Improved Max-value Entropy Search for Multi-objective Bayesian Optimization with Constraints

Daniel Fernández-Sánchez[1] **Eduardo C. Garrido-Merchán**[2] **Daniel Hernández-Lobato**[1]

[1]Computer Science Department, Universidad Autónoma de Madrid
[2]Faculty of Economics and Business Administration, Universidad Pontificia Comillas

Abstract We present Improved Max-value Entropy search for Multi-Objective Bayesian optimization with Constraints (MESMOC+) for the constrained optimization of expensive-to-evaluate black-boxes. It is based on minimizing the entropy of the solution of the problem in function space (*i.e.*, the Pareto front) to guide the search for the optimum. Its cost is linear in the number of black-boxes, and due to its expression, it can be used in a decoupled evaluation setting in which we chose where and also what black-box (objective or constraint) to evaluate. Our synthetic experiments show that MESMOC+ has similar performance to other state-of-the-art acquisition functions, but it is faster to execute, simpler to implement and it is more robust with respect to the number of samples of the Pareto front.

1 Introduction

Consider the problem of optimizing K objectives $f_1(\mathbf{x}), \ldots, f_K(\mathbf{x})$ while fulfilling C constraints $c_1(\mathbf{x}), \ldots, c_C(\mathbf{x})$, over a bounded input space in $\mathcal{X} \subset \mathbb{R}^d$, where d is the dimensionality of $\mathcal{X}$. For example, we might want to maximize the speed of a robot while minimizing its energy consumption (Ariizumi et al., 2014), and avoiding breaking any of its joints. Another example is to minimize simultaneously the classification error and the prediction time of a deep neural network (DNN) while the DNN is constrained to not exceeding a certain amount of memory.

In these problems, most of the times there is no single optimal point but a set of optimal points called the Pareto set $\mathcal{X}^\star$ (Collette and Siarry, 2004). The objective values associated to the points in $\mathcal{X}^\star$ constitute the Pareto front $\mathcal{Y}^\star$. All the points in $\mathcal{X}^\star$ are optimal because they are not *dominated* by any other point in $\mathcal{X}$. In a minimization context, a point $\mathbf{x}_1$ *dominates* $\mathbf{x}_2$ if $f_k(\mathbf{x}_1) \leq f_k(\mathbf{x}_2)$, $\forall k \in \{1, ..., K\}$, with at least one strictly minor inequality. Thus, given $\mathcal{X}^\star$ it is impossible to improve the value in one objective without deteriorating the other objectives. Moreover, the points in $\mathcal{X}^\star$ must be feasible, *i.e.*, they must satisfy $c_j(\mathbf{x}^\star) \geq 0, \forall \mathbf{x}^\star \in \mathcal{X}^\star, \forall j = \{1, ..., C\}$. The potential size of $\mathcal{X}^\star$ is infinite, so it must be approximated by a finite set of points.

The problems described have three main characteristics. (i) There is no analytical form for the objectives nor the constraints, *i.e.*, the black-boxes. (ii) The evaluations may be contaminated by noise. (iii) New evaluations are expensive in some way, *e.g.*, economically or temporally. To solve this type of problems while minimizing the number of evaluations, one can use Bayesian optimization (BO) (Brochu et al., 2009). BO methods first use a model to estimate the potential values of the black-boxes in unexplored regions of the space $\mathcal{X}$. Usually, Gaussian processes (GPs) (Rasmussen and Williams, 2006) are the models employed (Shahriari et al., 2015). Then, an acquisition function is used to measure the expected utility of evaluating the black-boxes at each input point of $\mathcal{X}$ given the model's predictions. The maximizer of the acquisition function is the next location to evaluate. This process is repeated for a fixed number of iterations. After this, the models are optimized to obtain an approximate solution of the optimization problem. This approach is expected to be very useful if the black-boxes are very expensive to evaluate, and the acquisition function is very cheap to compute (Shahriari et al., 2015).

In the context of constrained multi-objective BO problems, current acquisition functions are divided in two groups. Adaptations of *expected hyper-volume improvement* (EHI) (Emmerich and Klinkenberg, 2008), and entropy search. The hyper-volume is the space of points above the Pareto front, assuming minimization, and is maximized by the solution of the optimization problem. Adaptations of EHI need to deal with an intractable integral from a constrained definition of EHI. They approximate this integral by Monte Carlo. Examples of these techniques are (Feliot et al., 2017; Daulton et al., 2020, 2021). However, some of them have problems estimating the acquisition function, which is zero almost everywhere after a few evaluations (Daulton et al., 2020, 2021). Moreover, none of these methods can handle decoupled scenarios in which one chooses not only the next point to evaluate but also what black-box to evaluate next.

PESMOC is an acquisition function of the second group (Garrido-Merchán and Hernández-Lobato, 2019). PESMOC approximates an intractable expression that evaluates the expected reduction in the entropy of $\mathcal{X}^\star$. For this PESMOC, uses expectation propagation (EP), an approximate method that is costly and difficult to implement. MESMOC is a simpler and faster alternative to PESMOC that is based on approximately evaluating the expected reduction in the entropy of the Pareto front $\mathcal{Y}^\star$ (Belakaria et al., 2021). Nevertheless, the approximation of MESMOC is very crude and it simply tries to maximize each objective and constraint independently. Appendix B has more details about this. Both PESMOC and MESMOC can deal with decoupled evaluations. $\{PF\}^2ES$ is another information-based strategy that reduces the entropy of $\mathcal{Y}^\star$. It uses variational inference to approximate the mutual information and it allows for parallel evaluations. If non-parallel evaluations are considered, $\{PF\}^2ES$ is outperformed by our method (Qing et al., 2022).

As an alternative to MESMOC, we provide here a more accurate approximation to the expected reduction in the entropy of $\mathcal{Y}^\star$. We call our method MESMOC+. At each iteration, MESMOC+ chooses to evaluate the point that is expected to reduce the most the entropy of $\mathcal{Y}^\star$. Reducing the entropy of $\mathcal{Y}^\star$ means that more information about the solution of the problem is available, so we are closer to finding the problem's solution (Villemonteix et al., 2009; Hennig and Schuler, 2012).

2 Improved Max-value Entropy Search for Multi-objective BO with Constraints

Let $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ be the set of evaluations performed up to iteration N, where $\mathbf{x}_n$ is the point evaluated in the n-th iteration and $\mathbf{y}_n$ is a vector with the values of the K+C black-boxes at $\mathbf{x}_n$, *i.e.*, $\mathbf{y}_n = (f_1(\mathbf{x}_n), \ldots, f_K(\mathbf{x}_n), c_1(\mathbf{x}_n), \ldots, c_C(\mathbf{x}_n))$. MESMOC+ tries to reduce the entropy of $\mathcal{Y}^\star$ after performing an evaluation at $\mathbf{x}_{N+1}$. Therefore, MESMOC+'s acquisition function is:

$$\alpha(\mathbf{x}) = H\left(\mathcal{Y}^\star|\mathcal{D}\right) - \mathbb{E}_{\mathbf{y}}\left[H\left(\mathcal{Y}^\star|\mathcal{D} \cup \{(\mathbf{x}, \mathbf{y})\}\right)\right], \tag{1}$$

where $H\left(\mathcal{Y}^\star|\mathcal{D}\right)$ is the entropy of the $\mathcal{Y}^\star$, given the current dataset $\mathcal{D}$; $H(\mathcal{Y}^\star|\mathcal{D} \cup \{(\mathbf{x}, \mathbf{y})\})$ is the entropy of $\mathcal{Y}^\star$ after including the new data point $(\mathbf{x}, \mathbf{y})$ in the dataset; and the expectation $\mathbb{E}_{\mathbf{y}}[\cdot]$ is calculated over the potential values for $\mathbf{y}$ at $\mathbf{x}$, according to the GPs.

Evaluating the entropy of $\mathcal{Y}^\star$ is very challenging. In order to avoid this problem, we follow (Hernández-Lobato et al., 2016) and rewrite (1) in an equivalent form, as in (Wang and Jegelka, 2017), by noting that (1) is the mutual information between $\mathcal{Y}^\star$ and $\mathbf{y}$, $I(\mathcal{Y}^\star; \mathbf{y})$ (Hernández-Lobato et al., 2014, 2016). Therefore, since $I(\mathcal{Y}^\star; \mathbf{y}) = I(\mathbf{y}; \mathcal{Y}^\star)$, we can swap the roles of $\mathcal{Y}^\star$ and $\mathbf{y}$ in (1) and MESMOC+'s acquisition function becomes:

$$\alpha(\mathbf{x}) = H\left(\mathbf{y}|\mathcal{D}, \mathbf{x}\right) - \mathbb{E}_{\mathcal{Y}^\star}\left[H\left(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{Y}^\star\right)\right], \tag{2}$$

where the first term of the *r.h.s* is the entropy of the current GPs predictive distribution, which is Gaussian. Namely, $H\left(\mathbf{y}|\mathcal{D}, \mathbf{x}\right) = \sum_{k=1}^K \log(2\pi e v_k^f(\mathbf{x}))/2 + \sum_{j=1}^C \log(2\pi e v_j^c(\mathbf{x}))/2$; and $\mathbb{E}_{\mathcal{Y}^\star}[H\left(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{Y}^\star\right)]$ is the expected entropy of the predictive distribution conditioned to $\mathcal{Y}^\star$ being the solution of the problem. This second term is intractable. To approximate the expectation we

use Monte Carlo sampling and sample $\mathcal{Y}^\star$ in an equivalent way to how Garrido-Merchán and Hernández-Lobato (2019) generate samples of $\mathcal{X}^\star$. We explain our approach for approximating the entropy of the conditional predictive distribution in the next section.

2.1 Approximating the Conditional Predictive Distribution

Consider first a noiseless scenario and let $\mathbf{f} = \{f_1(\mathbf{x}), \ldots, f_K(\mathbf{x})\}$ and $\mathbf{c} = \{c_1(\mathbf{x}), \ldots, c_C(\mathbf{x})\}$. The expression of $p(\mathbf{f}, \mathbf{c}|\mathcal{D}, \mathbf{x}, \mathcal{Y}^\star)$ is obtained using Bayes' rule:

$$p(\mathbf{f}, \mathbf{c}|\mathcal{D}, \mathbf{x}, \mathcal{Y}^\star) = Z^{-1} p(\mathbf{f}, \mathbf{c}|\mathcal{D}, \mathbf{x}) p(\mathcal{Y}^\star|\mathbf{f}, \mathbf{c}), \tag{3}$$

where Z^{-1} is a normalization constant, $p(\mathbf{f}, \mathbf{c}|\mathcal{D}, \mathbf{x})$ is the predictive distribution for the black-boxes given $\mathcal{D}$ at $\mathbf{x}$, and $p(\mathcal{Y}^\star|\mathbf{f}, \mathbf{c})$ is the probability that $\mathcal{Y}^\star$ is a valid Pareto front given $\mathbf{f}$ and $\mathbf{c}$. Note that $p(\mathbf{f}, \mathbf{c}|\mathcal{D}, \mathbf{x})$ is simply a product of Gaussians given by the predictive distribution of each GP.

The factor $p(\mathcal{Y}^\star|\mathbf{f}, \mathbf{c})$ in (3) removes all configurations of the objectives and constraints values, $(\mathbf{f}, \mathbf{c})$, that are incompatible with $\mathcal{Y}^\star$ being the Pareto front of the problem. Therefore, $p(\mathcal{Y}^\star|\mathbf{f}, \mathbf{c})$ must be 0 when $\mathbf{c}$ does not violate the constraints (*i.e*, $\mathbf{c}$ does satisfy $c_j(\mathbf{x}) \geq 0, \forall j \in \{1, ..., C\}$), and $\mathbf{f}$ is not *Pareto dominated* by any $\mathbf{f}^\star \in \mathcal{Y}^\star$. Similarly, $p(\mathcal{Y}^\star|\mathbf{f}, \mathbf{c})$ is 1 if all points $\mathbf{f}^\star$ in the Pareto front $\mathcal{Y}^\star$ dominate $\mathbf{f}$, or if $\mathbf{c}$ violates the constraints (*i.e.* at least one constraint is negative at $\mathbf{x}$). Thus,

$$p(\mathcal{Y}^\star|\mathbf{f}, \mathbf{c}) \propto \prod_{\mathbf{f}^\star \in \mathcal{Y}^\star} \left(1 - \prod_{j=0}^{C} \Theta(c_j) \prod_{k=0}^{K} \Theta\left(f_k^\star - f_k\right)\right) \propto \prod_{\mathbf{f}^\star \in \mathcal{Y}^\star} \Omega(\mathbf{f}^\star, \mathbf{f}, \mathbf{c}), \tag{4}$$

where $\Theta(\cdot)$ is the Heaviside step function, $f_k = f_k(\mathbf{x})$, $c_j = c_j(\mathbf{x})$, $f_k^\star$ is the k-th value of the vector of values $\mathbf{f}^\star$ of $\mathcal{Y}^\star$ and $\Omega(\mathbf{f}^\star, \mathbf{f}, \mathbf{c}) = 1 - \prod_{j=0}^{C} \Theta(c_j) \prod_{k=0}^{K} \Theta\left(f_k^\star - f_k\right)$. Note that (4) will be 1, if $\Omega(\mathbf{f}^\star, \mathbf{f}, \mathbf{c})$ is 1 for all the $\mathbf{f}^\star$ in $\mathcal{Y}^\star$. To make $\Omega(\mathbf{f}^\star, \mathbf{f}, \mathbf{c})$ be 1, either $\prod_{j=0}^{C} \Theta(c_j(\mathbf{x}))$ or $\prod_{k=0}^{K} \Theta\left(f_k^\star - f_k(\mathbf{x})\right)$ must be 0. This happens if all the values of $\mathbf{c}$ are greater or equal to 0 or if all the values of $\mathbf{f}^\star$ are lower or equal to those of $\mathbf{f}$, except one which must be strictly minor.

The computation of the entropy of (3) is intractable. Therefore, we need to approximate this distribution. Importantly, we would like the acquisition function to be cheap compared to the cost of evaluating the black-boxes. For this reason, we use Assumed Density Filtering (ADF) (Boyen and Koller, 1998; Minka, 2001). ADF simply approximates each non-Gaussian factor in (3) using a Gaussian. Unlike EP, ADF refines each non-Gaussian factor only one time. Thus, ADF is faster and simpler than EP. Since the predictive distribution of a GP is Gaussian, the only non-Gaussian factors are the $\Omega(\mathbf{f}^\star, \mathbf{f}, \mathbf{c})$ factors in (4). Recall that we assume independence among the objectives and constraints. Since the Gaussian distribution is closed under the product operation and the factors $\Omega(\mathbf{f}^\star, \mathbf{f}, \mathbf{c})$ only involve independent Gaussian random variables, the approximation of (3) is a factorizing Gaussian. In Appendix C we describe the specific ADF updates.

2.2 The MESMOC+ Acquisition Function

After the execution of ADF, the variances of the objectives and the constraints of the predictive distribution at $\mathbf{x}$, conditioned to the Pareto front $\mathcal{Y}^\star$, are available. Since we use ADF to approximate Gaussian distributions to (3), the approximate entropy has a similar form to that of $H(\mathbf{f}, \mathbf{c}|\mathcal{D}, \mathbf{x})$. Thus, our approximation of (2) is just $H(\mathbf{f}, \mathbf{c}|\mathcal{D}, \mathbf{x})$ minus entropy of $p(\mathbf{f}, \mathbf{c}|\mathcal{D}, \mathbf{x}, \mathcal{Y}^\star)$. Importantly, however, as a consequence of the step functions, ADF tends to decrease little the variance of approximate distributions. Therefore, when the unconditioned variance is small *e.g.*, 10^{-5}, ADF may reduce that variance too much *e.g.*, 10^{-6}. If we now calculate the entropy reduction, the result will be an acquisition value much larger than what it should be (so the acquisition function will always tend to evaluate similar points). To solve this, we modified MESMOC+'s acquisition function to take into account the absolute reduction in the variance instead, and ignore the log operation. Thus, the final expression of MESMOC+, after adding observational noise, is:

$$\alpha(\mathbf{x}) \approx \sum_{k=1}^{K} \left(v_k^f + (\sigma_k^f)^2\right) + \sum_{j=1}^{C} \left(v_j^c + (\sigma_j^c)^2\right) - \frac{1}{M} \sum_{m=1}^{M} \left[\sum_{k=1}^{K} \left(\tilde{v}_k^f + (\sigma_k^f)^2\right) + \sum_{j=1}^{C} \left(\tilde{v}_j^c + (\sigma_j^c)^2\right)\right], \tag{5}$$

where M is the number of samples of $\mathcal{Y}^\star$, $(\sigma_k^f)^2$ and $(\sigma_j^c)^2$ are the noise variances of the objectives and constraints, respectively, $v_k^f = v_k^f(\mathbf{x})$, $v_j^c = v_j^c(\mathbf{x})$, $\tilde{v}_k^f = \tilde{v}_k^f(\mathbf{x}|\mathcal{Y}_{(m)}^\star)$, $\tilde{v}_j^c = \tilde{v}_j^c(\mathbf{x}|\mathcal{Y}_{(m)}^\star)$ are the variances of the predictive distribution before and after conditioning to $\mathcal{Y}^\star$. This expression is a sum across the objectives and constraints. Therefore, it can be used to identify what black-box to evaluate next in a decoupled evaluation scenario. Note that the acquisition of each black box depends on the other black-boxes (more details, in Appendix C). The cost of evaluating (5) is $\mathcal{O}(M(K+C)|\mathcal{Y}^\star|)$. We approximate $\mathcal{Y}^\star$ using 50 points. Appendix D shows visually the computation of (5).

3 Experiments

We compare MESMOC+ and its decoupled variant MESMOC+$_{\text{dec}}$ with BMOO (Feliot et al., 2017), which is based on EHI, and with PESMOC, MESMOC, and random search (RANDOM). BMOO and PESMOC are provided in the Bayesian optimization software Spearmint. We have also implemented in that software MESMOC+ and MESMOC, closely following the code provided by Belakaria et al. (2021). The code for MESMOC+ can be found at `https://github.com/fernandezdaniel/Spearmint`. Regarding the probabilistic models, we use GPs with a Matérn52 kernel with ARD. To maximize the acquisition function we use L-BFGS and a grid of $d \times 1000$ points to choose a good starting point. The gradients of the acquisition function are approximated by differences. We report average results of each experiment after 100 repetitions. The recommendations of each method are obtained by optimizing the means of the GPs at each iteration. To avoid recommending infeasible solutions we follow Garrido-Merchán and Hernández-Lobato (2019).

We consider two synthetic experiments where the underlying functions of the black-boxes are samples from a GP. The input space of each black-box is the interval $[0, 1]$. We consider two scenarios: one with noiseless observations, and another where the observations are contaminated with standard Gaussian noise with variance 0.1. The performance of each method is measured as the relative difference (in log-scale) of the hyper-volume of the recommendation made and the maximum hyper-volume, as a function of the evaluations made The maximum hyper-volume is obtained by an exhaustive search. The maximum hyper-volume is found using a grid of points. Infeasible recommendations have an associated hyper-volume equal to 0. The first experiment has a 4-dimensional input and the methods have to optimize 2 objectives while fulfill 2 constraints. In this experiment, for learning the hyper-parameters of the GPs (the actual amplitude is 1 and all the length-scales are 1) we use slice sampling with 10 samples. For each sample, MESMOC+, MESMOC and PESMOC generate a different sample of $\mathcal{Y}^\star$, $\mathcal{Y}^\star$ and $\mathcal{X}^\star$, respectively.

The first row of Figure 1 show the results of the first experiment. We observe that the best methods are MESMOC+, PESMOC and PESMOC$_{\text{dec}}$. MESMOC+$_{\text{dec}}$ also achieves good results when there is no noise. In these experiments, MESMOC+ is highly superior to MESMOC, which performs poorly in the noisy settings. MESMOC$_{\text{dec}}$ also performs poorly in general. This is probably as a consequence of the poor approximation of the acquisition function in MESMOC and MESMOC$_{\text{dec}}$. See Appendix E for further details. In Table 1 we display the average time in seconds per iteration of MESMOC+, MESMOC and PESMOC and their decoupled variants in this first experiment. We observe that the times of MESMOC+ and MESMOC+$_{\text{dec}}$ are significantly lower than those of PESMOC and PESMOC$_{\text{dec}}$, respectively, thanks to their cheaper approximation. Regarding MESMOC, it is just a little faster than MESMOC+ but its performance is much worse.

Wang and Jegelka (2017) show that max-value entropy search is more robust than *predictive entropy search* (PES) with respect to the number of samples of the solution of the optimization problem. We check this comparing MESMOC+ and PESMOC in the second experiment (we have not included MESMOC in the comparison because of its bad performance). This experiment has a 6-dimensional input, 4 objectives and 2 constraints. For the robustness comparison, we sample 1, 10 and 100 times $\mathcal{Y}^\star$ and $\mathcal{X}^\star$ for MESMOC+ and PESMOC, respectively. For learning the hyper-parameters of the GPs (again, the actual amplitude is 1 and all the length-scales are 1.5) we maximize

the marginal likelihood, and to avoid over-fitting we use use 20 random initial evaluations of each black-box for each method. The second row of Figure 1 shows the results obtained. We observe a higher robustness of MESMOC+ than PESMOC with respect to M. MESMOC+$_1$ is always better than PESMOC$_1$. As we increase M, the differences among them become smaller. This confirms that MESMOC+ is better than PESMOC+ when M is small.

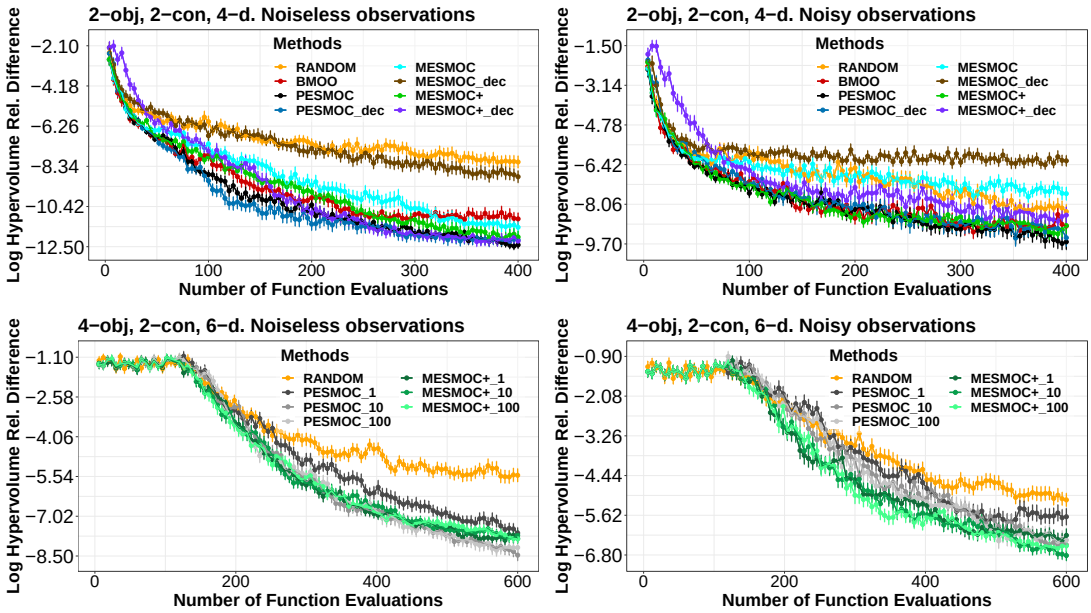

Figure 1: Average log hyper-volume relative difference between the recommendation of each method and the maximum hyper-volume, with respect to the number of evaluations made.

Table 1: Average execution time per iteration (in sec.) in the 4D experiment.

MESMOC+	MESMOC+$_{\text{dec}}$	MESMOC	MESMOC$_{\text{dec}}$	PESMOC	PESMOC$_{\text{dec}}$
12.48±1.15	25.73±3.94	10.34±0.84	12.09±0.83	29.71±3.70	89.33±5.36

4 Conclusions

We have developed MESMOC+, a method for multi-objective Bayesian optimization with constraints. MESMOC+ selects the next point to evaluate as the one that is expected to reduce the most the entropy of the solution of the optimization problem in the function space (*i.e.* the Pareto front $\mathcal{Y}^\star$). Since MESMOC+'s acquisition is expressed as a sum of acquisition functions, one per each different black-box, its computational cost is linear in the number of black-boxes. Moreover, it can be used in a decoupled evaluation setting in which one chooses not only the point at which to evaluate the black-boxes, but also what black-box to evaluate next. MESMOC+ improves the approximation of the acquisition function performed by MESMOC, an already existing acquisition function targeting the reduction of the entropy of the Pareto front. Specifically, the approximation of the acquisition function performed by MESMOC+ is more accurate than that of MESMOC. This is translated in better optimization results. Our experiments show that MESMOC+ is competitive with other state-of-the-art methods for BO, but MESMOC+ is significantly faster to execute. This is a consequence of measuring the expected reduction of the entropy of the Pareto front $\mathcal{Y}^\star$ instead of the Pareto set $\mathcal{X}^\star$. Moreover, MESMOC+ is more robust with respect to the number of Monte Carlo samples of $\mathcal{Y}^\star$ needed to approximate the acquisition function. Finally, we have observed that the decoupled variant of MESMOC+ sometimes obtains better results than the coupled variant.

Acknowledgements. The authors gratefully acknowledge the use of the facilities of Centro de Computación Científica (CCC) at Universidad Autónoma de Madrid. The authors also acknowledge financial support from Spanish Plan Nacional I+D+i, PID2019-106827GB-I00.

References

Ariizumi, R., Tesch, M., Choset, H., and Matsuno, F. (2014). Expensive multiobjective optimization for robotics with consideration of heteroscedastic noise. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2230–2235. IEEE.

Belakaria, S., Deshwal, A., and Doppa, J. R. (2021). Output space entropy search framework for multi-objective bayesian optimization. *Journal of Artificial Intelligence Research*, 72:667–715.

Boyen, X. and Koller, D. (1998). Tractable inference for complex stochastic processes. *International Conference on Uncertainty in Artificial Intelligence*, pages 33–42.

Brochu, E., Cora, V. M., and De Freitas, N. (2009). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *Technical Report TR-2009-023, University of British Columbia*.

Collette, Y. and Siarry, P. (2004). *Multiobjective optimization: principles and case studies*. Springer Science & Business Media.

Daulton, S., Balandat, M., and Bakshy, E. (2020). Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 9851–9864.

Daulton, S., Balandat, M., and Bakshy, E. (2021). Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In *Advances in Neural Information Processing Systems*, volume 34, pages 2187–2200.

Emmerich, M. and Klinkenberg, J.-w. (2008). The computation of the expected improvement in dominated hypervolume of pareto front approximations. *Rapport technique, Leiden University*, 34:7–3.

Feliot, P., Bect, J., and Vazquez, E. (2017). A Bayesian approach to constrained single-and multi-objective optimization. *Journal of Global Optimization*, 67(1-2):97–133.

Garrido-Merchán, E. C. and Hernández-Lobato, D. (2019). Predictive entropy search for multi-objective Bayesian optimization with constraints. *Neurocomputing*, 361:50–68.

Hennig, P. and Schuler, C. J. (2012). Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(6):1809–1837.

Hernández-Lobato, D., Hernandez-Lobato, J. M., Shah, A., and Adams, R. (2016). Predictive entropy search for multi-objective bayesian optimization. In *International Conference on Machine Learning*, pages 1492–1501. PMLR.

Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. (2014). Predictive entropy search for efficient global optimization of black-box functions. *Advances in neural information processing systems*, pages 918–926.

Minka, T. P. (2001). Expectation propagation for approximate bayesian inference. In *Uncertainty in Artificial Intelligence*, volume 17, pages 362–369.

Qing, J., Moss, H. B., Dhaene, T., and Couckuyt, I. (2022). $\{PF\}^2es$: Parallel feasible pareto frontier entropy search for multi-objective bayesian optimization under unknown constraints. *arXiv preprint arXiv:2204.05411*.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian Processes for Machine Learning*. MIT press.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Villemonteix, J., Vazquez, E., and Walter, E. (2009). An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534.

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

A Reproducibility Checklist

1. For all authors...

   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] [Out claims in the abstract and introduction accurately reflect the paper's contributions and scope.]

   (b) Did you describe the limitations of your work? [Yes] [We explain the optimization scenarios where our acquisition function can be used.]

   (c) Did you discuss any potential negative societal impacts of your work? [No] [We do not think that our work has any negative societal impacts.]

   (d) Have you read the ethics author's and review guidelines and ensured that your paper conforms to them? `https://automl.cc/ethics-accessibility/` [Yes] [We have read the ethics author's and review guidelines and we think that our paper conforms to them.]

2. If you are including theoretical results...

   (a) Did you state the full set of assumptions of all theoretical results? [Yes] [Our work doest not include theoretical results. However, we state most of the assumptions made in Section 2.]

   (b) Did you include complete proofs of all theoretical results? [No] [Our work doest not include theoretical results and hence no proofs are needed.]

3. If you ran experiments...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results, including all requirements (e.g., `requirements.txt` with explicit version), an instructive `README` with installation, and execution commands (either in the supplemental material or as a URL)? [Yes] [We include all the code and instructions in the main manuscript to reproduce our experiments. However, we do not include the code of the experiments itself.]

   (b) Did you include the raw results of running the given instructions on the given code and data? [No] [We describe how to implement the experiments and obtain the results, but we do not include raw results.]

(c) Did you include scripts and commands that can be used to generate the figures and tables in your paper based on the raw results of the code, data, and instructions given? [No] [We do not include the scripts and commands that can generate the figures and tables in our paper based on the results. However, they should be easily reproducible given the information provided in the paper.]

(d) Did you ensure sufficient code quality such that your code can be safely executed and the code is properly documented? [No] [We do not ensure that the code can be safely executed nor it is properly documented.]

(e) Did you specify all the training details (e.g., data splits, pre-processing, search spaces, fixed hyperparameter settings, and how they were chosen)? [Yes] [All the empirical details can be found in the experiments Section.]

(f) Did you ensure that you compared different methods (including your own) exactly on the same benchmarks, including the same datasets, search space, code for training and hyperparameters for that code? [Yes] [All the context parameters were exactly the same for all the methods compared.]

(g) Did you run ablation studies to assess the impact of different components of your approach? [No] [Our method does not have different components. We investigated in the Appendix the quality of the approximation considered]

(h) Did you use the same evaluation protocol for the methods being compared? [Yes] [The evaluation protocol is exactly the same for all the methods.]

(i) Did you compare performance over time? [Yes] [It can be found in the experiments Section.]

(j) Did you perform multiple runs of your experiments and report random seeds? [Yes] [All the results reported are an average across 100 repetitions of the experiments.]

(k) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] [All the results reported have error bars.]

(l) Did you use tabular or surrogate benchmarks for in-depth evaluations? [No] [We do not use tabular or surrogate benchmarks in our experiments.]

(m) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] [We do not include information about the resources used.]

(n) Did you report how you tuned hyperparameters, and what time and resources this required (if they were not automatically tuned by your AutoML method, e.g. in a NAS approach; and also hyperparameters of your own method)? [Yes] [We give specific details about how we select the hyper-parameters.]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] [We cite the creators.]

(b) Did you mention the license of the assets? [No] [We do not mention the license of the assets.]

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] [We include a link to the new assets in the paper.]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] [We do not discuss the consent of the data used since we used synthetic data.]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] [The data we are using are synthetic and does not contain personally identifiable information or offensive content.]

B Limitations of MESMOC

Max-value entropy search for multi-objective optimization with constraints (MESMOC) is an acquisition function developed independently (Belakaria et al., 2021), which also minimizes the entropy of $\mathcal{Y}^\star$, as MESMOC+ does. Thus, the expression considered by MESMOC for the acquisition function is also given by Eq. (2). However, the proposed approximation for the entropy of $p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{Y}^\star)$ is very different. In Belakaria et al. (2021), they focus on maximizing the objectives, while we are interested in minimization. Consider a maximization scenario. Ignore the constraints initially. Let the sampled Pareto frontier be $\mathcal{Y}^\star = \{\mathbf{z}_1, \ldots, \mathbf{z}_S\}$ with S the size of $\mathcal{Y}^\star$ and each $\mathbf{z}_j$, for $j = 1, \ldots, S$, a vector of size K with the associated objective values. Let the j-th component of $\mathbf{z}_i$ be z_i^j. Belakaria et al. (2021) argue that a sufficient condition for some point $\mathbf{y}$ being compatible with $\mathcal{Y}^\star$ as the solution of the problem is that $y^j \leq \max\{z_1^j, \ldots, z_S^j\}\ \forall j \in \{1, \ldots, K\}$ (recall a maximization scenario). That is, the value of $\mathbf{y}$ for the j-th objective cannot be larger than the largest value for that objective, according to $\mathcal{Y}^\star$. However, this condition is not complete because $\mathbf{y}$ can be optimal, (*i.e.*, $\mathbf{y}$ is incompatible with $\mathcal{Y}^\star$) even if none of its values are larger than the maximum value for the corresponding objective. E.g., let $K = 2$ and $\mathcal{Y}^\star = \{(1, 0), (0, 1)\}$. Consider now the point $\mathbf{y} = (0.7, 0.7)$. The components of $\mathbf{y}$ are smaller than $1 = \max\{z_1^j, \ldots, z_S^j\} \forall j \in \{1, \ldots, K\}$, but this point is optimal and non-dominated by any of the points in $\mathcal{Y}^\star$. This means that $\mathbf{y}$ is not compatible with $\mathcal{Y}^\star$. However, the condition employed in Belakaria et al. (2021), *i.e.*, that $\mathbf{y}$ must satisfy $y_j \leq \max\{z_1^j, \ldots, z_S^j\}\ \forall j \in \{1, \ldots, K\}$, will consider $(0.7, 0.7)$ as a potential point to be predicted by the conditional predictive distribution given $\mathcal{Y}^\star$, $p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{Y}^\star)$, which is incorrect. The constraints are incorporated in Belakaria et al. (2021) in an ad-hoc way, simply by enforcing that $c_j(\mathbf{x}) \leq \max\{\tilde{z}_1^j, \ldots, \tilde{z}_S^j\}$ for $j = 1, \ldots, C$, where $\{\tilde{\mathbf{z}}_i\}_{i=1}^S$ are the constraint values associated to the points in $\mathcal{Y}^\star$. That is, the constraint values have to be smaller than the maximum constraint values associated to the Pareto front $\mathcal{Y}^\star$, as it is done with the objectives. There is no justification for this.

Summing up, the conditional predictive distribution is given by a factorizing truncated Gaussian distribution. In particular, for the objectives each $f_j(\mathbf{x}) \leq \max\{z_1^j, \ldots, z_S^j\}\ \forall j \in \{1, \ldots, K\}$, and for the constraints, each $c_j(\mathbf{x}) \leq \max\{\tilde{z}_1^j, \ldots, \tilde{z}_S^j\}$ for $j = 1, \ldots, C$. The entropy of a truncated Gaussian distribution has a closed-form expression (Wang and Jegelka, 2017). Therefore, the approximate acquisition function of MESMOC is

$$\alpha(\mathbf{x}) \approx \frac{1}{M} \sum_{m=1}^{M} \left[\sum_{k=1}^{K} \left(\frac{\varrho_{k,m}^f \phi(\varrho_{k,m}^f)}{2\Phi(\varrho_{k,m}^f)} - \log \Phi(\varrho_{k,m}^f) \right) + \sum_{j=1}^{C} \left(\frac{\varrho_{j,m}^c \phi(\varrho_{j,m}^c)}{2(\Phi(\varrho_{j,m}^c))} - \log(\Phi(\varrho_{j,m}^c)) \right) \right], \tag{6}$$

where $\varrho_{k,m}^f = \varrho_{k,m}^f(\mathbf{x})$, $\varrho_{j,m}^c = \varrho_{j,m}^c(\mathbf{x})$,

$$\varrho_{k,m}^f(\mathbf{x}) = \frac{\zeta_{k,m}^\star - m_k^f(\mathbf{x})}{(v_k^f(\mathbf{x}))^{1/2}}, \qquad \varrho_{j,m}^c(\mathbf{x}) = \frac{\varsigma_{j,m}^\star - m_j^c(\mathbf{x})}{(v_j^c(\mathbf{x}))^{1/2}}, \tag{7a,7b}$$

and

$$\zeta_{k,m}^\star = \min\{z_1^{k,m}, \ldots, z_S^{k,m}\}\ \forall k \in \{1, \ldots, K\}, \quad \varsigma_{j,m}^\star = \max\{\tilde{z}_1^{j,m}, \ldots, \tilde{z}_S^{j,m}\}\ \forall j \in \{1, \ldots, C\}, \tag{8}$$

and S the size of $\mathcal{Y}_{(m)}^\star$, *i.e.*, the m-th sample of $\mathcal{Y}^\star$. The samples of $\mathcal{Y}^\star$ are obtained as in MESMOC+. Namely, by optimizing samples from the GP posterior distribution using a random feature approximation. Moreover, $z_s^{k,m}$ is the k-th objective value associated to the s-th point in $\mathcal{Y}_{(m)}^\star$, and $\tilde{z}_s^{j,m}$ is

the j-th constraint value associated to the s-th point in $\mathcal{Y}^{\star}_{(m)}$. Note that (6) also involves a sum of one acquisition function per constraint and objective, which means that MESMOC can be used in a decoupled evaluation setting.

Critically, in the provided code in Belakaria et al. (2021), the implementation considered that $\mathcal{Y}^{\star}_{(m)}$ is given by the Pareto front sample, and all the evaluations performed so far. The consequence is that the acquisition proposed in Belakaria et al. (2021) is simply the sum of the standard MES acquisition function for each objective and constraint (Wang and Jegelka, 2017). This makes sense, and is equivalent to maximizing all the objectives and constraints independently. Maximizing each objective is expected to give good solutions. Maximizing each constraint is expected to provide feasible solutions. This makes, however, difficult finding interesting points of the Pareto frontier that are not close to the maximum values of each objective. Such a strategy is hence expected to be sub-optimal. Besides this, the optimization of the resulting acquisition function in Belakaria et al. (2021) is restricted to those regions of the input space in which the GP means for the constraints are strictly positive. This becomes problematic in problems in which finding feasible points is difficult. In particular, if all the observations are infeasible, the GP means for the constraints will be negative in all the input space (even though the associated GP variance can be high). This will make the BO method to fail. If that is the case, a simple solution is to choose randomly the next point to evaluate.

C Obtaining the Approximate Truncated Gaussians by ADF

C.1 Introduction to ADF

Assumed Density Filtering (ADF) is a technique that is often used to calculate approximate posteriors Boyen and Koller (1998), and in this work, we use it to approximate the predictive distributions conditioned to the Pareto $\mathcal{Y}^{\star}$ front by truncated Gaussians. When ADF is used to approximate a distribution of interest to $p(\mathbf{a})$, a distribution $q(\mathbf{a})$ is first chosen from a family of distributions convenient for us to work with. This $q(\mathbf{a})$ distribution is adjusted to approximate the target distribution $p(\mathbf{a})$. For this, ADF minimizes the Kullback-Leibler divergence between $p(\mathbf{a})$ and $q(\mathbf{a})$, *i.e.* it minimizes $KL(p(\mathbf{a})||q(\mathbf{a}))$. Following (Wang and Jegelka, 2017) we chose a truncated Gaussian for $q(\mathbf{a})$, thus it belongs to the exponential family. Minimizing the divergence of Kullback-Leibler when we are approaching one distribution by another that belongs to the exponential family is equivalent to matching moments between the two distributions. Namely, we are going to adjust the means and variances of several truncated Gaussian distributions to approximate the predictive distributions conditioned to $\mathcal{Y}^{\star}$. This adjustment of means and variances is made while processing the points of a $\mathcal{Y}^{\star}$ sample.

C.2 ADF Updates for MESMOC+

ADF approximates $p(\mathbf{f}, \mathbf{c}|\mathcal{D}, \mathbf{x}, \mathcal{Y}^{\star})$ using a distribution $q(\mathbf{f}, \mathbf{c})$ that is Gaussian. Specifically, ADF minimizes, in an iterative way, the Kullback-Leibler divergence between $\hat{p}(\mathbf{f}, \mathbf{c}|\mathcal{D}, \mathbf{x}, \mathcal{Y}^{\star}) = Z^{-1}\Omega(\mathbf{f}^{\star}, \mathbf{f}, \mathbf{c})p(\mathcal{Y}^{\star}|\mathbf{f}, \mathbf{c})$ and $q(\mathbf{f}, \mathbf{c})$, with respect to q, where Z is a normalization constant, for each $\mathbf{f}^{\star} \in \mathcal{Y}^{\star}$. Minimizing the Kullback-Leibler divergence when the approximate distribution q is Gaussian is equivalent to matching moments between the two distributions, $\hat{p}$ and q (Minka, 2001). Therefore, we are going to adjust the means and covariances of q to have the same mean and covariances as $\hat{p}(\mathbf{f}, \mathbf{c}) = Z^{-1}\Omega(\mathbf{f}^{\star}, \mathbf{f}, \mathbf{c})q(\mathbf{f}, \mathbf{c})$. This process is repeated iteratively for each factor $\Omega(\mathbf{f}^{\star}, \mathbf{f}, \mathbf{c})$. Note that there is one different factor for each $\mathbf{f}^{\star}$ in $\mathcal{Y}^{\star}$. It is possible to understand this process as replacing each $\Omega(\mathbf{f}^{\star}, \mathbf{f}, \mathbf{c})$ using an approximate Gaussian factor (Minka, 2001). Of course, the processing order of each $\Omega(\mathbf{f}^{\star}, \mathbf{f}, \mathbf{c})$ affects the approximation. We simply process the factors in a random order.

Now, we describe how to use ADF to obtain the updates for the means and variances of the resulting Gaussian, after incorporating each $\Omega(\mathbf{f}^{\star}, \mathbf{f}, \mathbf{c})$ factor. Let $q(\mathbf{f}, \mathbf{c}) = p(\mathbf{f}, \mathbf{c}|\mathcal{D}, \mathbf{x})$. That is, we

initialize q to the predictive distribution given by the GP at $\mathbf{x}$. Let $\mathbf{a} = (\mathbf{f}, \mathbf{c})$. We can express $\hat{p}(\mathbf{f}, \mathbf{c})$ as a product of an arbitrary function $t(\mathbf{a})$ multiplied by a Gaussian distribution $\mathcal{N}(\mathbf{a}|\boldsymbol{\mu}, \Sigma)$. Namely,

$$Z = \int t(\mathbf{a})\mathcal{N}(\mathbf{a}|\boldsymbol{\mu}, \Sigma)d\mathbf{a}, \qquad \hat{p}(\mathbf{a}) = Z^{-1}t(\mathbf{a})\mathcal{N}(\mathbf{a}|\boldsymbol{\mu}, \Sigma)\,, \tag{9}$$

where our arbitrary factor $t(\mathbf{a})$ is $\Omega(\mathbf{f}^\star, \mathbf{f}, \mathbf{c})$, and $\mathcal{N}(\mathbf{a}|\boldsymbol{\mu}, \Sigma)$ is simply $q(\mathbf{a})$. with where $\boldsymbol{\mu}$ y Σ are the vector of means and the covariance matrix of $\mathbf{a}$ under q, respectively. Because we assume independence among the GPs, Σ is diagonal.

Then, we can use the following expressions from (Minka, 2001) to refine the mean and variance of de updated q distribution after incorporating $\Omega(\mathbf{f}^\star, \mathbf{f}, \mathbf{c})$:

$$\begin{aligned} \mathbb{E}_{\hat{p}(\mathbf{a})}[\mathbf{a}] &= \boldsymbol{\mu} + \Sigma\frac{\partial \log(Z)}{\partial \boldsymbol{\mu}}\,, \\ \mathbb{E}_{\hat{p}(\mathbf{a})}[\mathbf{a}\mathbf{a}^{\mathrm{T}}] - \mathbb{E}_{\hat{p}(\mathbf{a})}[\mathbf{a}]\mathbb{E}_{\hat{p}(\mathbf{a})}[\mathbf{a}]^{\mathrm{T}} &= \Sigma - \Sigma\left(\frac{\partial \log(Z)}{\partial \boldsymbol{\mu}}\left(\frac{\partial \log(Z)}{\partial \boldsymbol{\mu}}\right)^{\mathrm{T}} - 2\frac{\partial \log(Z)}{\partial \Sigma}\right)\Sigma\,. \end{aligned} \tag{10}$$

Hence, to use ADF, we need to compute Z and the partial derivatives of $\log(Z)$ respect to the means and variances of the objectives and constraints at $\mathbf{x}$. The computation of Z is as follows:

$$\begin{aligned} Z &= \int p(\mathbf{f}, \mathbf{c}|\mathcal{D}, \mathbf{x})\Omega(\mathbf{f}^\star, \mathbf{f}, \mathbf{c})d\mathbf{f}d\mathbf{c} \\ &= \int p(\mathbf{f}, \mathbf{c}|\mathcal{D}, \mathbf{x})\left(1 - \prod_{j=0}^{C}\Theta(c_j(\mathbf{x}))\prod_{k=0}^{K}\Theta\left(f_k^\star - f_k(\mathbf{x})\right)\right)d\mathbf{f}d\mathbf{c} \\ &= 1 - \prod_{j=0}^{C}\Phi(\gamma_j^c)\prod_{k=0}^{K}\Phi(\gamma_k^f)\,, \end{aligned} \tag{11}$$

where $\Phi(\cdot)$ is the cumulative probability distribution of Gaussian, and:

$$\gamma_k^f = \frac{f_k^\star - m_k^f(\mathbf{x})}{(v_k^f(\mathbf{x}))^{1/2}}\,, \qquad \gamma_j^c = \frac{m_j^c(\mathbf{x})}{(v_j^c(\mathbf{x}))^{1/2}}\,, \tag{11a,11b}$$

where $m_k^f(\mathbf{x})$, $v_k^f(\mathbf{x})$, $m_k^c(\mathbf{x})$ and $v_k^c(\mathbf{x})$ are the mean and variance values of the k-th and j-th predictive distributions of the objectives and constraints at $\mathbf{x}$. At the same time, these are the values of the derivatives $\frac{\partial \log(Z)}{\partial m_k^f}$, $\frac{\partial \log(Z)}{\partial v_k^f}$, $\frac{\partial \log(Z)}{\partial m_j^c}$ and $\frac{\partial \log(Z)}{\partial v_j^c}$:

$$\frac{\partial \log(Z)}{\partial m_k^f} = \frac{(Z-1)}{Z\Phi(\gamma_k^f)}\left(\frac{-\mathcal{N}(\gamma_k^f|0,1)}{(v_k^f)^{1/2}}\right), \qquad \frac{\partial \log(Z)}{\partial v_k^f}\,, = \frac{(Z-1)}{Z\Phi(\gamma_k^f)}\left(\mathcal{N}(\gamma_k^f|0,1)\frac{-\gamma_k^f}{2v_k^f}\right), \tag{12}$$

$$\frac{\partial \log(Z)}{\partial m_j^c} = \frac{(Z-1)}{Z\Phi(\gamma_j^c)}\left(\frac{\mathcal{N}(\gamma_j^c|0,1)}{(v_j^c)^{1/2}}\right), \qquad \frac{\partial \log(Z)}{\partial v_j^c} = \frac{(Z-1)}{Z\Phi(\gamma_j^c)}\left(\mathcal{N}(\gamma_j^f|0,1)\frac{-\gamma_j^c}{2v_j^c}\right), \tag{13}$$

where $v_k^f = v_k^f(\mathbf{x})$ and $v_j^c = v_j^c(\mathbf{x})$.

Algorithm 1 shows the ADF steps for computing the mean and variance of the approximate conditional predictive distribution. We can see that in each iteration the values of $\tilde{\mathbf{m}}^f$, $\tilde{\mathbf{v}}^f$, $\tilde{\mathbf{m}}^c$ and $\tilde{\mathbf{v}}^c$ are updated using the values calculated in the derivatives of Eq. (12) and Eq. (13). We can also notice that the order of processing the $\mathbf{f}^\star$ points of the Pareto front sample will influence the final

result for the means and variances of the approximate conditional predictive distribution. For this reason, we have established this order as random.

Algorithm 1: ADF Algorithm

Input: $\mathbf{m}^f$, $\mathbf{v}^f$, $\mathbf{m}^c$ and $\mathbf{v}^c$

Initialize: $\tilde{\mathbf{m}}^f = \mathbf{m}^f$, $\tilde{\mathbf{v}}^f = \mathbf{v}^f$, $\tilde{\mathbf{m}}^c = \mathbf{m}^c$ and $\tilde{\mathbf{v}}^c = \mathbf{v}^c$

for each $\mathbf{f}^\star$ **in** $\mathcal{Y}^\star$ **do**

$\boldsymbol{\gamma}^f = (\mathbf{f}^\star - \mathbf{m}^f)/\sqrt{\mathbf{v}^f}$

$\boldsymbol{\gamma}^c = \mathbf{m}^c/\sqrt{\mathbf{v}^c}$

$Z = 1 - \prod_{j=0}^{C} \Phi(\gamma_j^c) \prod_{k=0}^{K} \Phi(\gamma_k^f)$

$\tilde{\mathbf{m}}^f = \tilde{\mathbf{m}}^f + \tilde{\mathbf{v}}^f \frac{\partial \log(Z)}{\partial \tilde{\mathbf{m}}^f}$

$\tilde{\mathbf{v}}^f = \tilde{\mathbf{v}}^f - \tilde{\mathbf{v}}^f \left(\frac{\partial \log(Z)}{\partial \tilde{\mathbf{m}}^f} \left(\frac{\partial \log(Z)}{\partial \tilde{\mathbf{m}}^f} \right)^{\mathrm{T}} - 2 \frac{\partial \log(Z)}{\partial \tilde{\mathbf{v}}^f} \right) \tilde{\mathbf{v}}^f$

$\tilde{\mathbf{m}}^c = \tilde{\mathbf{m}}^c + \tilde{\mathbf{v}}^c \frac{\partial \log(Z)}{\partial \tilde{\mathbf{m}}^f}$

$\tilde{\mathbf{v}}^c = \tilde{\mathbf{v}}^c - \tilde{\mathbf{v}}^c \left(\frac{\partial \log(Z)}{\partial \tilde{\mathbf{m}}^c} \left(\frac{\partial \log(Z)}{\partial \tilde{\mathbf{m}}^c} \right)^{\mathrm{T}} - 2 \frac{\partial \log(Z)}{\partial \tilde{\mathbf{v}}^c} \right) \tilde{\mathbf{v}}^c$

return $\tilde{\mathbf{m}}^f$, $\tilde{\mathbf{v}}^f$, $\tilde{\mathbf{m}}^c$ and $\tilde{\mathbf{v}}^c$;

D The MESMOC+ Acquisition Function in a toy problem

Figure 2 shows the execution of MESMOC+ in a toy problem with two objectives and one constraint and one-dimensional input. We consider a single sample of $\mathcal{Y}^\star$. The first column displays the current state of the GP predictive distribution for the objectives and the constraints. We show the predictive mean alongside with a one standard deviation confidence interval. The second column shows a sample of each black-box function and the corresponding Pareto front. The points of the Pareto front dominate all other points for which the corresponding values of the constraint are positive. In the third column of this figure, we display the predictive distribution of the black-boxes after conditioning to Pareto front sample (shown in the second column) using ADF. The fourth column is the resulting acquisition function of MESMOC+, for each black-box. This acquisition function is the absolute difference of the predictive variances before and after conditioning (shown in the first column and third column, respectively). So, the regions where this difference is greatest are where MESMOC+ expects to gain the most information (in terms of variance reduction) about the solution to the problem.

Figure 3 show the acquisition function obtained in the problem of Figure 2, but in a coupled evaluation scenario. Therefore, the acquisition in Figure 3 is simply the sum of the acquisition of all the black-boxes in the fourth column of Figure 2. We can notice that the maximizer in Figure 3 is different from the individual maximizers of the black-boxes. We can also observe that the maximum in the coupled setting is larger than the individual maximum of the decoupled setting. However, the sum of the individual maximums is larger than the maximum of the coupled setting. Therefore, we expect that a decoupled evaluation setting will be more useful to gain information about the solution to the optimization problem.

E Quality of the Approximation of the Acquisition Function

We compare in a simple problem the acquisition function of MESMOC+ and MESMOC with the exact acquisition function described in Eq. (2). Since, the problem considered has only two objectives and one constraint, in this setting, quadrature methods are feasible to evaluate the entropy of $p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{Y}^\star)$ (but with a much higher computational cost than MESMOC+ and MESMOC). A quadrature method is expected to provide an approximation that is almost equal to that of the

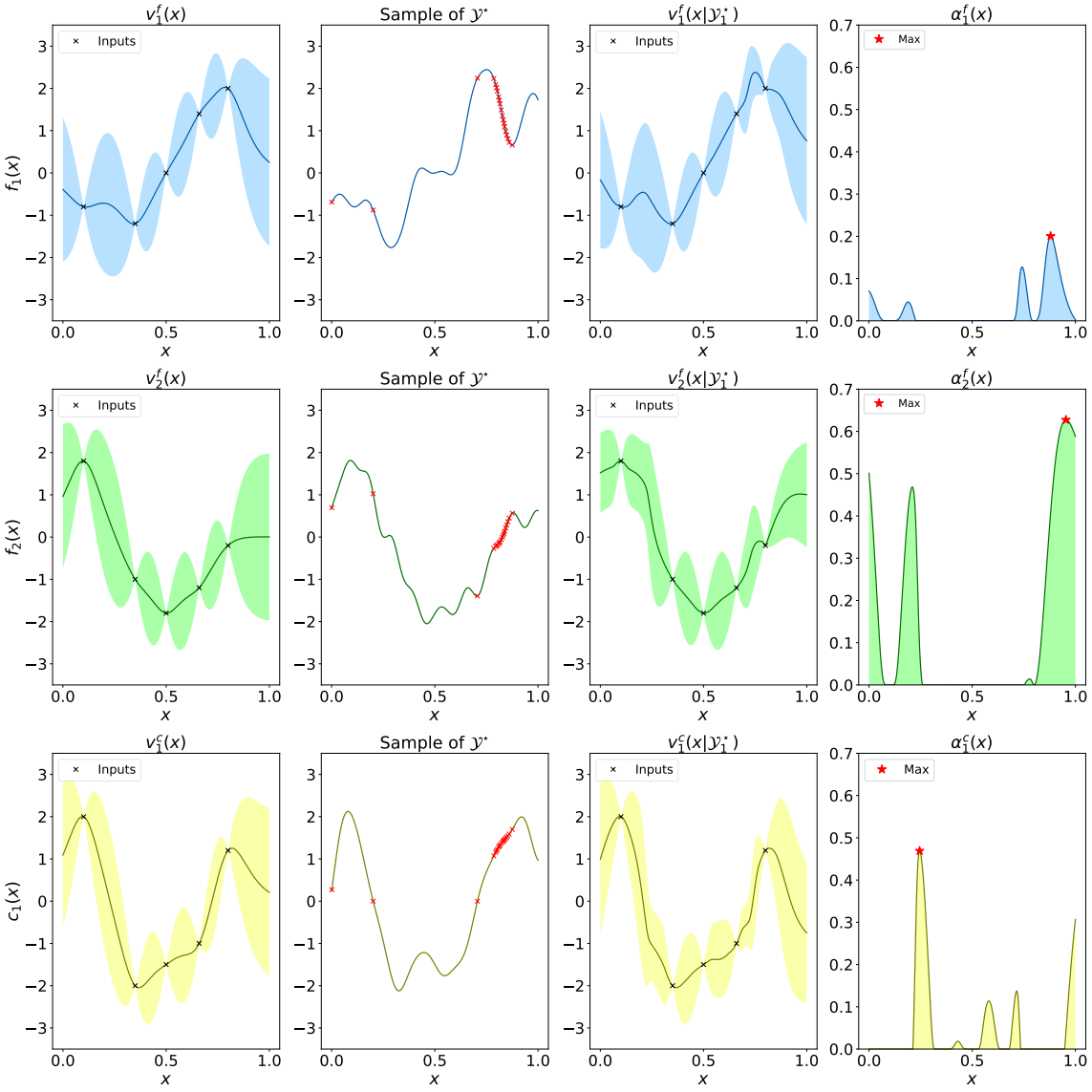

Figure 2: Different steps needed to compute MESMOC+'s acquisition function in a decoupled setting for two objectives and one constraint. (First column) Predictive distribution for each black-box function conditioned the current inputs, using a GP as probabilistic model. (Second column) Samples of each black-box and corresponding Pareto front $\mathcal{Y}^\star_{(m)}$ in the feasible space displayed using red crosses. (Third column) Predictive distribution of each black-box function conditioned to $\mathcal{Y}^\star_{(m)}$ being the solution to the optimization problem. (Fourth column) Acquisition function MESMOC+'s obtained by the absolute difference in the predictive variances before and after the conditioning.

exact acquisition. To compute the entropy of the coupled example in Figure 4 we have solved the following integral by quadrature:

$$H(\mathbf{f}, c|\mathcal{D}, \mathbf{x}, \mathcal{Y}^\star) = -\int_{-\infty}^{\infty} Z^{-1} p(\mathbf{f}, c|\mathcal{D}, \mathbf{x}) p(\mathcal{Y}^\star|\mathbf{f}, c) \log(Z^{-1} p(\mathbf{f}, c|\mathcal{D}, \mathbf{x}) p(\mathcal{Y}^\star|\mathbf{f}, c)) d\mathbf{f} dc, \qquad (14)$$

where $\mathbf{f} = (f_1, f_2)$, $f_1 = f_1(\mathbf{x}))$, $f_2 = f_2(\mathbf{x}))$ and $c = c(\mathbf{x})$.

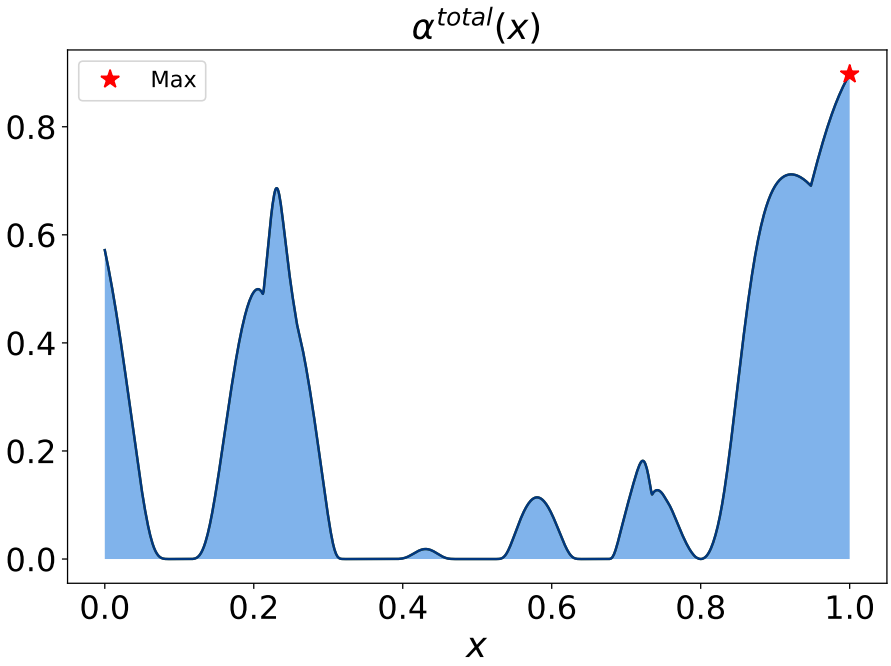

Figure 3: Acquisition function of MESMOC+ for the coupled setting. In this case $\alpha(\cdot)$ is simply the sum of the acquisition functions of the three black-boxes shown in Figure 2.

To compute the entropy in the decoupled case, first, we have marginalized the two random variables corresponding to the black-boxes for which we are not computing the entropy. Next, we computed the entropy of the black box of interest. We obtained the entropies of the black-boxes of the decoupled example of Fig. 5 using quadrature to solve these integrals:

$$\Upsilon_c = \int_{-\infty}^{\infty} Z^{-1} p(\mathbf{f}, c|\mathcal{D}, \mathbf{x}) p(\mathcal{Y}^\star|\mathbf{f}, c) d\mathbf{f}, \quad H(c|\mathbf{f}, \mathcal{D}, \mathbf{x}, \mathcal{Y}^\star) = -\int_{-\infty}^{\infty} Z^{-1} \Upsilon_c \log(\Upsilon_c) dc \tag{15}$$

$$\Upsilon_{f_1} = \int_{-\infty}^{\infty} Z^{-1} p(\mathbf{f}, c|\mathcal{D}, \mathbf{x}) p(\mathcal{Y}^\star|\mathbf{f}, c) df_2 dc, \quad H(f_1|f_2, c, \mathbf{f}, \mathcal{D}, \mathbf{x}, \mathcal{Y}^\star) = -\int_{-\infty}^{\infty} Z^{-1} \Upsilon_{f_1} \log(\Upsilon_{f_1}) df_1 \tag{16}$$

$$\Upsilon_{f_2} = \int_{-\infty}^{\infty} Z^{-1} p(\mathbf{f}, c|\mathcal{D}, \mathbf{x}) p(\mathcal{Y}^\star|\mathbf{f}, c) df_1 dc, \quad H(f_2|f_1, c, \mathcal{D}, \mathbf{x}, \mathcal{Y}^\star) = -\int_{-\infty}^{\infty} Z^{-1} \Upsilon_{f_2} \log(\Upsilon_{f_2}) df_2 \tag{17}$$

The left column of Figure 4 shows the current state of the predictive distributions for the objectives and constraints. The right column shows the sum of the acquisition function for MESMOC and MESMOC+. For MESMOC+, we show the results for the proposed method and the method that considers the logarithm of the variance described. We call this method MESMOC+$_{\log}$. Lastly, we also show the results of the quadrature method (Exact) using adaptative quadrature. We can notice that the approximation of MESMOC+$_{\log}$ seems to be the most accurate, closely followed by MESMOC+. By contrast, the approximation of MESMOC does not look similar to the exact acquisition. Furthermore, MESMOC manually imposes that the acquisition will have a negative value at any point where the mean GP of any of the constraints is negative. This causes it to have a very different value from the exact acquisition in those regions of the input space.

In Figure 5 we make a comparison of the quality of the approximation in a decoupled scenario. The figure on the top-left of Figure 4 shows the current observations and predictive distributions for the objectives and constraints. The remaining figures show the acquisition function for MESMOC,

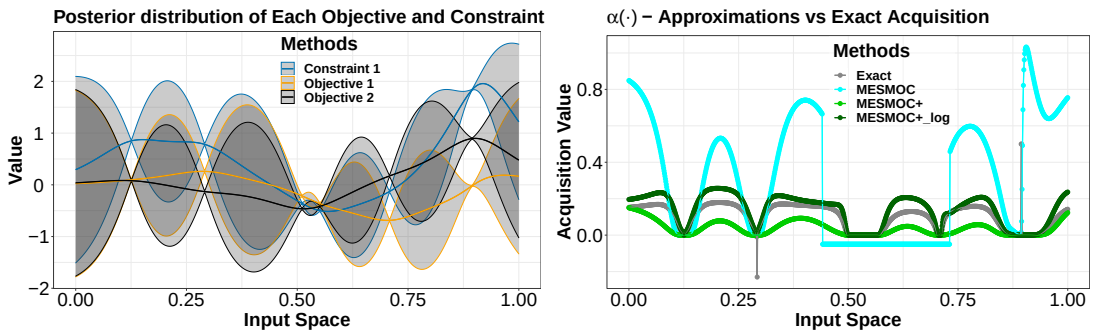

Figure 4: (left) GP predictive distributions for the objectives and constraints. (right) The corresponding estimated acquisition function of each method, MESMOC+, MESMOC+$_{\log}$ and MESMOC, and the exact acquisition (Exact). Best seen in color.

MESMOC+ and MESMOC+$_{\log}$, and the results of the quadrature method (Exact). We can observe that MESMOC+ and MESMOC+$_{\log}$ are very similar to the exact acquisition, for all the black-box functions. Specifically, where MESMOC+ and MESMOC+$_{\log}$ take large values, the exact acquisition takes large values, and where the exact acquisition decreases, MESMOC+ and MESMOC+$_{\log}$ also decrease. By contrast, MESMOC's approximation is dissimilar to the exact acquisition, for all the black-boxes. Furthermore, since MESMOC aims to maximize the constraints, it sometimes gives importance to evaluating the constraints in regions where there may be no expected benefit at all. This behavior can be problematic in the decoupled scenario, where MESMOC can waste useful evaluations simply on trying to maximize a constraint that is already feasible. Figure 5 shows precisely this behavior.

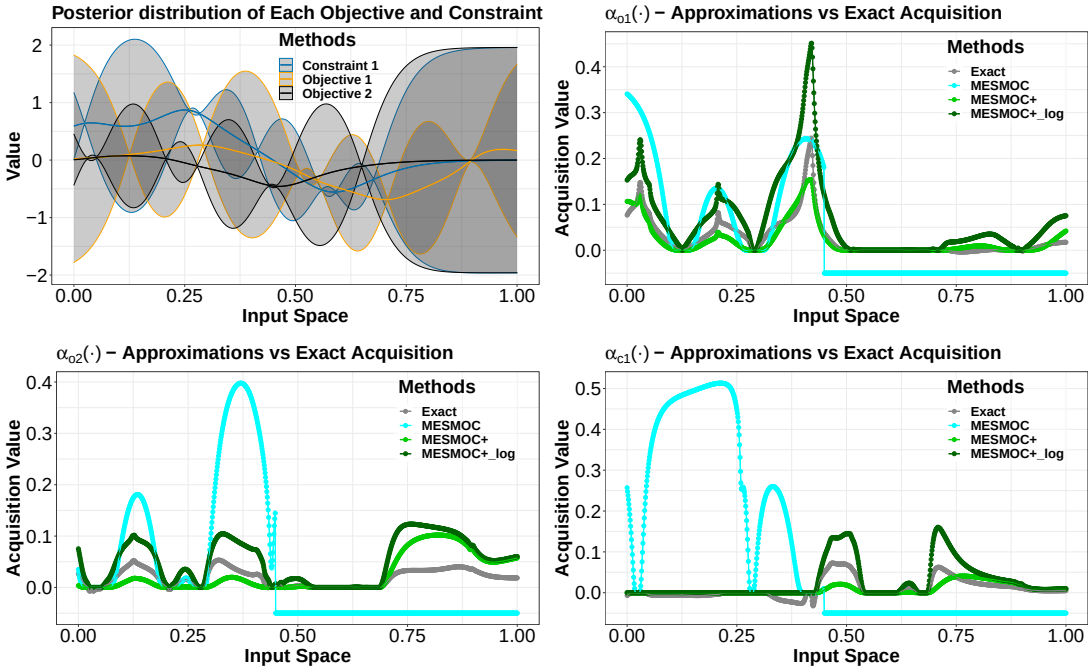

Figure 5: (Top left) GP predictive distributions for the objectives and constraints. (Top right to bottom right) The estimated acquisition function of each method, MESMOC+, MESMOC+$_{\log}$ and MESMOC, and the exact acquisition (Exact) for each black-box. Best seen in color.