory, the relationship between information and GP variance as well as related approximation and multi-objective optimisation schemes are discussed. The framework is demonstrated with multiple numerical examples.

The presented framework should be considered mainly as an initial step. Future research directions are abundant and include further investigation of the exploration-exploitation trade-off, adaptive weighting parameters, and random sampling methods for problems in higher dimensional spaces. Additional research topics are the relationship of the framework with genetic/evolutionary methods, dynamic control problems, and multi-person decision making, i.e. game theory.

9 Acknowledgements

The author wishes to thank Lacra Pavel, Slawomir Stanczak, Holger Boche, and Kivanc Mihcak for stimulating discussions on the subject. A precursor to this paper "A framework for optimisation under limited information," by the same author has been published in the 5th Intl. Conf. on Performance Evaluation Methodologies and Tools (ValueTools) in ENS, Cachan, France, May 2011.

References

1. Ackley, D.: A connectionist machine for genetic hillclimbing. Kluwer Boston Inc., Hingham, MA (1987)
2. Ahmed, N., Gokhale, D.: Entropy expressions and their estimators for multivariate distributions. IEEE Transactions on Information Theory **35**(3), 688–692 (1989). DOI 10.1109/18.30996

3. Alpcan, T.: A risk-based approach to optimisation under limited information. In: Proc. of the 20th Intl. Symp. on Mathematical Theory of Networks and Systems (MTNS) (2012)
4. Alpcan, T., Fan, X., Başar, T., Arcak, M., Wen, J.T.: Power Control for Multicell CDMA Wireless Networks: A Team Optimization Approach. Wireless Networks **14**(5), 647–657 (2008). DOI 10.1007/s11276-006-0006-5. URL papers/Alpcan-Winet.pdf
5. Back, T.: Evolutionary algorithms in theory and practice. Oxford University Press (1996)
6. Bellman, R.: Dynamic Programming. Dover Publications, Mineola, NY, USA (1957). URL http://www.bibsonomy.org/bibtex/29cdd821222218ded252c8ba5cd712666/pcbouman
7. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
8. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York, NY, USA (2004)
9. Boyle, P.: Gaussian Processes for Regression and Optimisation. Ph.D. thesis, Victoria University of Wellington, Wellington, New Zealand (2007). URL http://researcharchive.vuw.ac.nz/handle/10063/421
10. Dixon, L.C.W., Szego, G.P.: The optimization problem: An introduction. In: L.C.W. Dixon, G.P. Szego (eds.) Towards Global Optimization II. North-Holland, New York, NY, USA (1978)
11. Grodzevich, O., Romanko, O.: Normalization and Other Topics in Multi-Objective Optimization. Proceedings of the Fields–MITACS Industrial Problems Workshop (2006). URL http://www.maths-in-industry.org/miis/233
12. Hansen, N., Gemperle, F., Auger, A., Koumoutsakos, P.: When Do Heavy-Tail Distributions Help? In: T.P. Runarsson, et al. (eds.) Parallel Problem Solving from Nature PPSN IX, *Lecture Notes in Computer Science*, vol. 4193, pp. 62–71. Springer (2006)
13. Huang, D., Allen, T., Notz, W., Zeng, N.: Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models. Journal of Global Optimization **34**(3), 441–466 (2006). DOI 10.1007/s10898-005-2454-3
14. Jaynes, E.T.: Entropy and Search-Theory. In: C.R. Smith, J. W. T. Grandy (eds.) Maximum-Entropy and Bayesian Methods in Inverse Problems, p. 443. Springer (1985). URL http://bayes.wustl.edu/etj/articles/search.pdf
15. Jones, D.R.: A Taxonomy of Global Optimization Methods Based on Response Surfaces. Journal of Global Optimization **21**(4), 345–383 (2001). DOI 10.1023/A:1012771025575. URL http://dx.doi.org/10.1023/A:1012771025575
16. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. of IEEE Intl. Conf. on Neural Networks, vol. 4, pp. 1942–1948 (1995). DOI 10.1109/ICNN.1995.488968
17. Lawrence, N., Seeger, M., Herbrich, R.: Fast sparse Gaussian process methods: The informative vector machine. Advances in neural information processing systems **15**, 609–616 (2002)
18. Li, M., Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications, 2nd edn. Texts in Computer Science. Springer, New York, NY, USA (1997)
19. Lizotte, D., Wang, T., Bowling, M., Schuurmans, D.: Gaussian process regression for optimization. in NIPS 2005 Workshop on Value of Information in Inference, Learning and Decision-Making (2005). URL http://domino.research.ibm.com/comm/research_projects.nsf/pages/nips05workshop.index.html
20. MacKay, D.J.C.: Information-Based Objective Functions for Active Data Selection. Neural Computation **4**(4), 590–604 (1992). DOI 10.1162/neco.1992.4.4.590. URL http://www.mitpressjournals.org/doi/abs/10.1162/neco.1992.4.4.590
21. MacKay, D.J.C.: Introduction to Gaussian Processes. In: C.M. Bishop (ed.) Neural Networks and Machine Learning, NATO ASI Series, pp. 133–166. Kluwer Academic Press (1998)
22. MacKay, D.J.C.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press (2003). URL http://www.inference.phy.cam.ac.uk/mackay/itila
23. Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. Structural and Multidisciplinary Optimization **26**(6), 369–395 (2004). DOI 10.1007/s00158-003-0368-6. URL http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1007/s00158-003-0368-6

24. Pan, Y., Pavel, L., Alpcan, T.: A System Performance Approach to OSNR Optimization in Optical Networks. IEEE Transactions on Communications **58**(4), 1193–1200 (2010). DOI 10.1109/TCOMM.2010.04.090059. URL papers/TComm_preprint_v7.pdf
25. Petersen, K.B., Pedersen, M.S.: The Matrix Cookbook (2008). URL http://www2.imm.dtu.dk/pubdb/p.php?3274
26. Pierce, J.G.: A New Look at the Relation Between Information Theory and Search Theory. Tech. rep., Office of Naval Research, Arlington, VA, USA (1978). URL http://handle.dtic.mil/100.2/ADA063845
27. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press (2005)
28. Rinehart, M., Dahleh, M.: The value of sequential information in shortest path optimization. In: IEEE American Control Conference (ACC), pp. 4084–4089. IEEE (2010)
29. Royden, H.L.: Real Analysis, 3rd edn. Prentice-Hall, New Jersey, USA (1988)
30. Rutenbar, R.: Simulated annealing algorithms: an overview. IEEE Circuits and Devices Magazine **5**(1), 19–26 (1989). DOI 10.1109/101.17235
31. Sahinidis, N.V.: Optimization under uncertainty: state-of-the-art and opportunities. Computers & Chemical Engineering **28**(6-7), 971–983 (2004). DOI 10.1016/j.compchemeng.2003.09.017. URL http://www.sciencedirect.com/science/article/B6TFT-49YH97T-1/2/f15875aad97740410effc526416289aa. FOCAPO 2003 Special issue
32. Scholkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA (2001)
33. Seo, S., Wallat, M., Graepel, T., Obermayer, K.: Gaussian process regression: active data selection and test point rejection. In: Proc. of IEEE-INNS-ENNS Intl. Joint Conf. on Neural Networks IJCNN 2000, vol. 3, pp. 241–246 (2000). DOI 10.1109/IJCNN.2000.861310
34. Settles, B.: Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2009)
35. Sniedovich, M.: A new look at Bellman's principle of optimality. Journal of optimization theory and applications **49**(1), 161–176 (1986)
36. Sniedovich, M.: Dijkstra's algorithm revisited: the dynamic programming connexion. Control and cybernetics **35**(3), 599 (2006)
37. Sniedovich, M., Lew, A.: Dynamic Programming: an overview. Control and Cybernetics **35**(3), 513 (2006)
38. Srikant, R.: The Mathematics of Internet Congestion Control. Systems & Control: Foundations & Applications. Birkhauser, Boston, MA (2004)
39. Tempo, R., Bai, E.W., Dabbene, F.: Probabilistic robustness analysis: Explicit bounds for the minimum number of samples. Systems & Control Letters **30**(5), 237–242 (1997). DOI 10.1016/S0167-6911(97)00005-4. URL http://www.sciencedirect.com/science/article/B6V4X-3SP7DCD-4/2/3dc655107eff50f12b326ea10f58ef0a
40. Tempo, R., Calafiore, G., Dabbene, F.: Randomized Algorithms for Analysis and Control of Uncertain Systems. Springer-Verlag, London, UK (2005)
41. Thompson, K.R.: Implementation of gaussian process models for non-linear system identification. Ph.D. thesis, University of Glasgow, Glasgow, Scotland (2009)
42. Tipping, M.E.: Bayesian Inference: An Introduction to Principles and Practice in Machine Learning. In: Advanced Lectures on Machine Learning, pp. 41–62 (2003). URL http://springerlink.metapress.com/openurl.asp?genre=article
43. Turner, R., Deisenroth, M.P., Rasmussen, C.E.: State-Space Inference and Learning with Gaussian Processes. In: Proc. of 13th Intl. Conf. on Artificial Intelligence and Statistics (AISTATS). Chia Laguna Resort, Sardinia, Italy (2010)
44. Wolpert, D., Macready, W.: No free lunch theorems for optimization. Evolutionary Computation, IEEE Transactions on **1**(1), 67–82 (1997). DOI 10.1109/4235.585893
45. Zhu, Q., Oommen, J.: On the optimal search problem: the case when the target distribution is unknown. In: Proc. of XVII Intl. Conf. of Chilean Computer Science Society, pp. 268–277 (1997). DOI 10.1109/SCCC.1997.637100