Addressing Solution Quality in Data Generated Optimization Models

Orit Davidovich,[1] Parikshit Ram,[2] Segev Wasserkrug,[1] Dharmashankar Subramanian,[2] Nianjun Zhou,[2] Dzung Phan,[2] Pavankumar Murali,[2] Lam M. Nguyen[2]

[1]IBM Research, IBM Haifa Research Lab
[2]IBM Research, Yorktown Heights
orit.davidovich@ibm.com, parikshit.ram@ibm.com, segevw@il.ibm.com, dharmash@us.ibm.com, jzhou@us.ibm.com, phandu@us.ibm.com, pavanm@us.ibm.com, lamnguyen.mltd@ibm.com

Abstract

Mathematical optimization models can improve decision making in a wide variety of industrial applications, including process control in manufacturing and materials processing. However, creating an optimization model requires rare optimization and domain expertise and a significant amount of time, thereby limiting the widespread use of this technology. One way to overcome this is by utilizing historical data to learn the relevant parts of the optimization model. However, as historical decisions may be suboptimal, the data used for training may not be representative of an optimal operating region for the control set points, resulting in inaccuracies very different from the ones in traditional machine learning settings, and which pose significant challenges in learning good optimization models.

In this work, we present a formal approach for addressing such challenges in the automated generation of optimization models, in order to improve the quality of the solutions produced by the generated models. Our approach consists of: a) a formal definition of the measure of quality of the generated model; b) a Gaussian Process approach, with a strong theoretical basis which, under some assumptions, provides an accurate quality estimate of the generated models' quality, backed by extensive empirical analysis; and c) methods to augment the generated optimization model with additional constraints so as to obtain high quality (as defined by our measure) optimization models.

1 Introduction

Mathematical optimization can provide decision support to a variety of real world problems across several application domains. Creating an optimization model requires both modeling the *constraints* which govern the system, as well as the *objective function* to be optimized. Currently, creating a mathematical optimization model requires optimization modeling expertise, which is quite rare, as well as significant time, typically measured in months. This severely limits the application of mathematical optimization and the benefits it currently provides. Therefore, to realize the true business benefits of mathematical optimization, there is a need to enable non-experts to create such models in an automated manner, in a much shorter amount of time.

1.1 Problem Definition

Consider a complex production or manufacturing system such as oil and gas, mining, or food processing. In such problems, optimization decision variables include control variables, for which optimal set points are to be identified. Traditionally, first-principle models based on thermodynamics, heat and mass transfer, chemical kinetics etc. are used to model each process in such systems. Those equations would then need to be translated into a formulation that can be efficiently solved by mathematical solvers. This approach requires significant time, skills, and domain understanding. With the introduction of Industry 4.0, there is a widespread availability of fine-grained IoT sensor data that captures plant behavior. This allows the utilization of a data-driven approach towards process control (Phan et al. 2021), which aims to use machine learning techniques to learn and represent relationships between inputs and outputs for each process. The regression-based relationships can then be automatically converted into an optimization model to obtain temporal recommendations on optimal set points (Subramanian et al. 2019). This *two-stage prediction-optimization* based approach is an example of data driven automated optimization modeling.

To formally define our problem setup, we assume there is a system which can be described through a set of functions $f(x,p), g_k(x,p), k \in [K] = \{1,\ldots,K\}$ with $x \in \mathcal{X} \subset \mathbb{R}^{d_x}$ as the set of decision variables and $p \in \mathcal{P} \subset \mathbb{R}^{d_p}$ as uncontrollable inputs exogenously provided (with d_x and d_p being the dimensions of the x and p vectors respectively). Given an input $p \in \mathcal{P}$, the goal is to find the optimal decision $x^*(p)$ such that:

$$\begin{aligned} & x^*(p) = \arg\min_x f(x,p) \text{ subject to } x \in \mathcal{X} \cap \Omega(p) \\ & \text{where } \Omega(p) = \{x \in \mathbb{R}^{d_x} : g_k(x,p) \le 0, \forall k \in [K]\}. \end{aligned} \tag{1}$$

However, we do not have access to this model. Rather, we have access to historical data from which we wish to learn this model automatically. Such historical data for automated optimization model derivations needs to contain explicit (often suboptimal) choices made by decision makers, and which therefore cannot be naturally modeled by a probability distribution. As a result, such data may not cover, or even be representative of, all possible decisions or set points. In a traditional prediction-optimization approach regression models trained on such historical datasets may therefore erroneously generalize to decision domains not covered in the data. When such models are used within a set point optimization model, poor decisions may be produced. Therefore, in this work, we provide an approach for addressing optimization solution quality. We formally define the quality of the generated model, provide a data-driven method to estimate the model's quality, and describe approaches for generating models that meet the quality requirement.

1.2 Contributions

We summarize our main contributions as follows. Firstly, in §2, we provide a formal definition to ensure that the generated solution is, with a high enough probability, an improvement over the decisions used to generate the data. Secondly, in §3, we propose a Gaussian Process based approach that enables to measure the probability that the generated model provides an improving solution. Lastly, to deal with uncertainty for prediction models, in §4, we introduce two new notions: data sufficiency and model fidelity, and then leverage them to form additional constraints for the optimization model to improve the original solution. Our main goal is to provide a theoretical foundation and tools, deferring a comprehensive end-to-end solution to future work.

2 Defining Optimization Solution Quality

In this paper, we assume that the true model of the system defined by $\{f, \Omega\}$, as specified in Equation (1), is not known. Rather, we have a set of historical data $\mathcal{D}$ so that each $d_i \in \mathcal{D}$ is composed of a tuple $d_i = \langle x_i, p_i, f(x_i, p_i)\rangle$ such that $x_i \in \mathcal{X} \cap \Omega(p_i)$. Moreover, p is generated i.i.d. from some distribution. From this data, we need to learn an optimization model [1]. $\{\widehat{f}, \widehat{\Omega}\}$. Given a vector pair p drawn according to the predetermined distribution, we can then use this model to find $\widehat{x}^\star(p) = \arg\min_x \widehat{f}(x, p)$ subject to $x \in \mathcal{X} \cap \widehat{\Omega}(p)$.

The first condition for $\widehat{x}^\star$ to be a good solution is that it is feasible with a high enough probability, induced by distributions of p and $\mathcal{D}$, i.e.:

$$\Pr(\widehat{x}^\star(p) \in \Omega(p)) \geq (1 - \delta_1) \tag{2}$$

We also need to ensure that $\widehat{x}^\star$ provides a sufficiently good function value. One way to define this is that the solution found by the approximate model $\{\widehat{f}, \widehat{\Omega}\}$ ϵ-approximates the optimal solution of the true system with a high probability. However, this is extremely difficult to guarantee given the limited data and the fact that the above does not assume any prior knowledge about the distribution of x. Therefore, we will define a good solution by how much it improves over the existing *decision policy*, which would capture for example, the way that currently such decisions are made. Assuming that the decisions are sampled from a distribution $H_{|p}$ conditioned on the uncontrollable p, let $X_{H_{|p}} \sim H_{|p}$. Then we can define the following condition:

$$\Pr(f(X_{H_{|p}}, p) - f(\widehat{x}^\star(p), p) \geq \epsilon) \geq (1 - \delta_2) \tag{3}$$

which specifies that $\widehat{x}^\star(p)$ improves upon $H_{|p}$ by ϵ with a high enough probability[2].

Equations (2) and (3) formalize the desired properties of optimization solution quality using a probabilistic approach. These equations motivate the development of techniques to quantify the desired probabilities. In the next section, we present a novel Bayesian approach using Gaussian processes to quantify these probabilities.

3 Gaussian Process for Quality Estimate

Let us consider a simpler version of the optimization problem discussed in Section 2. Consider $\mathcal{X}, \Omega \subseteq \mathbb{R}^{d_x}$ in the space of decision variables, $\mathcal{P} \subseteq \mathbb{R}^{d_p}$ in the space of uncontrolled variables, and an objective target $f : \Omega \times \mathcal{P} \to \mathbb{R}$. We will assume Ω is closed and bounded. Given some $p \in \mathcal{P}$, we wish to solve the optimisation problem

$$x^\star(p) = \text{argmin}_{x \in \mathcal{X} \cap \Omega} f(x, p). \tag{4}$$

In this formulation, the feasibility region Ω is independent of p and $\mathcal{X}$ is known. In applications, Ω is defined through a finite set of inequalities $\{g_k(x) \leq 0\}$, while the set $\mathcal{X}$ may be discrete (e.g., a lattice in integer programming). In other words, in this setup, $\mathcal{X}$ imposes the (possible) discretization of the optimisation problem.

Assume we do not have direct knowledge of either f or Ω. Instead, we are given $\widehat{\Omega} \subseteq \mathbb{R}^d$ and $\widehat{f} : \widehat{\Omega} \times \mathcal{P} \to \mathbb{R}$. Given $p \in \mathcal{P}$, we can solve the ersatz optimization problem

$$\widehat{x}^\star(p) = \text{argmin}_{x \in \mathcal{X} \cap \widehat{\Omega}} \widehat{f}(x, p). \tag{5}$$

Our goal is to find a way to evaluate the quality of $\widehat{x}^\star(p)$ as a substitute for $x^\star(p)$. We wish to introduce a metric by which to assess $\widehat{x}^\star(p)$, independent of any means of producing $\widehat{\Omega}$ and $\widehat{f}$ or solving the optimization problem (5). Instead, of trying to assess $\widehat{x}^\star(p)$ against $x^\star(p)$ of which we have no direct knowledge, we can consider $\widehat{x}^\star(p)$ relative to some $x_0 \in \mathbb{R}^{d_x}$ appropriate for p, for instance, the current best practice or policy. Concretely, we would like to estimate the probability that the proposed optimum $\widehat{x}^\star(p)$ improves upon a known policy x_0 associated with p. We seek an estimate of the *probability of improvement* of $\widehat{x}^\star(p)$ relative to x_0 at p,

$$\Pr[f(\widehat{x}^\star(p), p) < f(x_0, p)] \tag{6}$$

To estimate the probability of improvement (6), we can view both f and Ω stochastically.[3] In fact, restricting Ω to subsets defined via a finite set of (affine, quadratic, etc.) inequalities $\{g_k(x) \leq 0\}$ provides us with a well-defined notion of stochasticity for Ω.[4] From now on, we will assume $\{g_k\}$ to be affine. Hence, all feasibility regions Ω will be assumed convex.[5] By limiting ourselves to affine inequalities and fixing the number of inequalities, $1 \leq k \leq K$, we can view

[1] As mentioned previously, x_i may not be the optimal decision for p_i. For example, it could be a decision made on best practices or rules of thumb

[2] Alternative formulations of relative improvements are also possible, but this is the one we focus on in this work

[3] A possible discretization via $\mathcal{X}$ may break stochasticity, therefore, we will not assume stochasticity for $\text{argmin}_{\mathcal{X} \cap \Omega} f(\cdot, p)$.

[4] This requires an upper bound on the number of inequalities.

[5] Introducing stochasticity both for f and Ω extends the notion of a stochastic optimization problem in which Ω is assumed

Ω as a random variable defined over $M_{(d_x+1)\times K}(\mathbb{R})$; each set of affine inequalities $\{g_k(x) \leq 0\}$ can be succinctly expressed as $A\bar{x} \leq 0$, where $\bar{x} = (x, 1) \in \mathbb{R}^{d_x+1}$ is the homogenization of x and $A \in M_{(d_x+1)\times K}(\mathbb{R})$ is a matrix of dimension $(d_x + 1) \times K$.

Gaussian processes (GPs) provide us with the mathematical machinery to view f stochastically. A Gaussian process governs the stochastic properties of a random function in the same way that a Gaussian probability distribution governs the properties of a random variable (Williams and Rasmussen 2006). A Gaussian process on f establishes $f(x, p) \sim \mathcal{N}(\mu(x, p), \sigma^2(x, p))$ for every $(x, p) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_p}$. Considered as two normally distributed random variables, $f(\widehat{x}^\star, p)$ and $f(x_0, p)$, the probability of improvement (6) immediately follows.

Assume we have a prior $\Omega \sim \mathcal{O}$. For example, we may assume a fixed number K of affine inequalities defining the feasibility region, so that $\mathcal{O}$ is a distribution over $M_{(d_x+1)\times K}(\mathbb{R})$. Starting from data $\mathcal{D} = \{\langle x_i, p_i, y_i\rangle\}$, we establish a joint posterior $GP_{f,\Omega}$ that ties f to Ω, so that

$$GP_{f,\Omega}(\Omega = \widehat{\Omega}) \equiv GP_{f\,|\,\mathcal{D}\cap\widehat{\Omega}} \tag{7}$$

its right-hand side being the GP posterior established on f given $\mathcal{D}$ restricted to $\widehat{\Omega}$. By abuse of notation, we use $\mathcal{D} \cap \widehat{\Omega}$ to mean $\mathcal{D}$ restricted to $\widehat{\Omega}$.

To calculate the probability of improvement (6), we use $GP_{f,\Omega}$ and marginalize over Ω as a way to capture the uncertainty on Ω introduced through $\mathcal{O}$. The right-hand side of (7) suggests defining $m : \widehat{\Omega} \mapsto \mathcal{D} \cap \widehat{\Omega} \in 2^{\mathcal{D}}$ that maps a feasibility region $\widehat{\Omega}$ into a subset of $\mathcal{D}$. The map m induces a finite partition on the space of Ω's.[6] When marginalizing over Ω, m allows us to translate the integral over Ω into a finite sum. Hence, $GP_{f,\Omega}$ determines the marginalized probability of improvement

$$\begin{aligned}\Pr[f(\widehat{x}^\star, p) < f(x_0, p)] = \\ \sum_{\mathcal{D}' \subseteq \mathcal{D}} \omega_{\mathcal{D}'} \Pr_{GP_{f|\mathcal{D}'}}[f(\widehat{x}^\star, p) < f(x_0, p)]\end{aligned} \tag{8}$$

with

$$\omega_{\mathcal{D}'} = \int_{m^{-1}(\mathcal{D}')} d\Omega \tag{9}$$

where the measure $d\Omega$ is strictly determined by $\mathcal{O}$.

There is no reason to assume monotonicity $\omega_{\mathcal{D}'} \leq \omega_{\mathcal{D}''}$, for $\mathcal{D}' \subset \mathcal{D}''$ or $\omega_{\mathcal{D}} = 1$. When the data $\mathcal{D}$ is contained in Ω with probability 1, i.e., $\Pr_{\mathcal{O}}[\mathcal{D} \subseteq \Omega] = 1$, then $\omega_{\mathcal{D}'} = 0$ for any $\mathcal{D}' \subsetneq \mathcal{D}$ and $\omega_{\mathcal{D}} = 1$. In that case, (8) reduces to

$$\begin{aligned}\Pr[f(\widehat{x}^\star, p) < f(x_0, p)] = \\ \Pr_{GP_{f|\mathcal{D}}}[f(\widehat{x}^\star, p) < f(x_0, p)]\end{aligned} \tag{10}$$

By definition (9), $\omega_{\mathcal{D}'}$ is negligible whenever it is improbable that a *likely* Ω will intersect $\mathcal{D}$ at $\mathcal{D}'$. In general, there is no reason to assume all data points are feasible.

To better understand the probability of improvement defined in (8), let us examine its relation to

$$r_1(\widehat{\Omega}) = |\widehat{\Omega} \cap \mathcal{D}| / |\mathcal{D}| \tag{11}$$

$$r_2(\widehat{\Omega}) = |\mathsf{Conv}\left(\widehat{\Omega} \cap \mathcal{D}\right)| / |\widehat{\Omega}| \tag{12}$$

where $|\cdot|$ designates cardinality in (11) and volume in (12). $r_1(\widehat{\Omega})$ measures how much of $\mathcal{D}$ is covered by $\widehat{\Omega}$, while $r_2(\widehat{\Omega})$ measures how much data-less volume $\widehat{\Omega}$ contains.[7]

Consider the case where $\omega_{\mathcal{D}}$ is close to 1, that is, $d\Omega$ is concentrated around feasibility regions that cover most of $\mathcal{D}$. In this case, $\omega_{\mathcal{D}'}$ is commensurable with $|\mathcal{D}'|/|\mathcal{D}|$: $\omega_{\mathcal{D}'}$ is low whenever $|\mathcal{D}'|/|\mathcal{D}|$ is low, and vice versa. On the other hand, whenever $\omega_{\mathcal{D}}$ is close to 0, it is highly unlikely that all data points are feasible. Here, the relation is reverse: $\omega_{\mathcal{D}'}$ is low whenever $|\mathcal{D}'|/|\mathcal{D}|$ is high, and vice versa. The contribution of $\mathcal{D}'$ to the finite sum in (8) is dampened when the probability of improvement on the right-hand side is calculated based on infeasible data points. Thus, $GP_{f,\Omega}$ captures $r_1(\cdot)$ through ω, either directly or inversely, depending on $\mathcal{O}$.

As we move away from the data, the GP posterior is predominantly determined by the prior on f. Consequently, a prior centered around the empirical mean of f in $\mathcal{D}$ produces a GP posterior that will never locate an optimal solution far away from $\mathcal{D}$. Hence, there will be little chance of improvement for $\widehat{x}^\star(p)$ located in data-less regions: the probability on the right-hand side of (8) will necessarily be low.[8]

Choosing to establish a GP posterior on f irrespective of $\mathcal{O}$ does not fully address $r_i(\cdot)$. As stated in (10), it is equivalent to calculating the probability of improvement assuming that $\Pr_{\mathcal{O}}[\mathcal{D} \subseteq \Omega] \approx 1$.

The probability of improvement (8) is a strong metric with which to assess prediction-optimization schemes, namely, any scheme that, given data $\mathcal{D}$ associated with an objective target and given knowledge, i.e., constraints, produces an optimal solution $\widehat{x}^\star(p)$ as in (5). It combines the potential for gain on f with the risk of choosing $\widehat{x}^\star(p)$ far from what the expert believes or the data considers to be sensible constraints, while also making sure that we learn f from the most relevant data points. As such, it provides an answer to the challenge we posed above.

3.1 Empirical Analysis of Quality Estimate

To gauge the value of the probability of improvement (8) as an assessment metric, we ran the following simulation. We generated multiple triplets $(f, \mathcal{D}, \Omega)$, where f is a continuous piece-wise linear function, $\mathcal{D}$ is noisy data associated

known. As an extension of a stochastic optimization problem, (4) assumes the form $x^\star = \operatorname{argmin}_{\mathcal{X}\cap\mathbb{E}[\Omega|\mathbf{x},\mathbf{p}]} \mathbb{E}\left[f|\mathbf{x}, \mathbf{p}\right](\cdot, p)$. When Ω is known, there is an equivalence of linear optimization problems on $\mathcal{X} \cap \Omega$ and $\mathsf{Conv}\,(\mathcal{X} \cap \Omega)$, allowing us to remove $\mathcal{X}$ from the formulation. However, when Ω is randomized, we can no longer do so.

[6] The space of all closed, bounded, convex domains Ω defined through sets of affine inequalities $\{g_k(x) \leq 0\}$

[7] The two can be combined into $r(\widehat{\Omega}) = r_1(\widehat{\Omega}) \cdot r_2(\widehat{\Omega}) \in [0, 1]$, where $r(\widehat{\Omega}) = 1$ iff $\widehat{\Omega} = \mathsf{Conv}\,(\mathcal{D})$ and $r(\widehat{\Omega}) = 0$ iff $\widehat{\Omega} \cap \mathcal{D} = \emptyset$.

[8] One key advantage of GP models is their being model-free. By contrast, a linear model (or any other parametric model) may very well locate the optimal solution away from the data.

with f, and Ω are linear constraints sampled from a distribution $\mathcal{O}$ (Davidovich 2021). All feature vectors in $\mathcal{D}$ were considered decision variables.

We ran a simple two-stage prediction-optimization algorithm and calculated the probability of improvement for its proposed solution $\widehat{x}^\star$ against policies x_0 in the data. We detail the synthetic data generation process for the analysis of the quality estimate in Appendix **??**.

For each generated triplet $(f, \mathcal{D}, \Omega)$ with $\mathcal{D} = (X, y)$, we trained a linear regression model on the feasible data points $(x, y) \in \mathcal{D}$ with $x \in \Omega$ and fed the resulting model as the linear objective target together with the sampled constraints Ω to the PuLP solver (Mitchell, O'Sullivan, and Dunning 2011). The result was a solution $\widehat{x}^\star \in \Omega$ to the ersatz minimization problem (5). This predict-then-optimize scheme is clearly inadequate. However, the purpose was not to produce a good algorithm, but rather to test the probability of improvement both on good and bad proposed solutions.

To establish the GP posterior and calculate the probability of improvement, we utilized the python package GPy (GPy 2012), where we maximized the marginal likelihood to establish the hyper-parameters of the GP kernel. Maximum likelihood (ML) is numerically sensitive to datasets that produce a flat-ish landscape or a landscape with multiple critical points. In such cases, optimization of the marginal likelihood may fail. Triplets $(f, \mathcal{D}, \Omega)$ for which ML failed were removed. We note the option of using MCMC methods, though they impose considerable run-time costs on a simulation at this scale.

Simulation results are summarized in Figure (1). We randomly generated 1718 triplets $(f, \mathcal{D}, \Omega)$ with domain $\text{dom}(f) \subseteq \mathbb{R}^d$ for $d = 2, 3$. The generated PWL functions f differed in complexity (e.g., number of simplices in $\mathcal{T}$), while the data $\mathcal{D}$ varied in number of policies (2, 3, or 5) and number of data points (500 or 1000).

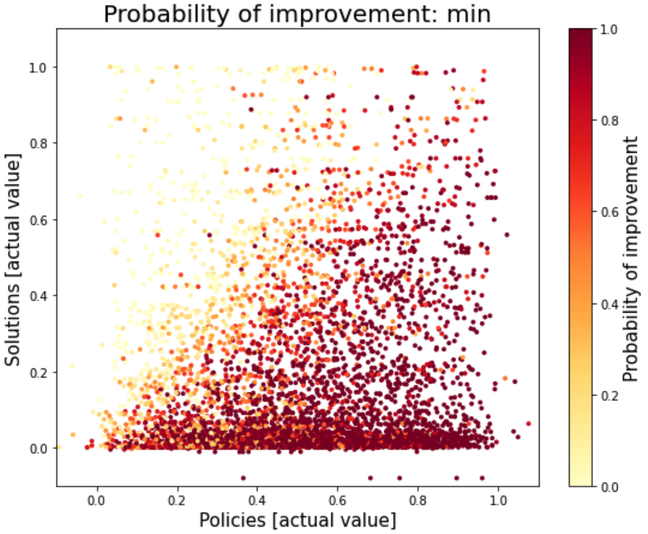


Figure 1: Probability of improvement calculated for 5593 solution-policy pairs associated with 1718 randomly generated triplets (f, D, Ω).

Each data point in Figure (1) stems from a single triplet $(f, \mathcal{D}, \Omega)$ and corresponds to the probability of improvement calculated for a pair $(\widehat{x}^\star, x_0)$, where $\widehat{x}^\star$ is an estimated arg min produced from $(\mathcal{D}, \Omega)$ and x_0 is a policy in $\mathcal{D}$. Figure (1) includes 5593 solution-policy pairs.[9] The horizontal axis of policies x_0 in Figure (1) is normalized according to $(f(x_0) - f_{\min})/(f_{\max} - f_{\min})$ using the true known values of f; $f_{\min}$ ($f_{\max}$) is the true known global minimum (maximum) of f in dom(f). The vertical axis of solutions $\widehat{x}^\star$ is normalized mutatis mutandis.

We expect a strong assessment metric of data-driven optimization algorithms to show a light-to-dark color gradient going from the top left corner of Figure 1 down to its bottom right. Near the top left corner, solutions are valued close to function maximum while policies are valued close to function minimum. The probability of improvement there should near 0, as we are considering minimization. Near the bottom right corner, the opposite holds; solutions are valued close to function minimum while policies are valued close to function maximum. The probability of improvement should then near 1.

There are two cases to consider where probability of improvement fails: false negatives and false positives. False negatives are instances of pairs $(\widehat{x}^\star, x_0)$ for which the solution resulted in an actual improvement, but the probability of improvement estimate was below the threshold (which we defined to be 0.5 in this analysis) . False negatives appear in Figure (1) as light colored data points below the diagonal . False negatives represent a missed opportunity for improvement rather than a call to action. False positives, on the other hand, are instances of pairs $(\widehat{x}^\star, x_0)$ for which the probability of improvement should have been below 0.5 but was estimated to be above 0.5. We can see false positives in Figure (1) as dark colored data points above the diagonal. They represent scenarios in which a change in policy will produce worse outcomes, and thus are far less desirable than false negatives.[10]

The results are summarized in Table (1). The bottom left cell represents false negatives (9.2%), and the upper right cell false positives (2.6%). As can be seen, the overall accuracy of the metric is quite high, and errors are biased strongly towards the less important false negatives [11]. Therefore, the probability of improvement (8) demonstrates the characteristics for a strong assessment metric of data-driven optimization algorithms.

[9] To run our simulation at scale, we used the cloud distribution framework ray (Moritz et al. 2018) and the event management framework rayvens built on top of ray (Gheorghe-Teodor Bercea, Olivier Tardieu 2021). We ran our simulation on an OpenShift cluster with 16 CPU x 64 RAM x 3 workers.

[10] The accumulation of data points at the bottom of Figure (1) shows instances in which the predict-then-optimize algorithm was successful at identifying the global minimum.

[11] Note that changing the threshold is expected to reduce even further the number of false positives

	$f(\widehat{x}^{\star}) < f(x_0)$	$f(\widehat{x}^{\star}) > f(x_0)$
Predicted PI > 0.5	4201	146
Predicted PI < 0.5	513	733

Table 1: False negatives and false positives in probability of improvement predicted vs. actual.

4 Obtaining Good Quality Solutions

4.1 Problem Augmentation with GPs

Assume all we have is data $\mathcal{D}$.[12] Is there a way for us to augment the feasibility region Ω using $\mathcal{D}$ so as to ensure a solution that meets our quality metric? To address this question, consider $GP_{f|\mathcal{D}}$ established on the entire dataset, with its derived SD $\sigma_{GP_{f|\mathcal{D}}}$. Given $\kappa_i \geq 0$, $i = 1, 2$, we consider the following function defined on feasibility regions,[13]

$$s_{\kappa_1,\kappa_2}(\Omega) = r_1(\Omega)^{-\kappa_1} r_2(\Omega)^{-\kappa_2} \frac{1}{|\Omega|} \int_{\Omega} \sigma_{GP_{f|\mathcal{D}}}(x)\, dx \tag{13}$$

where r_1 and r_2 are as in (11) and (12) respectively.

The function s_{κ_1,κ_2} incorporates three terms. The first term involving r_1 penalizes Ω's that contain too little of $\mathcal{D}$. The second term involving r_2 penalizes Ω's that encompass too much data-less volume. The third term involving $\sigma_{GP_{f|\mathcal{D}}}$ is an estimate of expected model uncertainty in Ω.

Definition 1. *A feasibility region Ω^* is optimal w.r.t. $GP_{f|\mathcal{D}}$ and (κ_1, κ_2) if*

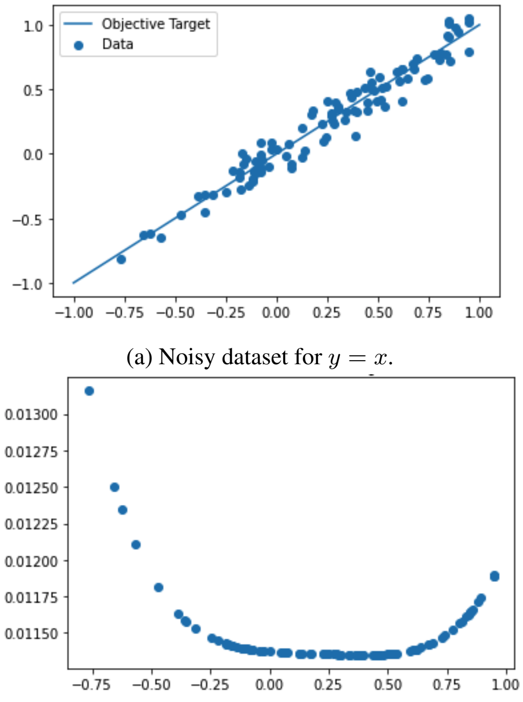
$$s_{\kappa_1,\kappa_2}(\Omega^*) = \min_{\Omega} s_{\kappa_1,\kappa_2} \tag{14}$$

Optimizing the first and second terms of (13) helps ensure that an optimal solution is located in a feasibility region with sufficient historical data. Optimizing the third term helps ensure that an optimal solution is located in a feasibility region with better model accuracy. Put together, they strike a balance between reducing error on the one hand and capturing more data-populated feasible space on the other.

As an example, consider the noisy dataset in Figure (2a) associated with the function $y = x$. The derived SD $\sigma_{GP_{f|\mathcal{D}}}$ is given in Figure (2b). In this 1-dimensional example, feasibility regions are given by $\Omega = [x, y] \subseteq \mathbb{R}$. The resulting function $s_{\kappa_1,\kappa_2} : \{(x, y) \in \mathbb{R}^2 \,|\, x < y\} \to \mathbb{R}$ is plotted in Figure (3) for $\kappa_1 = 0.05 = \kappa_2$.

An important feature of s_{κ_1,κ_2} observed in Figure (3) is the presence of a single local (as well as global) minimum, turning (14) into a convex optimization problem. The global minimum in this example is achieved at $\Omega^* = [-0.475, 0.956]$, which is intuitive considering Figure (2b).

Further investigation of s_{κ_1,κ_2} (13) and the optimization problem posed in (14) shows promise, though not without its own challenges. Of primary concern is the choice of κ_1 and κ_2. The first and second terms in (13) need to be balanced against the third term with a suitable choice of κ_i. Otherwise, the concern for data sufficiency may numerically overwhelm the concern for model fidelity or vice versa. Nevertheless, solving the (possibly convex) optimization problem in Definition 1, suggests a way to augment a predict-then-optimize scheme by "learning" the optimal feasibility region in tandem.



(a) Noisy dataset for $y = x$.

(b) $\sigma_{GP_{f|\mathcal{D}}}$.

4.2 Iterative improvement approach

Our approach for Gaussian Process Based Problem Augmentation encapsulates two concepts which could be extremely helpful in providing high quality solutions: *Data sufficiency*, which means we should rely on the optimization model only if the recommendations are close to regions in which there is sufficient data, and *model fidelity*, which means that the learnt optimization models are accurate enough. As described, r_1 and r_2 address data fidelity, and the integral represents data sufficiency.

In this section we therefore sketch an additional approach for high quality solutions based on these concepts for cases in which the assumptions required for the Gaussian Process based augmentation do not hold. We do this by first explicitly defining regions of the problem in which both data sufficiency and model fidelity hold.

To define such regions in the $\mathcal{X} \times \mathcal{P}$, we would need some notion of "distance" between pairs of points from each of $\mathcal{X}$ and $\mathcal{P}$. In one setup, we can consider distance functions $d^X : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ and $d^P : \mathcal{P} \times \mathcal{P} \to \mathbb{R}_+$ for $\mathcal{X}$ and $\mathcal{P}$ respectively. These distances can be given (from domain knowledge) or learned in some way. To define regions of

[12] As in Section 3.1, we will continue to assume all covariates are decision variables.

[13] As in section 3, feasibility regions are defined via a finite set of affine inequalities $\{g_k(x) \leq 0\}$ with the number of inequalities globally bounded from above.

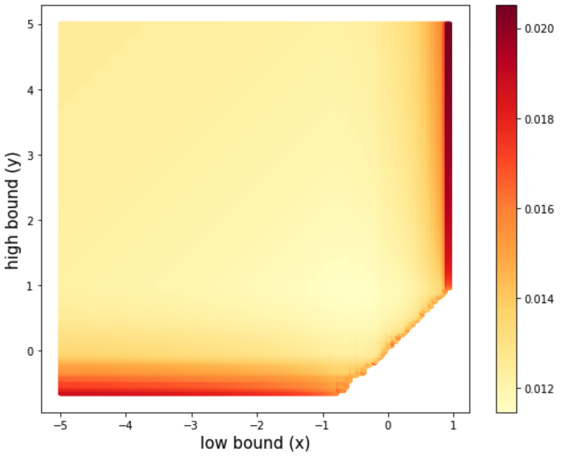


Figure 3: s_{κ_1,κ_2}, for $\kappa_1 = \kappa_2 = 0.05$

sufficient data and model fidelity, we have historical data $\mathcal{D}$ in the form of tuples $d_i = \langle x_i, p_i, z_i \rangle \in \mathcal{D}$ such that $x_i \in \mathcal{X}, p_i \in \mathcal{P}$ with $x_i \in \Omega(p_i), z_i = f(x_i, p_i)$.

Given the notions of distance, thresholds $\delta_P > 0, \delta_X > 0$ and some historical data $\mathcal{D}$, we can define "close-by" uncontrollables p_i for a given uncontrollable $p \in \mathcal{P}$ under consideration (for which we are solving problem (1)) as

$$S(p) = \{\langle x_i, p_i, z_i \rangle \in \mathcal{D} : d^P(p, p_i) \leq \delta_P\}. \tag{15}$$

Then for the current problem (1) with uncontrollable p, with thresholds $\epsilon_S \in (0,1)$, we consider a decision $x \in \mathcal{X}$ to be in a *"region of sufficient data"* if

$$\begin{aligned} &|Q(x,p|/|\mathcal{D}| \geq \epsilon_S, \text{ where} \\ &Q(x,p) = \left\{\langle x_i, p_i, z_i \rangle \in S(p), d^X(x, x_i) \leq \delta_X\right\}. \end{aligned} \tag{16}$$

To define regions of sufficient model fidelity, we first have to define some "models". We first think of a regression model $\widehat{f} : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$, which could be possibly learned using examples of (x, p, z) tuples. Second, we can learn a classifier $\widehat{g} : \mathcal{X} \times \mathcal{P} \to [0,1]$ that tries to map a (x, p) to $\mathbb{I}(x \in \Omega(p))$ where $\widehat{g}(x,p) \approx \Pr(x \in \Omega(p))$. Then for problem (1) with current uncontrollable p, given the above models, and fidelity thresholds $\epsilon_M^f > 0, \epsilon_M^g \in (0,1)$, we consider a decision $x \in \mathcal{X}$ to be in a *"region of sufficient model fidelity"* with respect to $\widehat{f}$ and $\widehat{g}$ if

$$\begin{aligned} &\sigma(\{|z_i - \widehat{f}(x_i, p_i)| : (x_i, p_i, z_i) \in Q(x,p)\}) \leq \epsilon_M^f, \\ &\sigma(\{1 - \widehat{g}(x_i, p_i) : (x_i, p_i, z_i) \in Q(x,p)\}) \leq \epsilon_M^g, \end{aligned} \tag{17}$$

where $\sigma(\{\cdots\})$ is some aggregator over a set of scalars such as mean, median, max (or some percentile in between). For better generalizability, it would be better to determine the regions for model fidelity with respect to some historical data D which is **not** used for the learning of the models $\widehat{f}$ and $\widehat{g}$.

Then the optimization problem for any given uncontrollable $p \in \mathcal{P}$ we solve could be written as:

$$\min_{x \in \mathcal{X}, \widehat{f}, \widehat{g}} \widehat{f}(x,p) \text{ s. t. } \begin{cases} \widehat{g}(x,p) \geq 1 - \epsilon_C, \\ \text{Data sufficiency cst. (16),} \\ \text{Model fidelity cst. (17),} \end{cases} \tag{18}$$

where $\epsilon_C \in (0,1)$. Let $\widehat{x}^\star(p)$ be a solution of such a problem for a given uncontrollable $p \in \mathcal{P}$.

Motivation for proposed constraints. For the current uncontrollable p under consideration, let $S(p)$ be the set of "nearby" uncontrollables in the historical data $\mathcal{D}$ as defined in (15). Let $X(p)$ defined as following be the set of "historical decisions for all nearby uncontrollables":

$$X(p) = \{x_i : \langle x_i, p_i, z_i \rangle \in S(p)\}. \tag{19}$$

Assuming that $X(p) \subset \Omega(p)$, consider that we have a surrogate $\widehat{f} : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$ for the true objective $f : \mathcal{X} \times \mathcal{P}$. Let us further make the following two assumptions:

(A1) High fidelity surrogate for nearby data. $\forall (x_i, p_i, z_i) \in S(p)$, such that $x_i \in X(p)$, we have $|z_i - \widehat{f}(x_i, p_i)| \leq \epsilon_f$.

(A2) Guidance from historical decisions. We assume that, while the historical decisions are not optimal (i. e., $x_i \neq x^\star(p_i)$ for any $(x_i, p_i, z_i) \in \mathcal{D}$), nearby decisions provide "guidance" by ensuring that the optimal decision $x^\star(p)$ for current uncontrollable p lies in the convex hull of these nearby decisions, that is $x^\star(p) \in \mathsf{Conv}(X(p))$.

Then our following result demonstrates how the aforementioned data sufficiency (16) and model fidelity constraints (17) allow us to bound the optimality gap of the selected decision (the proof is in Appendix **??**):

Theorem 2 (Bounded optimality gap)**.** *Under the conditions of problem* (1)*, for any uncontrollable $p \in \mathcal{P}$, assume that A1 and A2 holds for some $\delta_P > 0$ and $\epsilon_f > 0$. Let the functions $f(x, \cdot)$ be L_f^P-Lipschitz and $\widehat{f}(x, \cdot)$ be $L_{\widehat{f}}^P$-Lipschitz for any $x \in \mathcal{X}$. Also let the functions $f(\cdot, p)$ be L_f^X-Lipschitz and $\widehat{f}(\cdot, p)$ be $L_{\widehat{f}}^X$-Lipschitz for the current p. Let $x^\star(p)$ be the optimal decision and $\widehat{x}^\star$ be the decision selected using the surrogate $\widehat{f}$ as*

$$\widehat{x}^\star := \arg \min_{x \in \mathsf{Conv}(X(p))} \widehat{f}(x,p). \tag{20}$$

Then, the gap between the objective for the selected decision $\widehat{x}^\star$ and the optimal decision $x^\star(p)$ is bounded as:

$$\begin{aligned} &f(\widehat{x}^\star, p) - f(x^\star(p), p) \leq \\ &\quad 2\left(\delta_P\left(L_f^P + L_{\widehat{f}}^P\right) + \epsilon_f + \left(L_f^X + L_{\widehat{f}}^X\right)\Delta(p)\right), \end{aligned} \tag{21}$$

where $\Delta(p) := \min_{x_i \in X(p)} \max_{x_j \in X(p)} \|x_i - x_j\|_2$.

Our proposed model fidelity constraint (17) ensures A1, and implies that a tighter model fidelity constraint (a smaller ϵ_f) ensures a smaller optimality gap bound in (21). A2 implies that historical decisions are somewhat informative of optimal decisions for nearby uncontrollables. While this may not be true in general, larger set of nearby uncontrollables can make the convex hulls larger, boosting the chance of including the optimal decision. Constraint (16) ensures that we are in such regions, thereby promoting A2.

The bound (21) does highlight some intuitive tradeoffs in this decision optimization problem. The term $\Delta(p)$ is related to the "nearness" threshold δ_X in (16), and smaller

Algorithm 1: Iterative meta-algorithm for uncontrollable p.

```
**Input**: Initial values for $\epsilon_S, \epsilon_M^f, \epsilon_M^g, \epsilon_C > 0$, change threshold upper bound $\overline{\alpha}$.
**Output**: Decision $\widehat{x}^\star$
Initialize $\alpha \leftarrow \infty, \rho_{\text{old}} \leftarrow 0, \rho_{\text{best}} \leftarrow 0$.
while $\alpha \leq \overline{\alpha}$ do
    Solve (18) for current $\epsilon_S, \epsilon_M^f, \epsilon_M^g, \epsilon_C$ to get $\widehat{x}^\star(p)$.
    Compute $\rho$ as in (22) using the heldout set $\mathcal{V}$.
    $\alpha \leftarrow |\rho - \rho_{\text{old}}|, \rho_{\text{old}} \leftarrow \rho$.
    if $\rho > \rho_{\text{best}}$ then
        $\rho_{\text{best}} \leftarrow \rho$.
    end if
    Increase values of $\epsilon_S, \epsilon_M^f, \epsilon_M^g, \epsilon_C$.
end while
return Decision $\widehat{x}^\star(p)$ corresponding to $\rho_{\text{best}}$.
```

δ_P and δ_X imply smaller optimality gap bound. However, smaller values make the satisfaction of the data sufficiency constraints (and hence A2) more challenging. We do not control the Lipschitz constants L_f^P, L_f^X of the unknown objective f, but we can control the Lipschitz constants $L_{\widehat{f}}^P, L_{\widehat{f}}^X$ of the surrogate $\widehat{f}$. However, a surrogate with a small Lipschitz constant (such as a linear $\widehat{f}$) along with a small δ_P, δ_X would probably not be able to satisfy the model fidelity constraint (17) as easily.

4.3 Estimating quality and iterative improvement

Consider a heldout set of historical data denoted by $\mathcal{V}$. Using the previously proposed probability of improvement based quality estimate, we assess the quality of a particular ***solution technique*** that produces a decision $\widehat{x}^\star(p)$ for any given uncontrollable p as

$$\rho = \sum_{(x,p,z) \in \mathcal{V}} \Pr\left(f(\widehat{x}^\star(p), p) < z\right). \tag{22}$$

Here ρ is an aggregation of the probability of improvements over multiple instances of the problem (18) for different settings of uncontrollables p such that $(x, p, z) \in \mathcal{V}$.

Using the previously presented tools, we present an *iterative meta-algorithm* with any solution technique for any uncontrollable $p \in \mathcal{P}$ in Algorithm 1. This meta-algorithm can employ any scheme to solve multiple instances of (18) and return the decision with the highest aggregated probability of improvement on the heldout historical data.

5 Related Work

There are existing works on creating optimization models from data. For example, Arcangioli, Bessiere, and Lazaar (2016) and De Raedt, Passerini, and Teso (2018) focus on learning constraints from data. In Lombardi, Milano, and Bartolini (2017), a machine learning model of a complex system is generated from data, and then transformed into an optimization model. The work in Subramanian et al. (2019) learns machine learning models for multiple components of a complex industrial process, and incorporates knowledge specification in the form of intermediate storage nodes, from which a process-wide optimization model is generated. However, while these works address automatic derivation of optimization models, they do not address the resulting solution quality,.

Works such as Wilder, Dilkina, and Tambe (2019) and Elmachtoub and Grigas (2020) address the issue of integrating machine learning and optimization models using decision-focused learning. Indeed, the loss function for the machine learning model is defined to account for the optimization objective. However, the problem addressed by these works are significantly different than the problem described in Section 2: it assumes that the structure of the functions describing the objectives and constraints are given, and that there is uncertainty only regarding the covariates of the objective function. Moreover, these works do not enable accounting for the influence the decision variables have on the machine learning models' features, and therefore, do not attempt to address historical decision data.

Another well known optimization approach is *Reinforcement Learning*, or RL (Sutton and Barto 2018). which offers some ideas that have direct relevance to this work: Thomas, Theocharous, and Ghavamzadeh (2015) defines a problem similar in spirit to Equation (3) in the sense that a policy is sought that improves, with a high enough probability, the objective as compared to the objective obtained by the policy used to generate the data used for RL. For offline RL, Yu et al. (2020) learn an environment model is learned from historical data, which is then used as input to the RL algorithm, and the uncertainty of the environment model is explicitly taken into account in the reward function. Much like offline RL, we have historical data, and the uncontrollable p can be thought of as state variables, and the decisions x as the actions. However, unlike offline RL, p is truly uncontrollable, and our "action" at any state (current p) does not have any control over the next state (or next p). Finally, in RL, we seek to optimize for the cumulative reward over a sequence of actions/decisions, while in our case, we seek an improved decision (Equation (3)) for each state (uncontrollable p).

6 Summary and Future Work

In this paper we study the solution quality for the decision-making problem when the optimization models are learnt from historical data, addressing the challenging issue related to the uncertainty of such machine learning models when the data includes historical decisions. We provided a formal definition of optimization solution quality and provided an accurate estimate of the probability of improvement using using Gaussian Processes. We also proposed an approach to including additional constraints based on the notions of data sufficiency and model fidelity in order to improve the solution quality. To the best of our knowledge, this is the first work to quantify solution quality for such generated optimization model with theoretical guarantee. For future work, we intend to enhance the theoretical basis to address this problem, implement an end-to-end framework ensuring the solution quality, and validating it via numerical experiments.

References

Arcangioli, R.; Bessiere, C.; and Lazaar, N. 2016. Multiple Constraint Acquisition. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, 698–704. AAAI Press. ISBN 9781577357704.

Davidovich, O. 2021. The Combinatorial Generation of Piece-wise Linear Functions. *arXiv preprint TBD*.

De Raedt, L.; Passerini, A.; and Teso, S. 2018. Learning constraints from examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Elmachtoub, A. N.; and Grigas, P. 2020. Smart "Predict, then Optimize". arXiv:1710.08005.

Gheorghe-Teodor Bercea, Olivier Tardieu. 2021. rayvens: Event sources and sinks on Ray. https://github.com/project-codeflare/rayvens.

GPy. 2012. GPy: A Gaussian process framework in python. http://github.com/SheffieldML/GPy.

Lombardi, M.; Milano, M.; and Bartolini, A. 2017. Empirical decision model learning. *Artificial Intelligence*, 244: 343–367. Combining Constraint Solving with Mining and Learning.

Mitchell, S.; O'Sullivan, M.; and Dunning, I. 2011. PuLP: A Linear Programming Toolkit for Python.

Moritz, P.; Nishihara, R.; Wang, S.; Tumanov, A.; Liaw, R.; Liang, E.; Elibol, M.; Yang, Z.; Paul, W.; Jordan, M. I.; and Stoica, I. 2018. Ray: A Distributed Framework for Emerging AI Applications. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, 561–577. Carlsbad, CA: USENIX Association. ISBN 978-1-939133-08-3.

Phan, D.; Nguyen, L.; Murali, P.; Pham, N.; Liu, H.; and Kalagnanam, J. 2021. Regression Optimization for System-level Production Control. In *American Control Conference*, 5023–5028.

Subramanian, D.; Murali, P.; Zhou, N.; Ma, X.; Da Silva, G. C.; Pavuluri, R.; and Kalagnanam, J. 2019. A prediction-optimization framework for site-wide process optimization. In *2019 IEEE International Congress on Internet of Things (ICIOT)*, 125–132. IEEE.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book. ISBN 0262039249.

Thomas, P.; Theocharous, G.; and Ghavamzadeh, M. 2015. High Confidence Policy Improvement. In Bach, F.; and Blei, D., eds., *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 2380–2388. Lille, France: PMLR.

Wilder, B.; Dilkina, B.; and Tambe, M. 2019. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1658–1665.

Williams, C. K.; and Rasmussen, C. E. 2006. *Gaussian Processes for Machine Learning*. MIT Press Cambridge, MA.

Yu, T.; Thomas, G.; Yu, L.; Ermon, S.; Zou, J. Y.; Levine, S.; Finn, C.; and Ma, T. 2020. MOPO: Model-based Offline Policy Optimization. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 14129–14142. Curran Associates, Inc.