Mixed Strategies for Robust Optimization of Unknown Objectives

Pier Giuseppe Sessa ETH Zürich
Ilija Bogunovic ETH Zürich
Maryam Kamgarpour ETH Zürich
Andreas Krause ETH Zürich

Abstract

We consider robust optimization problems, where the goal is to optimize an *unknown* objective function against the worst-case realization of an uncertain parameter. For this setting, we design a novel sample-efficient algorithm GP-MRO, which sequentially learns about the unknown objective from noisy point evaluations. GP-MRO seeks to discover a robust and randomized *mixed strategy*, that maximizes the worst-case expected objective value. To achieve this, it combines techniques from online learning with nonparametric confidence bounds from Gaussian processes. Our theoretical results characterize the number of samples required by GP-MRO to discover a robust near-optimal mixed strategy for different GP kernels of interest. We experimentally demonstrate the performance of our algorithm on synthetic datasets and on human-assisted trajectory planning tasks for autonomous vehicles. In our simulations, we show that robust deterministic strategies can be overly conservative, while the mixed strategies found by GP-MRO significantly improve the overall performance.

1 Introduction

Many real-world problems require taking decisions under uncertainty. Latter can manifest itself in the form of uncertain parameters, perturbations, or an adversary that can corrupt the decision (Bertsimas et al., 2011). In such problems, one often seeks to optimize an objective function while being *robust* to the worst possible uncertainty realization. This can be achieved by phrasing such problems in the framework of Robust Optimization (RO) (Ben-Tal et al., 2009). RO has found successful applications in various domains including supply chain management (Bertsimas and Thiele, 2004), portfolio optimization (Ben-Tal et al., 2000), influence maximization (He and Kempe, 2016), and robotics (Jørgensen et al., 2018), to name a few.

In various practical problems, however, the objective function to be optimized is a-priori *unknown*, and one can only learn about it from *sequential* and *noisy* point evaluations. Gaussian process (GP) optimization is an established framework for model-based sequential optimization of such unknown functions (Srinivas et al., 2010). An array of algorithms that use Bayesian non-parametric GP models (Rasmussen and Williams, 2006), and balance exploration (learning the function globally) and exploitation (maximizing the function) have been developed over the years, e.g., (Srinivas et al., 2010; Bogunovic et al., 2016b; Chowdhury and Gopalan, 2017; Wang and Jegelka, 2017; Frazier, 2018).

In this paper, we study the *robust* optimization problem where (i) the objective function is *unknown* and (ii) the goal is to be robust against the worst possible realization of its *uncertain parameter*. This problem differs from the classical RO formulation where the objective function is assumed to be known, and is also different from the standard GP optimization where robustness requirement is typically not pursued.

Instead of finding a robust deterministic solution to this problem (as in (Bogunovic et al., 2018)), we seek to discover a randomized, i.e., *mixed* strategy, from a relatively small number of noisy function evaluations. The primary motivation for seeking such strategies is that, in general, they can provide arbitrarily better worst-case expected performance than deterministic ones (Krause et al., 2011; Vorobeychik and Li, 2014; Sinha et al., 2018), i.e., randomization prevents a potential adversary to know the actual decision until it is realized. Consequently, we design and use a novel GP-based *sample efficient* algorithm to discover near-optimal mixed strategies. We empirically demonstrate the effectiveness of the identified robust

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

mixed strategies in a trajectory planning task for autonomous vehicles, where deterministic strategies are shown to be overly conservative.

Related work. Over the past couple of years, robust optimization has been extensively studied in the machine learning community. While most of the works focus on convex settings (e.g., (Shalev-Shwartz and Wexler, 2016; Namkoong and Duchi, 2016)), more recent works also consider general non-convex objectives, e.g., (Chen et al., 2017; Sinha et al., 2017; Staib et al., 2018). Among those, Chen et al. (2017) provide robust algorithmic strategies that are shown to be successful in several learning tasks. The proposed algorithm is based on the idea of simulating a zero-sum game between a learner and an adversary. Similar strategies have been also explored in other adversarial settings, e.g., in submodular optimization (Krause et al., 2011; Kawase and Sumita, 2019). Our approach is based on the similar algorithmic idea of Chen et al. (2017), but unlike this and other works mentioned above that assume the objective function is perfectly *known* (or a maximization oracle is available), it also requires performing a non-trivial function estimation.

In non-robust GP optimization, various optimization algorithms (Srinivas et al., 2010; Chowdhury and Gopalan, 2017; Bogunovic et al., 2016b; Contal et al., 2013; Wang and Jegelka, 2017) have been proposed to sequentially optimize the unknown function from noisy and zeroth-order observations. Similarly to these algorithms, our algorithm relies on a non-parametric GP model to obtain shrinking confidence bounds around the unknown objective function. Besides the standard problem, GP optimization has been considered in several other practical settings such as contextual (Krause and Ong, 2011), time-varying (Bogunovic et al., 2016a), safe exploration (Sui et al., 2015), etc.

Recently, a novel algorithm for *robust* GP optimization StableOpt has been proposed by Bogunovic et al. (2018). StableOpt discovers a *deterministic* solution that is robust with respect to the worst-case realization of the uncertain parameter. This work is closest to ours, but instead of seeking deterministic solutions, our focus is on the *mixed strategies* which are preferable in certain scenarios (see Section 4.2), where deterministic solutions turn out to be overly conservative. We also note that other forms of robustness have been studied in GP optimization. For instance, Nogueira et al. (2016); Oliveira et al. (2019) consider robustness against uncertain inputs (typical in robotics applications), Sessa et al. (2019) study robust aspects in multi-agent unknown repeated games, Williams et al. (2000); Tesch et al. (2011) deal with uncontrolled environmental variables, while robustness with respect to outliers is addressed by Martinez-Cantin et al. (2018).

Contributions. We consider robust optimization of *unknown* and generally non-convex objectives.

- We propose an algorithm, GP-MRO, which returns a *mixed strategy*, i.e., a probability distribution over actions, that is robust against the worst-case realization of the *uncertain* parameter.
- Our theoretical analysis shows the number of samples required for GP-MRO to discover a near-optimal robust mixed strategy.
- We propose a variant of GP-MRO which can effectively trade-off worst-case and average-case performance.
- Finally, we consider the problem of trajectory planning in autonomous driving guided by user's evaluations. In our experiments, we demonstrate the effectiveness of the robust mixed strategies discovered by GP-MRO in comparison to those identified by existing robust methods.

2 Problem Formulation

Let $f : \mathcal{X} \times \Theta \to [0, 1]$ be a reward function over domain $D = \mathcal{X} \times \Theta$, where $\mathcal{X}$ is a continuous and compact decision set and $\Theta = \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_m\}$ is a finite set of parameter values. The reward function is *unknown*, and we learn about it from sequential *noisy* point observations, i.e., so-called *bandit* feedback. At each time step t, we choose $\boldsymbol{x}_t \in \mathcal{X}$ and $\boldsymbol{\theta}_t \in \Theta$, and observe a noisy sample $y_t = f(\boldsymbol{x}_t, \boldsymbol{\theta}_t) + \xi_t$, where $\xi_t \sim \mathcal{N}(0, \sigma^2)$, and ξ_t's are independent over time (our approach allows also for sub-Gaussian noise).

After T rounds (i.e., T samples), our goal is to report a strategy for selecting points in $\mathcal{X}$ that is robust against the worst-possible parameter value from Θ. We assume that during the optimization phase (i.e., training/simulation) one can choose $\boldsymbol{\theta}$, while later, during the implementation (i.e., test) phase, the parameter $\boldsymbol{\theta}$ becomes uncontrollable. Hence, it is important to design a robust strategy for selecting the first parameter.

Optimization goal. Let $\Delta(\mathcal{X})$ denote the set of all probability distributions, or *mixed strategies* on $\mathcal{X}$. Our goal is to find a distribution in $\Delta(\mathcal{X})$ that achieves high reward in the worst-case over $\boldsymbol{\theta} \in \Theta$. The *maximin* optimal value is given by:

$$\tau^* = \max_{\mathcal{P} \in \Delta(\mathcal{X})} \min_{\boldsymbol{\theta} \in \Theta} \mathbb{E}_{\boldsymbol{x} \sim \mathcal{P}}[f(\boldsymbol{x}, \boldsymbol{\theta})], \tag{1}$$

and we seek to report a robust solution $\mathcal{P}^{(T)} \in \Delta(\mathcal{X})$ that for some specified accuracy value $\epsilon \geq 0$ achieves

$$\min_{\boldsymbol{\theta} \in \Theta} \mathbb{E}_{\boldsymbol{x} \sim \mathcal{P}^{(T)}}[f(\boldsymbol{x}, \boldsymbol{\theta})] \geq \tau^* - \epsilon. \tag{2}$$

Besides achieving (2), our goal is also to minimize the total number of required samples T.

We note that our optimization goal is different from the one of computing *deterministic* (pure

strategy) solution $\boldsymbol{x} \in \mathcal{X}$ and competing against $\tau = \max_{\boldsymbol{x}\in\mathcal{X}} \min_{\boldsymbol{\theta}\in\Theta} f(\boldsymbol{x},\boldsymbol{\theta})$ as considered in (Bogunovic et al., 2018). Our goal is to discover a *randomized* strategy and compete against $\tau^* \geq \tau$, which can be arbitrarily larger than τ. Hence, mixed strategies considered in this work can provide arbitrarily better expected performance than such deterministic ones. Conceptually, randomization allows the decisions to be less predictable, and is a key feature necessary in many applications including security games (Sinha et al., 2018), adversarial learning (Vorobeychik and Li, 2014) and sensing (Krause et al., 2011). This is also the case in the autonomous driving scenario considered in Section 4.2, where we show that deterministic strategies can be overly conservative. Finally, we also note that the same objective (1) is considered in (Chen et al., 2017), in the case of *known* reward functions $f_i(\cdot) := f(\cdot,\boldsymbol{\theta}_i)$, and $i \in \{1,\dots,m\}$.

Our Model. We assume that the unknown objective f is fixed and belongs to a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_k(D)$ corresponding to a positive semi-definite kernel function $k(\cdot,\cdot) : D \times D \to \mathbb{R}$. Furthermore, we require f to have a bounded RKHS norm, i.e., $\|f\|_k = \sqrt{\langle f, f\rangle_k} \leq B$ where $\|\cdot\|_k$ stands for the RKHS norm and B is a known positive constant. The RKHS norm represents a measure of smoothness of f as measured by the corresponding kernel. We note that these are the standard assumptions used in GP optimization (see, e.g., (Srinivas et al., 2010; Chowdhury and Gopalan, 2017; Bogunovic et al., 2018)).

For the kernel function, we assume $k\big((\boldsymbol{x},\boldsymbol{\theta}),(\boldsymbol{x},\boldsymbol{\theta})\big) \leq 1$ for all $(\boldsymbol{x},\boldsymbol{\theta}) \in D$, which is without loss of generality if appropriate re-scaling is applied. Our setup also allows for composite kernels that can be constructed by using individual kernels $k_1 : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ and $k_2 : \Theta\times\Theta \to \mathbb{R}$, to obtain, for example, *additive kernel* $k\big((\boldsymbol{x},\boldsymbol{\theta}),(\boldsymbol{x}',\boldsymbol{\theta}')\big) := k_1(\boldsymbol{x},\boldsymbol{x}') + k_2(\boldsymbol{\theta},\boldsymbol{\theta}')$ or *product kernel* $k\big((\boldsymbol{x},\boldsymbol{\theta}),(\boldsymbol{x}',\boldsymbol{\theta}')\big) := k_1(\boldsymbol{x},\boldsymbol{x}')\cdot k_2(\boldsymbol{\theta},\boldsymbol{\theta}')$. Popularly used kernels are linear, squared exponential (SE) and Matérn:

$$k_{\text{Lin}}(\boldsymbol{x},\boldsymbol{x}') = \boldsymbol{x}^T\boldsymbol{x}',$$
$$k_{\text{SE}}(\boldsymbol{x},\boldsymbol{x}') = \exp\left(-\tfrac{1}{2l^2}\|\boldsymbol{x}-\boldsymbol{x}'\|^2\right), \text{and}$$
$$k_{\text{Mat}}(\boldsymbol{x},\boldsymbol{x}') = \tfrac{2^{1-\nu}}{\Gamma(\nu)}\left(\tfrac{\sqrt{2\nu}\|\boldsymbol{x}-\boldsymbol{x}'\|}{l}\right) J_\nu\left(\tfrac{\sqrt{2\nu}\|\boldsymbol{x}-\boldsymbol{x}'\|}{l}\right),$$

where l is the length-scale parameter and $\nu > 0$ is a parameter that determines the smoothness (Rasmussen and Williams, 2006).

Under such assumptions, the uncertainty over f is naturally modeled as a Gaussian process $\text{GP}(0, k((\boldsymbol{x},\boldsymbol{\theta}),(\boldsymbol{x}',\boldsymbol{\theta}')))$. Further on, a Gaussian likelihood model for the observations can be used assuming the noise $\xi_t = y_t - f(\boldsymbol{x}_t,\boldsymbol{\theta}_t)$ is drawn, independently across t, from $\mathcal{N}(0,\lambda)$. Here, λ denotes a free hyper-parameter that may differ from the true noise variance σ^2. With this model in place, conditioned on the history of inputs $\{(\boldsymbol{x}_1,\boldsymbol{\theta}_1),\dots,(\boldsymbol{x}_t,\boldsymbol{\theta}_t)\}$ and their noisy observations $\{y_1,\dots,y_t\}$, the posterior distribution under this prior is also Gaussian with the closed form posterior mean and variance:

$$\mu_t(\boldsymbol{x},\boldsymbol{\theta}) = \boldsymbol{k}_t(\boldsymbol{x},\boldsymbol{\theta})^T\big(\boldsymbol{K}_t + \lambda\mathbf{I}_t\big)^{-1}\boldsymbol{y}_t, \tag{3}$$
$$\sigma_t^2(\boldsymbol{x},\boldsymbol{\theta}) = k((\boldsymbol{x},\boldsymbol{\theta}),(\boldsymbol{x},\boldsymbol{\theta})) - \boldsymbol{k}_t(\boldsymbol{x},\boldsymbol{\theta})^T\big(\boldsymbol{K}_t + \lambda\mathbf{I}_t\big)^{-1}\boldsymbol{k}_t(\boldsymbol{x},\boldsymbol{\theta}), \tag{4}$$

s.t. $\boldsymbol{k}_t(\boldsymbol{x},\boldsymbol{\theta}) = \big[k\big((\boldsymbol{x}_j,\boldsymbol{\theta}_j),(\boldsymbol{x},\boldsymbol{\theta})\big)\big]_{j=1}^t$, and $\boldsymbol{K}_t = \big[k\big((\boldsymbol{x}_j,\boldsymbol{\theta}_j),(\boldsymbol{x}_{j'},\boldsymbol{\theta}_{j'})\big)\big]_{j,j'}$ is the kernel matrix. As described bellow, we make use of this model in our algorithm to sequentially learn about the unknown objective function.

3 Proposed Algorithm and Theory

Our algorithm, GP-MRO, is shown in Algorithm 1. It can be interpreted as a zero-sum game between a simulated adversary and a learner. The adversary plays actions from the set Θ, while the learner plays actions from $\mathcal{X}$. Because the true reward function $f(\cdot,\cdot)$ is unknown, the algorithm maintains and makes use of the optimistic upper confidence bound $\overline{\text{ucb}}_t(\cdot,\cdot)$ (defined below) of the unknown reward function. We define the confidence bounds as follows:

$$\text{ucb}_t(\boldsymbol{x},\boldsymbol{\theta}) := \mu_t(\boldsymbol{x},\boldsymbol{\theta}) + \beta_{t+1}\sigma_t(\boldsymbol{x},\boldsymbol{\theta}) \tag{5}$$
$$\text{lcb}_t(\boldsymbol{x},\boldsymbol{\theta}) := \mu_t(\boldsymbol{x},\boldsymbol{\theta}) - \beta_{t+1}\sigma_t(\boldsymbol{x},\boldsymbol{\theta}), \tag{6}$$

where β_t is the *confidence parameter* that we set according to Lemma 1 bellow. We also define their truncated versions:

$$\overline{\text{ucb}}_t(\boldsymbol{x},\boldsymbol{\theta}) := \min\{\text{ucb}_t(\boldsymbol{x},\boldsymbol{\theta}), 1\} \tag{7}$$
$$\overline{\text{lcb}}_t(\boldsymbol{x},\boldsymbol{\theta}) := \max\{\text{lcb}_t(\boldsymbol{x},\boldsymbol{\theta}), 0\}, \tag{8}$$

which we use in our algorithm. At every round t, GP-MRO simulates the adversary by selecting a distribution over the m values of $\boldsymbol{\theta}$, i.e., $\boldsymbol{w}_t \in \{\boldsymbol{w} \in [0,1]^m : \sum_{i=1}^m \boldsymbol{w}[i] = 1\}$, where $\boldsymbol{w}_t[i]$ denotes the probability of selecting $\boldsymbol{\theta}_i$. Subsequently, the learner *best responds* by selecting $\boldsymbol{x}_t$ based on the knowledge of $\boldsymbol{w}_t$. After T iterations, GP-MRO returns the uniform distribution over $\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_T\}$, denoted with $\mathcal{U}^{(T)}$. Next, we explain how $\boldsymbol{w}_t$ and $\boldsymbol{x}_t$ are chosen in Algorithm 1.

The multiplicative weight updates (MWU) rule (Freund and Schapire, 1997) is used to select $\boldsymbol{w}_t$ at every round t. We note that this algorithm is an online learning *no-regret* algorithm that requires *full-information* feedback at every round, i.e., observations that correspond to every pair $\{(\boldsymbol{x}_t,\boldsymbol{\theta}_i)\}_{i=1}^m$. This is not possible in our setting where the learner only receives a single noisy observation that corresponds to the chosen pair $(\boldsymbol{x}_t,\boldsymbol{\theta}_t)$. To cope with this, we make use of the up-

per confidence bound functions to effectively emulate the full information feedback.[1] Hence, the MWU rule used in our algorithm is given by:

$$\boldsymbol{w}_t[i] \propto \exp\Big\{ -\eta_T \sum_{j=1}^{t-1} \overline{\text{ucb}}_{j-1}(\boldsymbol{x}_j, \boldsymbol{\theta}_i)\Big\},$$

where η_T is the learning rate parameter that we set in Theorem 2 bellow. Another equivalent way of writing this rule is via the following recursive update:

$$\boldsymbol{w}_t[i] = \frac{\boldsymbol{w}_{t-1}[i] \cdot \exp\big(-\eta_T \cdot \overline{\text{ucb}}_{t-1}(\boldsymbol{x}_{t-1}, \boldsymbol{\theta}_i)\big)}{\sum_{j=1}^m \boldsymbol{w}_{t-1}[j] \cdot \exp(-\eta_T \cdot \overline{\text{ucb}}_{t-1}(\boldsymbol{x}_{t-1}, \boldsymbol{\theta}_j))}\ .$$

The learner then observes $\boldsymbol{w}_t$, and plays the best response $\boldsymbol{x}_t$ that is obtained by using the upper confidence bound instead of the true unknown function:

$$\boldsymbol{x}_t = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \left(\sum_{i=1}^m \boldsymbol{w}_t[i] \cdot \overline{\text{ucb}}_{t-1}(\boldsymbol{x}, \boldsymbol{\theta}_i)\right). \quad (9)$$

Finally, the unknown function is queried at $(\boldsymbol{x}_t, \boldsymbol{\theta}_t)$, where $\boldsymbol{\theta}_t$ is selected as the parameter value that has the highest uncertainty for the selected $\boldsymbol{x}_t$, i.e.,

$$\boldsymbol{\theta}_t \in \arg\max_{\boldsymbol{\theta}\in\Theta} \sigma_{t-1}(\boldsymbol{x}_t, \boldsymbol{\theta}). \quad (10)$$

The observed data $(\boldsymbol{x}_t, \boldsymbol{\theta}_t, y_t)$ is then used to update the model via (3) and (4).

3.1 Main result

To characterize our regret bounds, we make use of a suitable measure of complexity of the function class, the so-called *maximum information gain*. It has been introduced by Srinivas et al. (2010), and subsequently used in many different works on Bayesian (GP) optimization. At time t, it is defined as

$$\gamma_t = \max_{\{(\boldsymbol{x}_1,\boldsymbol{\theta}_1),\ldots,(\boldsymbol{x}_t,\boldsymbol{\theta}_t)\}} \frac{1}{2}\log\det(\mathbf{I}_t + \lambda^{-1}\boldsymbol{K}_t), \quad (11)$$

and is used to measure the reduction in uncertainty about f after receiving t noisy observations that correspond to $\{(\boldsymbol{x}_1, \boldsymbol{\theta}_1), \ldots, (\boldsymbol{x}_t, \boldsymbol{\theta}_t)\}$. In the case $D \subset \mathbb{R}^d$, this kernel-dependent quantity is sublinear in t for various kernel functions, e.g., $\mathcal{O}((\log t)^{d+1})$ for squared exponential and $\mathcal{O}(t^{(d+1)d/((d+1)d+2\nu)}\log t)$ for the Matérn kernel with $\nu > 1$ (Srinivas et al., 2010).

We use the following well-known result in GP optimization (Srinivas et al., 2010; Chowdhury and Gopalan, 2017), that allows for construction of statistical confidence bounds around the unknown function.

Lemma 1. *Let $f \in \mathcal{H}_k(D)$ with $\|f\|_k \le B$, and consider the sampling model*

$$y_t = f(\boldsymbol{x}_t, \boldsymbol{\theta}_t) + \xi_t, \text{ where } \xi_t \sim \mathcal{N}(0, \sigma^2).$$

If the confidence parameter is set to

$$\beta_t = B + \sigma\lambda^{-1/2}\sqrt{2(\gamma_{t-1} + \ln(1/\delta))}, \quad (12)$$

[1] A similar idea has recently been used by Sessa et al. (2019) in the context of multi-agent repeated games.

Algorithm 1 GP-MRO

Input: Sets Θ, $\mathcal{X}$, kernel k, parameters η_T, $\{\beta_t\}_{t\ge1}$

1: **for** $t = 1, 2, \ldots, T$ **do**
2: For every $i \in \{1, \ldots, m\}$ set
$$\boldsymbol{w}_t[i] \propto \exp\Big\{ -\eta_T \sum_{j=1}^{t-1} \overline{\text{ucb}}_{j-1}(\boldsymbol{x}_j, \boldsymbol{\theta}_i)\Big\}$$
3: Set
$$\boldsymbol{x}_t \leftarrow \arg\max_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^m \boldsymbol{w}_t[i] \cdot \overline{\text{ucb}}_{t-1}(\boldsymbol{x}, \boldsymbol{\theta}_i)$$
4: $\boldsymbol{\theta}_t \leftarrow \arg\max_{\boldsymbol{\theta}\in\Theta} \sigma_{t-1}(\boldsymbol{x}_t, \boldsymbol{\theta}_i)$
5: Observe $y_t = f(\boldsymbol{x}_t, \boldsymbol{\theta}_t) + \xi_t$
6: Update $\mu_t(\cdot,\cdot)$ and $\sigma_t(\cdot,\cdot)$ according to (3) and (4) by including $\{(\boldsymbol{x}_t, \boldsymbol{\theta}_t, y_t)\}$
7: **end for**

Output: Uniform distr. $\mathcal{U}^{(T)}$ over $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$.

the following holds for every $(\boldsymbol{x}, \boldsymbol{\theta}) \in D$ and $t \ge 1$, with probability at least $1 - \delta$:

$$|\mu_{t-1}(\boldsymbol{x}, \boldsymbol{\theta}) - f(\boldsymbol{x}, \boldsymbol{\theta})| \le \beta_t \sigma_{t-1}(\boldsymbol{x}, \boldsymbol{\theta}), \quad (13)$$

where $\mu_{t-1}(\cdot,\cdot)$ and $\sigma_{t-1}(\cdot,\cdot)$ are given in (3) *and* (4) *with $\lambda > 0$.*

Given the definitions (7)-(8), and by conditioning on the event (13) in Lemma 1 holding true we have:

$$1 \ge \overline{\text{ucb}}_t(\boldsymbol{x}, \boldsymbol{\theta}) \ge f(\boldsymbol{x}, \boldsymbol{\theta}) \ge \overline{\text{lcb}}_t(\boldsymbol{x}, \boldsymbol{\theta}) \ge 0, \quad (14)$$

for every pair $(\boldsymbol{x}, \boldsymbol{\theta}) \in D$ and $t \ge 1$.

Next, we state our main theorem in which we bound the performance of GP-MRO. All the proofs from this section are provided in the supplementary material.

Theorem 2. *Fix $B > 0$, $\epsilon > 0$, $\delta \in (0, 1)$, $m \in \mathbb{Z}^+$, $\lambda \ge 1$, and suppose the following holds*

$$T \ge \frac{1}{\epsilon^2}\left(\frac{\log(m)}{2} + \beta_T\sqrt{32\lambda\gamma_T\log(m)} + 16\beta_T^2\lambda\gamma_T\right),$$

for some $T \in \mathbb{Z}^+$. For any $f : D \to [0, 1]$, such that $f \in \mathcal{H}_k(D)$ and $\|f\|_k \le B$, GP-MRO *with β_t set as in Lemma 1 and $\eta_T = \sqrt{\frac{8\log m}{T}}$ achieves*

$$\min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{U}^{(T)}}[f(\boldsymbol{x}, \boldsymbol{\theta})] \ge \max_{\mathcal{P}\in\Delta(\mathcal{X})} \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x}, \boldsymbol{\theta})] - \epsilon,$$

after T rounds with probability at least $1 - \delta$, where $\mathcal{U}^{(T)}$ is the distribution returned by GP-MRO.

Our analysis is based on the regret bounding techniques for zero-sum games similarly to (Chen et al., 2017) (we bound the rate of convergence to an equilibrium of the game simulated by GP-MRO), but with additional non-trivial challenges to characterize the excess regret due to the fact that f is unknown. The result in this theorem holds for general kernels and it can

be made more specific by substituting the bounds on γ_T for different kernels. For example, for $D \subset \mathbb{R}^d$ and the widely used squared exponential kernel, we obtain $T = \mathcal{O}^*\big(\frac{1}{\epsilon^2}\big(\log(m) + (\log\frac{1}{\epsilon})^d\sqrt{\log(m)} + (\log\frac{1}{\epsilon})^{2d}\big)\big)$, for constant λ, σ, B, d, m, where $\mathcal{O}^*(\cdot)$ is used to hide dimension-independent log factors. In the same setting, StableOpt (Bogunovic et al., 2018) requires $T = \mathcal{O}^*\big(\frac{1}{\epsilon^2}\big(\log\frac{1}{\epsilon}\big)^{2d}\big)$ samples to discover a *deterministic* maximin strategy that is near-optimal with respect to a generally *weaker* benchmark. Finally, in comparison to the result of Chen et al. (2017) where $T = \mathcal{O}\big(\frac{\log(m)}{\epsilon^2}\big)$ and f is assumed to be *known*, our bound characterizes an additional number of samples required for estimating the unknown RKHS function.

3.1.1 Trading Off Worst-Case and Average-Case Performance

In many scenarios, one might care about the performance of the reported distribution in the worst-case while also ensuring a good performance on "average". A natural problem to consider is to trade off these two quantities by using the following objective:

$$W(\mathcal{P}) := (1-\chi)\cdot \mathop{\mathbb{E}}_{\substack{\boldsymbol{\theta}\sim\mathcal{Q}\\ \boldsymbol{x}\sim\mathcal{P}}}[f(\boldsymbol{x},\boldsymbol{\theta})] + \chi \cdot \min_{\boldsymbol{\theta}\in\Theta} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\theta})],$$

for some fixed distribution $\mathcal{Q} \in \Delta(\Theta)$ (e.g., the uniform distribution) and trade-off parameter $\chi \in (0,1]$. Note that by setting $\chi = 1$, we recover the worst-case objective. Hence, our goal is to output $\mathcal{P}^{(T)} \in \Delta(\mathcal{X})$ after T rounds, such that for some accuracy $\epsilon > 0$

$$W(\mathcal{P}^{(T)}) \geq W(\mathcal{P}^*) - \epsilon, \tag{15}$$

where $\mathcal{P}^* \in \arg\max_{\mathcal{P}\in\Delta(\mathcal{X})} W(\mathcal{P})$.

Extending our algorithm to this case amounts to modifying the best response rule (Line 3 of Algorithm 1) as:

$$\boldsymbol{x}_t = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \Big[(1-\chi)\cdot \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim\mathcal{Q}}[\overline{\mathrm{ucb}}_{t-1}(\boldsymbol{x},\boldsymbol{\theta})] + \chi\cdot\sum_{i=1}^{m} \boldsymbol{w}_t[i]\cdot \overline{\mathrm{ucb}}_{t-1}(\boldsymbol{x},\boldsymbol{\theta}_i)\Big]. \tag{16}$$

The theoretical guarantees of GP-MRO in this setting with the best-response rule as given in (16) are provided in the following corollary.

Corollary 3. *Let Q be a fixed distribution in $\Delta(\Theta)$ and let $\chi \in (0,1]$ be a trade-off parameter. Under the setup of Theorem 2, and when the following holds*

$$T \geq \frac{1}{\epsilon^2}\left(\frac{\chi^2\log(m)}{2} + \chi\beta_T\sqrt{32\lambda\gamma_T\log(m)} + 16\beta_T^2\lambda\gamma_T\right),$$

for some $T \in \mathbb{Z}^+$, GP-MRO *with best-response rule as in* (16)*, achieves*

$$W(\mathcal{U}^{(T)}) \geq W(\mathcal{P}^*) - \epsilon,$$

after T rounds with probability at least $1-\delta$, where $\mathcal{U}^{(T)}$ is the returned uniform distribution over the queried points $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_T\}$.

The proof closely follows the one of Theorem 2. When $\chi = 1$, we recover Theorem 2, while the performance clearly improves for smaller values of χ, i.e., when $\chi \in (0,1)$. We also note that for $\chi = 0$, our algorithm solves the stochastic optimization problem, and achieves the standard regret bound (as in (Srinivas et al., 2010)) which is known to be nearly optimal for various kernels (see (Scarlett et al., 2017)).

4 Experiments

In this section, we evaluate the performance of GP-MRO on synthetic benchmarks and demonstrate the applicability of GP-MRO in planning safe trajectories for autonomous vehicles guided by user's preferences.

4.1 Synthetic Experiments

For a function $f : \mathcal{X}\times\Theta \to \mathbb{R}$, we compute the *performance* of a mixed strategy $\mathcal{P}^{(T)} \in \Delta(\mathcal{X})$ as:

$$\min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}^{(T)}}\big[f(\boldsymbol{x},\boldsymbol{\theta})\big]. \tag{17}$$

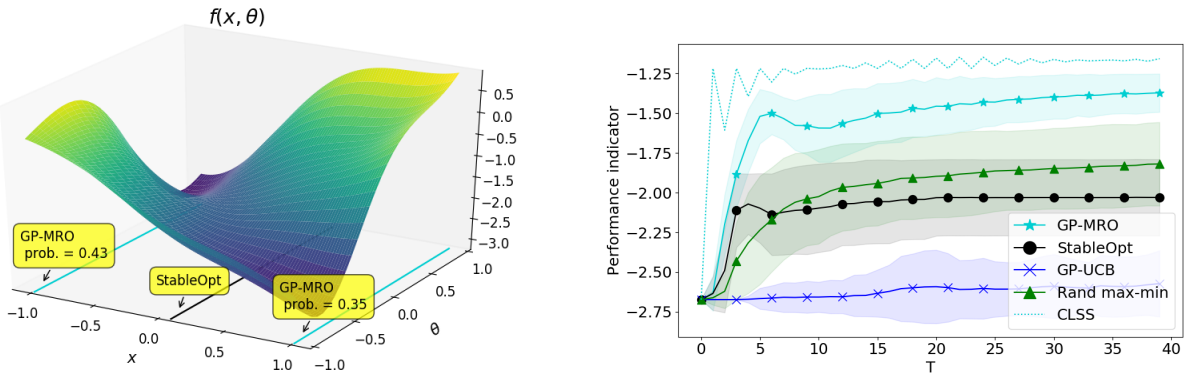
In case the strategy is deterministic $\boldsymbol{x}_T \in \mathcal{X}$, the performance is computed by considering the Dirac distribution centered at $\boldsymbol{x}_T$. We compare the performance of GP-MRO with the following baselines:

- StableOpt (Bogunovic et al., 2018) searches for the deterministic max-min point.
- GP-UCB (Srinivas et al., 2010) seeks for a non-robust global optimum and selects $(\boldsymbol{x}_t, \boldsymbol{\theta}_t) = \arg\max_{(\boldsymbol{x},\boldsymbol{\theta})\in\mathcal{X}\times\Theta} \overline{\mathrm{ucb}}_{t-1}(\boldsymbol{x},\boldsymbol{\theta})$ at every t. After T iterations, we consider $\boldsymbol{x}_T$ to be the returned point.
- RandMaxMin selects the point reported by StableOpt *or* GP-UCB with equal probability at every round, and returns a uniform distribution over these points.

We set $\beta_T = 2.0$ for each of the above algorithms (we found the theoretical choice to be overly conservative, as also noted in previous works (Srinivas et al., 2010; Bogunovic et al., 2018)), while η_T is set according to Theorem 2. As an idealized benchmark, we also test against (Chen et al., 2017, Algorithm 1) (which we name via the authors' surnames as CLSS) which assumes *oracle* access to f and thus upper bounds the achievable performance.

In the first experiment, we let $\mathcal{X}, \Theta \subset [-1,1]$ with $|\mathcal{X}| = 100$, and $|\Theta| = 30$, and sample a random function $f : \mathcal{X}\times\Theta \to \mathbb{R}$ from a $\mathrm{GP}(0,k)$ with kernel $k = k_{\mathrm{Lin}}\cdot k_{\mathrm{SE}}$. Moreover, we run the different baselines with the true prior $\mathrm{GP}(0,k)$ and noise standard deviation $\sigma = 1.0$.

In Figure 1a, we show f as well as the strategies returned by StableOpt and GP-MRO after $T = 40$ iterations. StableOpt converges to the max-min



(a) Selected points (b) Performance comparison

Figure 1: (a) Strategies returned by StableOpt and GP-MRO after $T = 40$ iterations. (b) Comparison of the performance of the considered baselines, computed as in (17). The proposed GP-MRO algorithm outperforms all the baselines. CLSS(Chen et al., 2017, Algorithm 1) assumes oracle access to f and upper bounds the achievable results.

point of f, while the distribution returned by GP-MRO assigns most of the probability mass to points $x = +1.0$ and $x = -1.0$. As shown in Figure 1b, this leads to a higher performance compared to all the considered baselines.

Next, we consider the synthetic function $g_{\text{poly}} : \mathbb{R}^2 \to \mathbb{R}$ from (Bertsimas et al., 2010), and the robust optimization task from (Bogunovic et al., 2018). The goal is to select points $\boldsymbol{x} = (x_1, x_2)$ that maximize g_{poly} subject to the worst-case perturbation $\boldsymbol{\theta} \in \Theta$. We map such problem to our setting by defining $f(\boldsymbol{x}, \boldsymbol{\theta}) = g_{\text{poly}}(\boldsymbol{x} - \boldsymbol{\theta})$. The decision space $\mathcal{X}$ consists of a uniformly spaced grid of $10'000$ points, while the set of perturbations Θ is obtained by drawing 100 random points from the unit ball centered at the origin.

We set noise standard deviation $\sigma = 1.0$ and run all the algorithms using Matérn kernel k_{Mat} for $T = 200$ iterations (kernel hyperparameters are found via maximum-likelihood method). In Figure 2a, we plot the function g_{poly} as well as the support of the strategies returned by StableOpt (in black) and GP-MRO (in cyan). For GP-MRO we plot only points selected with probability mass greater than 0.01. StableOpt is able to discover the max-min point of g_{poly}, while GP-MRO randomizes between points in the max-min region and points close to the global optimum. This leads to a higher performance compared to other baselines, as shown in Figure 2b.

4.2 Human-assisted trajectory planning for autonomous vehicles

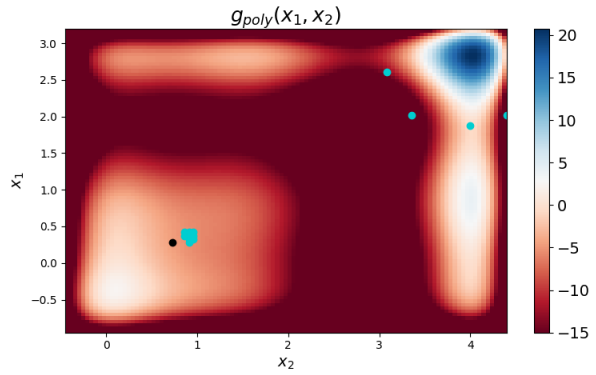
We study the problem of planning safe trajectories for an Autonomous Vehicle (AV) driving on roads shared with human-driven vehicles (HVs). We consider the situation depicted in Figure 3a, where the AV (in yellow) is approaching, with a speed of 20 m/s, a HV (in red) driving at a constant speed of 10 m/s. The intentions of the HV are uncertain and this should be taken into account when planning the AV's trajectory.

In the context of autonomous driving and AV-HV interactions, deterministic strategies would make AVs' actions predictable, hence giving a significant advantage to HVs. We observe this fact in our simulations, where such strategies tend to be overly conservative and prevent the AV from completing the overtake manoeuvre. Similarly, we expect this to occur in many other challenging scenarios such as intersections (Liu et al., 2018), or when merging into dense lanes (Bouton et al., 2019). Instead, we model such problem according to Section 2 and seek for robust *mixed strategies* for the AV. This is in contrast with previous works (e.g., (Fisac et al., 2019; Sadigh et al., 2016)) where deterministic strategies are found, assuming a specific behavioral model for the HV.

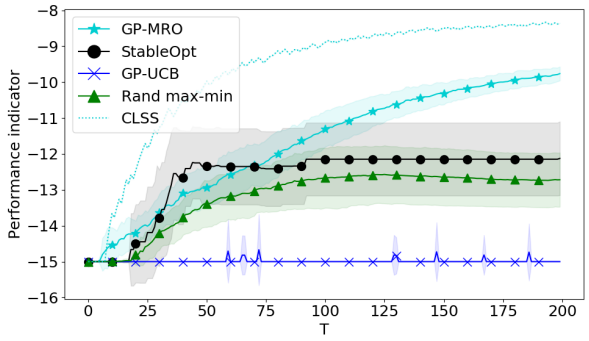
Further on, our goal is to plan trajectories for the AV which best reflect typical *human driving preferences* (e.g., driving styles, security measures, and safe behaviors that the AV should follow). For instance, in the specific situation of Figure 3a, a good trajectory for the AV should depend on the importance that humans give to overtaking rather than breaking behind the HV. We encode such driving preferences with an unknown *scoring function*. We assume we can learn such function by sequential evaluations obtained interacting with a *user* who assists our planning phase.

Computing such mixed strategies requires enough computation and relies on sequential interactions with the user. Hence, after illustrating our approach, we propose an *offline* scheme to pre-compute a control *policy* for the AV using GP-MRO.

Decision sets. A strategy for the AV consists of selecting a steering angle $x_1 \in [-\frac{\pi}{60}, \frac{\pi}{60}]$, and an acceleration $x_2 \in [-10, 1]$. Once chosen, both are assumed



(a) Selected points



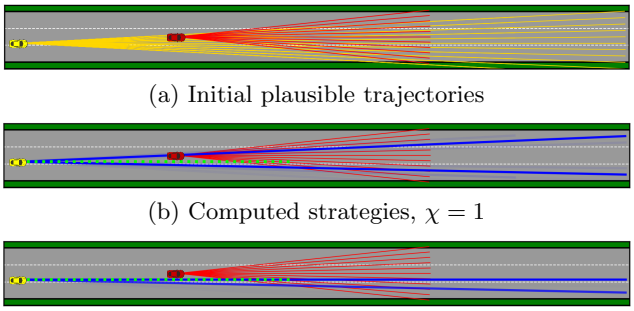
(b) Performance comparison

Figure 2: (a) Supports of the strategies returned by StableOpt (in black) and GP-MRO (in cyan) after $T = 200$ iterations. StableOpt reports a deterministic strategy, while GP-MRO returns a randomized strategy. (b) Performance of the different baselines, computed as in (17). The mixed strategy returned by GP-MRO outperforms all the baselines. The CLSS algorithm has oracle access to f and upper bounds the achievable performance.

to be constant for the horizon of 8 s. Hence, we let $\mathcal{X}$ be the set of points $\boldsymbol{x} = (x_1, x_2)$. Similarly, we assume the HV travels at a constant speed and can choose a steering angle $\boldsymbol{\theta} \in \Theta = [-\frac{\pi}{30}, \frac{\pi}{30}]$. We discretize both $\mathcal{X}$ and Θ using uniform grids of 121 and 11 points, respectively. Car trajectories (depicted in Figure 3a) are computed using the commonly used discrete-time bicycle model (Polack et al., 2017) with time steps of 0.04 s.

Optimization goal. We let the scoring function $f : \mathcal{X} \times \Theta \to [0, 1]$ reflect the humans' driving preferences for the AV. As discussed later, f measures how rewarding is for the AV to select a possible $\boldsymbol{x} \in \mathcal{X}$ when the HV decides to steer with angle $\boldsymbol{\theta} \in \Theta$. Our goal is to compute a robust mixed strategy which solves the problem in (1). More generally, according to Section 3.1.1, we can incorporate *priors* $\mathcal{Q} \in \Delta(\Theta)$ on HV's behaviors and find strategies that can trade-off worst-case and average-case performance, for a trade-off parameter $\chi \in (0, 1]$.

Scoring function. We assume that f is initially unknown but can be learned by iteratively querying the user. Querying f at a given point $(\boldsymbol{x}, \boldsymbol{\theta})$ consists of: 1) Forward simulating the AV's and HV's trajectories corresponding to $\boldsymbol{x}$ and $\boldsymbol{\theta}$ and 2) Presenting the outcome of such simulation to the user who assigns a score to the considered trajectories. In this experiment, we assume such score is determined by a feature vector $\mathbf{z} = [z_1, z_2, z_3] \in \mathbb{R}^3$ that can be extracted from the simulated trajectories. Such vector consists of: longitudinal distance travelled by the AV (z_1), AV's maximum absolute lateral position (z_2), and the minimum distance between the AV and the human-driven car (z_3). We use a model of the unknown f of the following form: $f_{\text{p}}(z_1) + f_{\text{r}}(z_2) + f_{\text{eb}}(z_3)$, where f_{p} rewards progress, f_{r} penalizes exiting the road limits, while f_{eb} penalizes the AV if it gets too close to the human-driven car and therefore needs to activate emergency breaking. In future work, we plan to replace our model and test our approach with scores coming from real users.



(a) Initial plausible trajectories

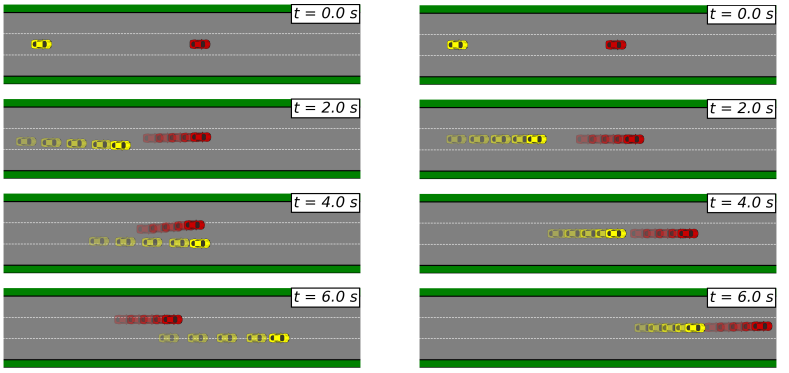
(b) Computed strategies, $\chi = 1$

(c) Computed strategies, $\chi = 0.8$

Figure 3: (a) Initial plausible trajectories of the AV (yellow) and the HV (red). In (b) and (c), the robust deterministic strategy (in dotted light-green) corresponds to breaking and not overtaking. The mixed strategy $\mathcal{U}^{(T)}$ found by GP-MRO is represented by the blue trajectories (intensities proportional to their probabilities) and depends on the trade-off parameter χ.

4.3 Illustration of the mixed strategies computed by GP-MRO

We consider the configuration in Figure 3a and compute a mixed strategy for the AV running GP-MRO for $T = 100$ iterations. We set trade-off parameter $\chi = 1$, $\beta_T = 0.5$, and $\eta_T = 0.5$. To learn f, we fit a GP with kernel function $k(\mathbf{z}, \mathbf{z}') = k^1_{\text{Mat}}(z_1, z'_1) + k^2_{\text{Mat}}(z_2, z'_2) + k^3_{\text{Mat}}(z_3, z'_3)$ where the feature vector $\mathbf{z}$ is computed as explained above. In Figure 3b, we depict (in blue) the support of the mixed strategy $\mathcal{U}^{(T)}$ where the color intensity of a trajectory is proportional to its probability. Additionally, we show (in dotted light-green) the trajectory corresponding to the robust deterministic strategy $\boldsymbol{x}_{\text{r}} \in \arg\max_{\boldsymbol{x} \in \mathcal{X}} \min_{\boldsymbol{\theta} \in \Theta} f(\boldsymbol{x}, \boldsymbol{\theta})$.



(a) GP-MRO (b) Max-min deterministic strategy

Figure 4: Closed-loop simulation of the AV (yellow) and human-driven car (red). At every iteration, the AV implements (a) the randomized policy found by GP-MRO or (b) the deterministic max-min strategy. The human-driven car follows the noisy rational Boltzmann policy (18). The robust deterministic strategies are overly conservative, while GP-MRO algorithm allows the AV to safely overtake.

The strategy $\mathcal{U}^{(T)}$ randomizes between an overtake from the left or the right side. Instead, $\boldsymbol{x}_{\mathrm{r}}$ amounts to breaking and thus never overtaking.

Our next goal is to find a strategy for the AV which can trade off the worst-case with average-case performance. Let us assume that, with probability 0.2, the HV doesn't realize the presence of the AV and thus has no intention to steer. In this case, we can seek for the optimal strategy for the AV by setting $\chi = 0.8$ and letting $\mathcal{Q} \in \Delta(\Theta)$ be a Dirac distribution corresponding to the HV proceeding straight. In Figure 3c we depict the strategy returned by GP-MRO, together with the trajectory $\boldsymbol{x}_{\mathrm{r}} \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} (1-\chi)\cdot\mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{Q}}\big[f(\boldsymbol{x},\boldsymbol{\theta})\big] + \chi\cdot \min_{\boldsymbol{\theta}\in\Theta} f(\boldsymbol{x},\boldsymbol{\theta})$. In this case, $\mathcal{U}^{(T)}$ favors an overtake from the right, while $\boldsymbol{x}_{\mathrm{r}}$ still leads to no overtaking.

4.4 Closed-loop simulations

We propose the following *offline* procedure to precompute a control *policy* for the AV. We consider a finite set of $\sim 8'000$ possible scenarios $\mathbf{s} \in \mathcal{S} \subset \mathbb{R}^5$, each describing the initial and relative positions and velocities of the two cars. We compute a mixed strategy $\mathcal{U}^{(T)}(\mathbf{s})$ for each scenario $\mathbf{s} \in \mathcal{S}$ using GP-MRO with $\chi = 1$. Moreover, to make our approach more tractable, we query f at chosen points $(\boldsymbol{x}_t, \boldsymbol{\theta}_t)$ (Line 5 in Algorithm 1) only if $\sigma_{t-1}(\boldsymbol{x}_t, \boldsymbol{\theta}_t)$ is greater than 0.005. By doing so, we end up with a policy mapping scenarios $\mathbf{s} \in \mathcal{S}$ to distributions $\mathcal{U}^{(T)}(\mathbf{s}) \in \Delta(\mathcal{X})$ after a total number of 136 queries of the unknown function.

We evaluate the policy *online*, in a receding-horizon fashion: Starting from given initial positions and velocities, every 2 s we map the cars' positions and velocities to the closest $\mathbf{s} \in \mathcal{S}$ (using a nearest-neighbour tree-based algorithm) and let the AV sample its trajectory from $\mathcal{U}^{(T)}(\mathbf{s})$. For the behavior of the HV we implement a noisy rational Boltzmann policy (as in (Fisac et al., 2019)) where, in a given scenario $\mathbf{s} \in \mathcal{S}$, $\boldsymbol{\theta} \in \Theta$ is sampled with probability

$$\mathbb{P}[\boldsymbol{\theta} = \boldsymbol{\theta}_i \mid \mathbf{s}\,] \propto \exp\Big(\mathbb{E}_{\boldsymbol{x}\sim\mathcal{U}^{(T)}(\mathbf{s})}\, f_H(\boldsymbol{\theta}_i, \boldsymbol{x})\Big). \qquad (18)$$

The function f_H rewards progress for the HV and penalizes exiting the road or getting too close to the AV, the same way as f does for the AV.

	GP-MRO	Deterministic max-min
# of overtakes	408/1000	0/1000
avg. final pos. AV	169.4 m	123.1 m
avg. final pos. human	139.8 m	139.9 m

Table 1: Number of overtakes and cars' average final positions out of 1000 closed-loop simulations of 10 s.

In Figure 4, we plot several snapshots of a closed-loop simulation of 10 s where the AV samples trajectories from the pre-computed policy (a), and where the AV chooses the max-min strategy $\boldsymbol{x}_{\mathrm{r}}$ at every iteration (b). As can be seen from Figure 4, the proposed approach allows the AV to safely overtake, while the robust deterministic strategy is too conservative and forces the AV to break behind the HV. We repeat the closed-loop simulation for $1'000$ times (for fixed initial positions and velocities of the two cars). As reported in Table 1, the deterministic strategy is non-overtaking and the AV reaches an average final longitudinal positions of 123.1 m. Instead, using the pre-computed randomized policy the AV successfully overtakes the human-driven car in 408 cases (in the remaining cases it breaks behind the HV), reaching an average final position of 169.4 m.

5 Conclusion

We have studied a robust optimization problem in which the objective function is unknown and depends on an uncertain parameter. For this problem, we have proposed a novel sample-efficient algorithm GP-MRO, which can discover a near-optimal randomized and robust strategy. We have established rigorous theoretical guarantees and designed a variant of GP-MRO that effectively trades off worst-case and average-case performance. In synthetic experiments and trajectory planning tasks, we have showed that our proposed algorithm significantly outperforms existing baselines.

Acknowledgments

This work was gratefully supported by the Swiss National Science Foundation, under the grant SNSF 200021_172781, by the European Union's ERC grant 815943, and ETH Zürich Postdoctoral Fellowship 19-2 FEL-47.

References

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust optimization*. Princeton University Press.

Ben-Tal, A., Margalit, T., and Nemirovski, A. (2000). *Robust Modeling of Multi-Stage Portfolio Problems*, pages 303–328. Springer US.

Bertsimas, D., Brown, D. B., and Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501.

Bertsimas, D., Nohadani, O., and Teo, K. M. (2010). Robust optimization for unconstrained simulation-based problems. *Operations Research*, 58(1):161–178.

Bertsimas, D. and Thiele, A. (2004). A robust optimization approach to supply chain management. In *Integer Programming and Combinatorial Optimization*, pages 86–100. Springer Berlin Heidelberg.

Bogunovic, I., Scarlett, J., and Cevher, V. (2016a). Time-varying Gaussian process bandit optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 314–323.

Bogunovic, I., Scarlett, J., Jegelka, S., and Cevher, V. (2018). Adversarially robust optimization with Gaussian processes. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 5760–5770.

Bogunovic, I., Scarlett, J., Krause, A., and Cevher, V. (2016b). Truncated variance reduction: A unified approach to Bayesian optimization and level-set estimation. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Bouton, M., Nakhaei, A., Fujimura, K., and Kochenderfer, M. J. (2019). Cooperation-aware reinforcement learning for merging in dense traffic. *arXiv preprint arXiv:1906.11021*.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge University Press.

Chen, R. S., Lucier, B., Singer, Y., and Syrgkanis, V. (2017). Robust optimization for non-convex objectives. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 4705–4714.

Chowdhury, S. R. and Gopalan, A. (2017). On kernelized multi-armed bandits. In *International Conference on Machine Learning (ICML)*, pages 844–853.

Contal, E., Buffoni, D., Robicquet, A., and Vayatis, N. (2013). Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 225–240. Springer.

Fisac, J. F., Bronstein, E., Stefansson, E., Sadigh, D., Sastry, S. S., and Dragan, A. D. (2019). Hierarchical game-theoretic planning for autonomous vehicles. *International Conference on Robotics and Automation (ICRA)*.

Frazier, P. I. (2018). A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139.

He, X. and Kempe, D. (2016). Robust influence maximization. In *Int. Conf. Knowledge Discovery and Data Mining (KDD)*, pages 885–894.

Jørgensen, T. B., Wolniakowski, A., Petersen, H. G., Debrabant, K., and Krüger, N. (2018). Robust optimization with applications to design of context specific robot solutions. *Robotics and Computer-Integrated Manufacturing*, 53:162–177.

Kawase, Y. and Sumita, H. (2019). Randomized strategies for robust combinatorial optimization. *AAAI Conference on Artificial Intelligence (AAAI)*.

Krause, A. and Ong, C. S. (2011). Contextual Gaussian process bandit optimization. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 2447–2455.

Krause, A., Roper, A., and Golovin, D. (2011). Randomized sensing in adversarial environments. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Liu, X., Hsieh, P.-C., and Kumar, P. R. (2018). Safe intersection management for mixed transportation systems with human-driven and autonomous vehicles. *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*.

Martinez-Cantin, R., Tee, K., and McCourt, M. (2018). Practical Bayesian optimization in the presence of outliers. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Namkoong, H. and Duchi, J. C. (2016). Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Nogueira, J., Martinez-Cantin, R., Bernardino, A., and Jamone, L. (2016). Unscented Bayesian optimization for safe robot grasping. In *Interna-*

tional Conference on Intelligent Robots and Systems (IROS).

Oliveira, R., Ott, L., and Ramos, F. (2019). Bayesian optimisation under uncertain inputs. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Polack, P., Altché, F., d'Andréa-Novel, B., and de La Fortelle, A. (2017). The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In *IEEE Intelligent Vehicles Symposium (IV)*.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian processes for machine learning*, volume 1. MIT press Cambridge.

Sadigh, D., Sastry, S. S., Seshia, S. A., and Dragan, A. D. (2016). Planning for autonomous cars that leverage effects on human actions. In *Robotics: Science and Systems*.

Scarlett, J., Bogunovic, I., and Cevher, V. (2017). Lower bounds on regret for noisy Gaussian process bandit optimization. In *Conference on Learning Theory (COLT)*.

Sessa, P. G., Bogunovic, I., Kamgarpour, M., and Krause, A. (2019). No-regret learning in unknown games with correlated payoffs. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Shalev-Shwartz, S. and Wexler, Y. (2016). Minimizing the maximal loss: How and why. In *International Conference on Machine Learning (ICML)*.

Sinha, A., Fang, F., An, B., Kiekintveld, C., and Tambe, M. (2018). Stackelberg security games: Looking beyond a decade of success. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Sinha, A., Namkoong, H., and Duchi, J. (2017). Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, pages 1015–1022.

Staib, M., Wilder, B., and Jegelka, S. (2018). Distributionally robust submodular maximization. *arXiv preprint arXiv:1802.05249*.

Sui, Y., Gotovos, A., Burdick, J., and Krause, A. (2015). Safe exploration for optimization with Gaussian processes. In *International Conference on Machine Learning*, pages 997–1005.

Tesch, M., Schneider, J., and Choset, H. (2011). Adapting control policies for expensive systems to changing environments. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 357–364.

Vorobeychik, Y. and Li, B. (2014). Optimal randomized classification in adversarial settings. In *International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*.

Wang, Z. and Jegelka, S. (2017). Max-value entropy search for efficient Bayesian optimization. In *International Conference on Machine Learning (ICML)*, pages 3627–3635.

Williams, B. J., Santner, T. J., and Notz, W. I. (2000). Sequential design of computer experiments to minimize integrated response functions. In *Statistica Sinica*, pages 1133–1152.