Supplementary Material for: Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints.

Eduardo C. Garrido-Merchán

Computer Science Department
Universidad Autónoma de Madrid
Francisco Tomás y Valiente 11, Madrid, Spain

Daniel Hernández-Lobato

Computer Science Department
Universidad Autónoma de Madrid
Francisco Tomás y Valiente 11, Madrid, Spain

Email addresses: eduardo.garrido@uam.es (Eduardo C. Garrido-Merchán), daniel.hernandez@uam.es (Daniel Hernández-Lobato)

URL: http://arantxa.ii.uam.es/~egarrido/ (Eduardo C. Garrido-Merchán), https://dhnzl.org/ (Daniel Hernández-Lobato)

Supplementary Material for: Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints.

Eduardo C. Garrido-Merchán

Computer Science Department
Universidad Autónoma de Madrid
Francisco Tomás y Valiente 11, Madrid, Spain

Daniel Hernández-Lobato

Computer Science Department
Universidad Autónoma de Madrid
Francisco Tomás y Valiente 11, Madrid, Spain

1. Description of Expectation Propagation

In order to get the approximation to the PESMOC acquisition described in this work, it is necessary to compute a Gaussian Approximation to the Conditional Predictive Distribution using the *Expectation Propagation* (EP) algorithm proposed by Minka [3]. This supplementary material explains in detail the operations, performed by the EP algorithm, required to obtain the approximation to the Conditional Predictive Distribution mentioned before.

EP is a widely used technique in Approximate Inference for obtaining Gaussian Distributions from intractable distributions such as the one presented in this work. The EP algorithm usually approximates a product of intractable exact factors (unknown analytic-form distributions) with Gaussian Distributions. The EP algorithm generates one Gaussian Distribution for every intractable individual factor. As the product operation is closed under Gaussian distributions, the product of Gaussian distributions is also a Gaussian Distribution, which is the final result of the EP algorithm.

Email addresses: eduardo.garrido@uam.es (Eduardo C. Garrido-Merchán), daniel.hernandez@uam.es (Daniel Hernández-Lobato)

URL: http://arantxa.ii.uam.es/~egarrido/ (Eduardo C. Garrido-Merchán), https://dhnzl.org/ (Daniel Hernández-Lobato)

This idea can be applied for the Bayes Theorem, where the prior distribution is multiplicated by a product of Gaussian Distributions, that represents the likelihood of the data. As the product of Gaussian Distribution factorizes, the posterior of the data is also Gaussian, because of the closed product operation. In this case, the prior is said to be conjugate. EP can be applied if the likelihood or the prior are not Gaussian Distributions. In this case, the EP factors usually represent one single prior and multiple likelihood factors. There exist alternatives to the EP algorithm such as Variational Inference or the Laplace Approximation, which behave differently. For example, the Laplace Approximation fits a single Gaussian Distribution to the whole posterior, failing hence to capture the possible asymmetry of the posterior distribution.

EP approximates every factor by minimizing the Kullback-Leibler (KL) divergences between each exact factor and the approximated one. In other words, it matches the first and second moments between the exact and the approximate factors. But, as it is interesting to have a good approximation in regions where the overall posterior probability is high, EP performs the moment matching in the *context* of all the approximate factors that are present in the mentioned product. This intuitive idea, where all the factors collaborate in the approximation, is expressed by the following equations. If we are interested in approximating the distribution

$$q(\boldsymbol{x}) = \prod_{n=1}^{N} q_n(\boldsymbol{x}) \,, \tag{1}$$

where each $q_n(\boldsymbol{x})$ represents one of the exact mentioned factors which product gives the distribution that will be approximated. EP approximates this product by

$$\tilde{q}(\boldsymbol{x}) = \frac{1}{Z} \prod_{n=1}^{N} \tilde{q}_n(\boldsymbol{x}) \,. \tag{2}$$

where Z is a constant that represents a normalization quantity for the distribution, each $\tilde{q}_n(\boldsymbol{x})$ represents a Gaussian Distribution with specific parameters which product gives an approximation to the original distribution $q(\boldsymbol{x})$. The EP algorithm approximates each of the factors $q_n(\boldsymbol{x})$, by an approximate factor $\tilde{q}_n(\boldsymbol{x})$, given the context of $\tilde{q}_n(\boldsymbol{x})$. This context is represented by the *cavity* distribution $\tilde{q}^{n}(\boldsymbol{x})$, which is the full approximated distribution minus the factor $\tilde{q}_n(\boldsymbol{x})$, the corresponding approximate factor of $q_n(\boldsymbol{x})$. That is:

$$\tilde{q}^{\setminus n}(\boldsymbol{x}) = \prod_{n' \neq n}^{N} \tilde{q}_{n'}(\boldsymbol{x}) = \frac{\tilde{q}(\boldsymbol{x})}{\tilde{q}_n(\boldsymbol{x})}\,. \tag{3}$$

This context will be useful for the computation of the distribution moments done by the EP algorithm. Recall that instead of matching the moments of $q_n(\boldsymbol{x})$ and $\tilde{q}_n(\boldsymbol{x})$, EP matches the moments (which is equivalent to minimizing the KL divergence) of the distributions $q_n(\boldsymbol{x})\tilde{q}^{\setminus n}(\boldsymbol{x})$ and $\tilde{q}_n(\boldsymbol{x})\tilde{q}^{\setminus n}(\boldsymbol{x}) = \tilde{q}(\boldsymbol{x})$, that is, of the approximated and exact factors taken into account the context $\tilde{q}^{\setminus n}(\boldsymbol{x})$. The advantage of this approach is that it makes the approximation quality higher in regions where the approximate distribution $\tilde{q}(\boldsymbol{x})$ is high. The main drawback is that this comes at the price of obtaining less quality in regions where the density probability function $\tilde{q}(\boldsymbol{x})$ is near zero. The moments of the distribution $q_n(\boldsymbol{x})\tilde{q}^{\setminus n}(\boldsymbol{x})$ are computed using the Eqs (5.12) and (5.13) proposed by Minka [3]. The normalization constant for the approximate factors also needs to be computed. This normalization constant is given by the following expression:

$$Z_n = \int q_n(\boldsymbol{x})\tilde{q}^{\setminus n}(\boldsymbol{x})d\boldsymbol{x}\,. \tag{4}$$

With this information, we can compute the approximate factor, just by substracting the cavity to $\tilde{q}(\boldsymbol{x})$, multiplied by the normalization constant:

$$\tilde{q}_n(\boldsymbol{x}) = \frac{\tilde{q}(\boldsymbol{x})}{\tilde{q}^{\setminus n}(\boldsymbol{x})}\,. \tag{5}$$

Then, the algorithm follows an iterative scheme, recomputing all the factors over iterations until some convergence criterion is satisfied. Summing up all the exposed information, the EP algorithm behaves according to the next steps:

1. Initialize the value of the parameters of the approximate factors $\tilde{q}_n(\boldsymbol{x})$.
2. Initialize the posterior approximation by setting $\tilde{q}(\boldsymbol{x}) \approx \prod_{n=1}^{N} \tilde{q}_n(\boldsymbol{x})$. Choose a value for the index variable n.

3. Compute the cavity distribution $\tilde{q}^{\setminus n}(\boldsymbol{x})$ given by Eq. (3) using the formula for dividing Gaussians or the substraction of the Natural Parameters of the Gaussian Distribution.
4. Compute the first and second moments of $q_n(\boldsymbol{x})\tilde{q}^{\setminus n}(\boldsymbol{x})$ using Eqs (5.12) and (5.13) making them match the moments of $\tilde{q}_n(\boldsymbol{x})\tilde{q}^{\setminus n}(\boldsymbol{x})$ of Minka [3].
5. Update $\tilde{q}_n(\boldsymbol{x})$ as the ratio between $\tilde{q}(\boldsymbol{x})$ and the cavity approximate distribution $\tilde{q}^{\setminus n}(\boldsymbol{x})$, using the formula for dividing Gaussians or the substraction of the natural parameters of the Gaussian Distributions.
6. Repeat steps 2 to 5 until convergence.

Once the EP updates are computed, the new approximation $\tilde{q}(\boldsymbol{x})$ is computed, according to Eq. (2), as the product of all the updated factors $\tilde{q}_n(\boldsymbol{x})$ and the normalization constant. These computations may be done using the formula for multiplying Gaussians or the summation of all the natural parameters of the Gaussian Distributions, that represent the approximate factors.

2. The Gaussian Approximation to the Conditional Predictive Distribution

Recall from the main manuscript that, in this work, we wish to approximate the Conditional Predictive Distribution of the set defined by the points $\mathcal{X} = \{\{\boldsymbol{x}_n\}_{n=1}^{N} \cup \mathcal{X}^* \cup \{\boldsymbol{x}\}\}$. This set is, the union between the $\mathcal{N}$ observation points in the input space $\{\boldsymbol{x}_n\}_{n=1}^{N}$, the $\mathcal{M}$ Pareto Set points $\mathcal{X}^*$ and the candidate point $\{\boldsymbol{x}\}$ to be evaluated. The Gaussian Approximation will then be a multivariate Gaussian Distribution over $\mathcal{N}$+$\mathcal{M}$+1 variables. The Conditional Predictive Distribution is given by the following expression

$$p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star) \propto \int p(\mathbf{y}|\mathbf{x}, \mathbf{f}, \mathbf{c}) p(\mathcal{X}^\star|\mathbf{f}, \mathbf{c}) p(\mathbf{f}|\mathcal{D}) p(\mathbf{c}|\mathcal{D}) d\mathbf{f} d\mathbf{c} \tag{6}$$

, where $p(\mathbf{y}|\mathbf{x}, \mathbf{f}, \mathbf{c}) = \prod_{k=1}^{K} \delta(y_k - f_k(\mathbf{x})) \prod_{j=1}^{J} \delta(y_{K+j} - c_j(\mathbf{x}))$ and $p(\mathcal{X}^\star|\mathbf{f}, \mathbf{c})$ is given by

$$p(\mathcal{X}^\star|\mathbf{f}, \mathbf{c}) \propto \prod_{\mathbf{x}^\star \in \mathcal{X}^\star} \left(\left[\prod_{j=1}^{J} \Phi_j(\mathbf{x}^\star) \right] \left[\prod_{\mathbf{x}' \in \mathcal{X}} \Omega(\mathbf{x}', \mathbf{x}^\star) \right] \right) , \tag{7}$$

where $\Phi_j(\mathbf{x}^\star) = \Theta(c_j(\mathbf{x}^\star))$ with $\Theta(\cdot)$ the Heaviside step function, and $\Omega(\mathbf{x}', \mathbf{x}^\star)$ is defined as:

$$\Omega(\mathbf{x}', \mathbf{x}^\star) = \left[\prod_{j=1}^{J} \Theta(c_j(\mathbf{x}')) \right] \psi(\mathbf{x}', \mathbf{x}^\star) + \left[1 - \prod_{j=1}^{J} \Theta(c_j(\mathbf{x}')) \right] \cdot 1 \, , \tag{8}$$

where $\psi(\mathbf{x}', \mathbf{x}^\star)$ is defined as

$$\psi(\mathbf{x}', \mathbf{x}^\star) = 1 - \prod_{k=1}^{K} \Theta(f_k(\mathbf{x}^\star) - f_k(\mathbf{x}')) \,. \tag{9}$$

The last two probability densities, $p(\mathbf{f}|\mathcal{D})$ and $p(\mathbf{c}|\mathcal{D})$, involved in the Conditional Predictive Distribution are potentially infinite-dimensional Gaussians given by the Gaussian Process predictive distributions for the objectives $\mathbf{f}$ and constraints $\mathbf{c}$ values. As these distributions are Gaussian, they do not need to be approximated.

In order to find a Gaussian Approximation to Eq.(4) it is necessary to perform several steps. First of all, we separate the factors that depend and not depend on $\mathbf{x}$ so that they will be approximated separately. By doing this, the factors that depend on $\boldsymbol{x}$ are refined only once by EP and the other factors are refined iteratively by EP until they change no more.

The factors that depend on $\mathbf{x}$ are the Dirac Delta functions that can be replaced by Gaussians with the corresponding noise variance in the noise case and no variance in the noiseless case. As they are Gaussian Distributions, there is nothing to approximate.

The other factors are the ones that do not depend on $\mathbf{x}$. We define the sampled Pareto Set as $\mathcal{X}^* = \{\boldsymbol{x}_1^*, ..., \boldsymbol{x}_M^*\}$ of size $\mathcal{M}$ and the set of $\mathcal{N}$ observations in the input space as $\hat{\mathcal{X}} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ with the corresponding observations of the k-th objective $\boldsymbol{y}_k$ and of the c-th constraint $\boldsymbol{y}_j$. Then, the values of the posterior distributions of the GPs of the objectives and the constraints at that points are defined by $\boldsymbol{f}_k = (f_k(\boldsymbol{x}_1^*), ..., f_k(\boldsymbol{x}_M^*), f_k(\boldsymbol{x}_1), f_k(\boldsymbol{x}_N))^T$ and $\boldsymbol{c}_j = (c_j(\boldsymbol{x}_1^*), ..., c_j(\boldsymbol{x}_M^*), c_j(\boldsymbol{x}_1), c_j(\boldsymbol{x}_N))^T$. If we define $\boldsymbol{f} = \{\boldsymbol{f}_1, ..., \boldsymbol{f}_K\}$ and $\boldsymbol{c} = \{\boldsymbol{c}_1, ..., \boldsymbol{c}_J\}$, let $q(\boldsymbol{f}, \boldsymbol{c})$ be the distribution that we want to approximate, $p(\mathcal{X}^\star|\mathbf{f}, \mathbf{c})p(\mathbf{f}|\mathcal{D})p(\mathbf{c}|\mathcal{D})$, with the factors that do not depend on $\mathbf{x}$, that is:

$$q(\boldsymbol{f}, \boldsymbol{c}) = \prod_{\mathbf{x}^\star \in \mathcal{X}^\star} \left(\left[\prod_{j=1}^{J} \Phi_j(\mathbf{x}^\star) \right] \left[\prod_{\mathbf{x}' \in \mathcal{X}} \Omega(\mathbf{x}', \mathbf{x}^\star) \right] \right) p(\mathbf{f}|\mathcal{D})p(\mathbf{c}|\mathcal{D}) d\mathbf{f} d\mathbf{c} \,. \tag{10}$$

Because Eq.(10) is not tractable, we approximate the normalized version of $q(\boldsymbol{f}, \boldsymbol{c})$ with a product of Gaussians, the Gaussian Approximation to the Conditional Predictive Distribution. Eq.(10) can be expressed as this normalized product:

$$q(\boldsymbol{f}, \boldsymbol{c}) = \frac{1}{Z_q}\left[\prod_{k=1}^{K}\mathcal{N}\big(\boldsymbol{f}_k \mid \boldsymbol{m}_{pred}^{\boldsymbol{f}_k}, \boldsymbol{V}_{pred}^{\boldsymbol{f}_k}\big)\right]\left[\prod_{j=1}^{J}\mathcal{N}\big(\boldsymbol{c}_j \mid \boldsymbol{m}_{pred}^{\boldsymbol{c}_j}, \boldsymbol{V}_{pred}^{\boldsymbol{c}_j}\big)\right]\times$$
$$\prod_{\mathbf{x}^\star \in \mathcal{X}^\star}\left(\left[\prod_{j=1}^{J}\Phi_j(\mathbf{x}^\star)\right]\left[\prod_{\mathbf{x}' \in \mathcal{X}}\Omega(\mathbf{x}', \mathbf{x}^\star)\right]\right), \tag{11}$$

where $\boldsymbol{m}_{pred}^{\boldsymbol{f}_k}$ and $\boldsymbol{V}_{pred}^{\boldsymbol{f}_k}$ are the mean and covariance matrix of the posterior distributions of $\boldsymbol{f}_k$ given the data in $\mathcal{D}$ and $\boldsymbol{m}_{pred}^{\boldsymbol{c}_j}$ and $\boldsymbol{V}_{pred}^{\boldsymbol{c}_j}$ are the mean and covariance matrix of the posterior distribution of $\boldsymbol{c}_j$ given the data in $\mathcal{D}$. These means and variances are computed according to the equations 2.22-2.24 provided by Rasmussen [5]:

$$\begin{aligned}\boldsymbol{m}_{pred}^{\boldsymbol{f}_k} &= \boldsymbol{K}_*^k\big(\boldsymbol{K}^f + v_f^2\mathbb{I}\big)^{-1}\boldsymbol{y}^k\,,\\ \boldsymbol{V}_{pred}^{\boldsymbol{f}_k} &= \boldsymbol{K}_{*,*}^k - \boldsymbol{K}_*^k\big(\boldsymbol{K}^k + v_f^2\mathbb{I}\big)^{-1}\big[\boldsymbol{K}_*^k\big]\,,\end{aligned} \tag{12}$$

where $\boldsymbol{K}_*^k$ is an $(N+1)\times N$ matrix with the prior cross-covariances between elements of $\boldsymbol{f}_k$ and $f_{k,1},...,f_{k,n}$ and $\boldsymbol{K}_{*,*}^k$ is an $(N+1)\times(N+1)$ matrix with the prior covariances between the elements of $\boldsymbol{f}_k$ and v_k is the standard deviation of the additive Gaussian noise in the evaluations of $\boldsymbol{f}_k$. Following the same reasoning, we have that:

$$\begin{aligned}\boldsymbol{m}_{pred}^{\boldsymbol{c}_j} &= \boldsymbol{K}_*^j\big(\boldsymbol{K}^j + v_j^2\mathbb{I}\big)^{-1}\boldsymbol{y}^j\,,\\ \boldsymbol{V}_{pred}^{\boldsymbol{c}_j} &= \boldsymbol{K}_{*,*}^j - \boldsymbol{K}_*^j\big(\boldsymbol{K}^j + v_j^2\mathbb{I}\big)^{-1}\big[\boldsymbol{K}_*^j\big]\,,\end{aligned} \tag{13}$$

where $\boldsymbol{K}_*^j$ is an $(N+1)\times N$ matrix with the prior cross-covariances between elements of $\boldsymbol{c}_j$ and $c_{j,1},...,c_{j,n}$ and $\boldsymbol{K}_{*,*}^j$ is an $(N+1)\times(N+1)$ matrix with the prior covariances between the elements of $\boldsymbol{c}_j$ and v_j is the standard deviation of the additive Gaussian noise in the evaluations of $\boldsymbol{c}_j$.

The other non-Gaussian factors presented in Eq.(11), $\Phi_j(\mathbf{x}^\star)$ and $\Omega(\mathbf{x}', \mathbf{x}^\star)$, are the problematic ones, as they are not Gaussian Distributions. Hence they will be approximated by Gaussians with EP, as will be described in the next sections.

3. Using Expectation Propagation to Approximate the Conditional Predictive Distribution

This section explains how the EP algorithm approximate the previous product of factors, giving a product of Gausssian Distributions which we call the Gaussian Approximation to the Conditional Predictive Distribution, shown in the previous section. As it is a product where different factors are involved, we have to divide the problem in the approximation of the different factors for Gaussian Distributions. These are the $\Phi_j(\boldsymbol{x}^*$ factors and the $\Omega(\mathbf{x}', \mathbf{x}^\star)$ factors, which will be approximated by one-dimensional and two-dimensional Gaussian Distributions respectively.

The factors $\Phi(\boldsymbol{x}^*)$ that represent if a Pareto Set point $\boldsymbol{x}^*$ is feasible evaluated in a certain constraint $c_j(\boldsymbol{x}^*)$, are approximated by a one-dimensional un-normalized Gaussian distribution $\tilde{\Phi}(\boldsymbol{x}^*)$. This distribution is expressed in exponential family form in the next equation:

$$\Phi(\boldsymbol{x}^*) \approx \tilde{\Phi}(\boldsymbol{x}^*) \propto \exp\left\{ -\frac{c_j(\boldsymbol{x}^*)^2 \hat{v}_j^{\boldsymbol{x}^*}}{2} + c_j(\boldsymbol{x}^*)\hat{m}_j^{\boldsymbol{x}^*} \right\}, \tag{14}$$

where $\hat{v}_j^{\boldsymbol{x}^*}$ and $\hat{m}_j^{\boldsymbol{x}^*}$ are natural parameters that are going to be adjusted by EP. The variance of the Gaussian Distribution, $\hat{v}_j^{\boldsymbol{x}^*}$, EP factor in every point, $\boldsymbol{x}^*$, for every constraint, c_j will be denoted by $\hat{e}_j$ and the mean EP factor by $\hat{f}_j$. That is, the one-dimensional Gaussian Distribution approximation of $\Phi(\boldsymbol{x}^*)$, in every constraint c_j computed by EP, $\tilde{\Phi}(\boldsymbol{x}^*)$ is defined in every point $\boldsymbol{x}^*$ belonging to $\mathcal{X}^*$, by its mean $\hat{f}_j$ and its variance $\hat{e}_j$. There will be as many Gaussian Distributions as points multiplied by constants.

The factors $\Omega(\cdot, \cdot)$, that represent if a point $\boldsymbol{x}_j$ is not dominated by the other point $\boldsymbol{x}_i$ and it is feasible over all the constraints $\boldsymbol{c(x_j)}$, are approximated by a product of $\mathcal{J}$ one-dimensional un-normalized Gaussian Distributions where $\mathcal{J}$ are the number of constraints and $\mathcal{K}$ two-dimensional un-normalized Gaussian Distributions where $\mathcal{K}$ are the number of objectives. This product of distributions is expressed by the following equation:

$$\Omega(\boldsymbol{x}', \boldsymbol{x}^*) \approx \tilde{\Omega}(\boldsymbol{x}', \boldsymbol{x}^*) \propto \prod_{k=1}^{K} \exp\left\{ -\frac{1}{2}\boldsymbol{v}_k^T \tilde{\boldsymbol{V}}_k^{\Omega} \boldsymbol{v}_k + (\tilde{\boldsymbol{m}}_k^{\Omega})^T \boldsymbol{v}_k \right\} \times$$
$$\prod_{j=1}^{J} \exp\left\{ -\frac{c_j(\boldsymbol{x}^*)^2 \tilde{v}_j^{\Omega}}{2} + c_j(\boldsymbol{x}^*)\tilde{m}_j^{\Omega} \right\}, \qquad (15)$$

where $\boldsymbol{v}_k$ is defined as the vector $(f_k(\boldsymbol{x}'), f_k(\boldsymbol{x}^*))^T$, and $\tilde{\boldsymbol{V}}_k$, $\tilde{\boldsymbol{m}}_k$, $\tilde{v}_j^{\Omega}$ and $\tilde{m}_j^{\Omega}$ are natural parameters adjusted by EP. As the product represents a product of two-dimensional unnormalized Gaussian Distributions, $\tilde{\boldsymbol{V}}_k$ is a 2×2 matrix and $\tilde{\boldsymbol{m}}_k$ is a two-dimensional vector.

For the set of $\mathcal{N}$ observation points in the input space $\hat{\mathcal{X}}$ and the set of $\mathcal{M}$ Pareto Set points $\mathcal{X}^*$, we define the variance of the two-dimensional Gaussian Distribution, $\tilde{\boldsymbol{V}}_k$, EP factor of an observation point $\boldsymbol{x}_i$ with respect to a Pareto Point $\boldsymbol{x}_j$ as $\hat{\boldsymbol{A}}_{ij}$ and the mean EP factor as $\hat{\boldsymbol{b}_{ij}}$. We denote the variance of the one-dimensional Gaussian Distribution, $\tilde{v}_j^{\Omega}$, EP factor in every point $\boldsymbol{x}_j$ as $\hat{a}c_j$ and the mean EP factor by $\hat{b}c_j$.

For the set of $\mathcal{M}$ Pareto Set points $\mathcal{X}^*$, we define the variance of the two-dimensional Gaussian Distribution, $\tilde{\boldsymbol{V}}_k$, EP factor of a point $\boldsymbol{x}_i$ with respect to another Pareto Point $\boldsymbol{x}_j$ as $\hat{\boldsymbol{C}}_{ij}$ and the mean EP factor as $\hat{\boldsymbol{d}}_{ij}$. We denote the variance of the one-dimensional Gaussian Distribution $\tilde{v}_j^{\Omega}$ EP factor in every point $\boldsymbol{x}_j$ as $\hat{c}c_j$ and the mean EP factor by $\hat{d}c_j$.

That is, the approximation $\tilde{\Omega}(\boldsymbol{x}', \boldsymbol{x}^*)$ computed by EP consisting of a product of one-dimensional Gaussian Distributions and two-dimensional Gaussian distributions of the distribution $\Omega(\boldsymbol{x}', \boldsymbol{x}^*)$, is defined in the set of points $\hat{\mathcal{X}}$ and $\mathcal{X}^*$ by a product of one-dimensional Gaussian Distributions with mean $\hat{b}c_j$ and variance $\hat{a}c_j$ and a product of two-dimensional Gaussian Distributions with variance $\hat{\boldsymbol{A}}_{ij}$ and mean $\hat{\boldsymbol{b}}_{ij}$. The approximation for the set of points $\mathcal{X}^*$ is defined by a product of one-dimensional Gaussian Distributions with mean $\hat{d}c_j$ and variance $\hat{c}c_j$ and a product of two-dimensional Gaussian Distributions with variance $\hat{\boldsymbol{C}}_{ij}$ and mean $\hat{\boldsymbol{d}}_{ij}$.

In the next section, the computations of the Gaussian factor approximations $\tilde{\Phi}(\cdot)$ and $\tilde{\Omega}(\cdot, \cdot)$ defined by the EP factors $\hat{\boldsymbol{A}}_{ij}$, $\hat{\boldsymbol{b}}_{ij}$, $\hat{\boldsymbol{C}}_{ij}$, $\hat{\boldsymbol{d}}_{ij}$, $\hat{e}_j$, $\hat{f}_j$, $\hat{a}c_j$, $\hat{b}c_j$, $\hat{c}c_j$ and $\hat{d}c_j$, required by EP, are explained in detail, following the algorithm described in Section 1 of the Supplementary Material.

4. The EP Approximation to the $\Phi(\cdot)$ and $\Omega(\cdot, \cdot)$ factors

The EP algorithm updates each of the approximate factors presented in the previous section until convergence. The following sections will describe the necessary operations needed for the EP algorithm to update each of the factors. It the following subsection, it is assumed that we have already obtained the mean and variances of each of the $\mathcal{K}$ and $\mathcal{J}$ conditional predictive distributions, which will be explained in detail in section 4.3.

4.1. EP update operations for the $\Phi(\cdot)$ factors

As it was explained in section 3, for the $\mathcal{M}$ Pareto Set points defined by the set $\mathcal{X}^*$, in every point $\boldsymbol{x}_i \in \mathcal{X}^*$, the EP algorithm will generate J approximations for the $\Phi(\boldsymbol{x}_j)$ factors for every constraint c_j that will be defined by its mean $\hat{f}_j^{\boldsymbol{x}}$ and its variance $\hat{e}_j^{\boldsymbol{x}}$. Computations are done for all the points $\boldsymbol{x}_i \in \mathcal{X}^*$. The operations for these factors are described as follows.

4.1.1. Computation of the Cavity Distribution

The first step performed by the EP algorithm is the computation of the Cavity Distribution $\tilde{q}^{\setminus n}(\boldsymbol{x})$. In order to make the computations easier, we first obtain the natural parameters of the Gaussian Distributions for all the $\mathcal{M}$ Pareto Set points by using the equations:

$$\begin{aligned} \hat{\boldsymbol{m}}_j &= \frac{\boldsymbol{\xi}_j}{diag(\Xi_j)}\,, \\ \hat{\boldsymbol{v}}_j &= \frac{1}{diag(\Xi_j)}\,. \end{aligned} \tag{16}$$

Where Ξ_j is a vector of the variances of the $\mathcal{M}$ points for the constraint c_j and Ξ is the matrix of all the variances of all $\mathcal{M}$ and $\mathcal{N}$ points which construction will be explained in detail in section 4.3. The term *diag* holds for the diagonal of Ξ as we are only interested in the variance of the M points and not the variance of these points with the N points for the factor $\Phi(\cdot)$. In the same way, $\boldsymbol{\xi}_j$, is the vector of means for the constraint c_j and $\boldsymbol{\xi}$ contains all the means of all the points for every constraint in $\boldsymbol{c}$. $\hat{\boldsymbol{m}}_j$ and $\hat{\boldsymbol{v}}_j$ hold the mean and variance natural parameters corresponding for all the points in the set $\mathcal{X}^*$.

Once we have obtain the natural parameters $\hat{\boldsymbol{m}}_j$ and $\hat{\boldsymbol{v}}_j$, we obtain the cavity distribution. As we are dealing with natural parameters, it is not necessary to use the formula for the ratio of Gaussian Distributions, the cavity distribution defined by mean $\hat{\boldsymbol{m}}^{\backslash j}$ and variance $\hat{\boldsymbol{v}}^{\backslash j}$ will simply be obtained by the substraction of the natural parameters between the approximated distribution defined by parameters $\hat{\boldsymbol{m}}_{\boldsymbol{j}}$ and $\hat{\boldsymbol{v}}_{\boldsymbol{j}}$ (which is equivalent to the product of all the factors for all the constraints) and the factor $\hat{\boldsymbol{e}}_j$ and $\hat{\boldsymbol{f}}_j$ corresponding to the constraint c_j that we want to update:

$$\begin{aligned}\hat{\boldsymbol{v}}^{\backslash j}_{nat} &= \hat{\boldsymbol{v}}_{\boldsymbol{j}} - \hat{\boldsymbol{e}}_j\,,\\ \hat{\boldsymbol{m}}^{\backslash j}_{nat} &= \hat{\boldsymbol{m}}_{\boldsymbol{j}} - \hat{\boldsymbol{f}}_j\,. \end{aligned}\tag{17}$$

Once the substraction is done, we transform the natural parameters of the cavity distribution into Gaussian parameters again by using the formula that converts natural to Gaussian parameters.

$$\begin{aligned}\hat{\boldsymbol{v}}^{\backslash j} &= \frac{1}{\hat{\boldsymbol{v}}^{\backslash j}_{nat}}\,,\\ \hat{\boldsymbol{m}}^{\backslash j} &= \hat{\boldsymbol{m}}_{nat}\hat{\boldsymbol{v}}^{\backslash j}\,. \end{aligned}\tag{18}$$

The variances $\hat{\boldsymbol{v}}^{\backslash j}$ need to be positive for the following operations.

4.1.2. Computation of the Partial Derivatives of the Normalization Constant

Once the cavities $\hat{\boldsymbol{v}}^{\backslash j}$ and $\hat{\boldsymbol{m}}^{\backslash j}$ have been computed, the EP need to compute the quantities required for the update of the factors $\hat{\boldsymbol{e}}_j$ and $\hat{\boldsymbol{f}}_j$ in order to minimize the KL divergence between $\Phi(\cdot)$ and the approximation distribution. These quantities are the first and second moments of the distribution that we want to approximate. These are given by the log of the partial derivatives of Z_j, the constant that normalizes the distribution that we want to approximate, in this case, $\hat{\Phi}(\cdot)$.

$$Z_j = \int \hat{\Phi}(\boldsymbol{x}^*)\ dc_j\,. \tag{19}$$

As $\Phi(\boldsymbol{x}^*)$ is approximated by a Gaussian Distribution $\hat{\Phi}(\boldsymbol{x}^*)$ with mean $\hat{\boldsymbol{m}}^{\setminus j}$ and variance $\hat{\boldsymbol{v}}^{\setminus j}$, the normalization constant Z_j can be computed in closed form and its given by the cumulative distribution function ,$\Phi(\cdot)$, of this Gaussian Distribution:

$$Z_j = \Phi\left(\frac{\hat{\boldsymbol{m}}^{\setminus j}}{\sqrt{\hat{\boldsymbol{v}}^{\setminus j}}}\right). \tag{20}$$

Let $\alpha = \frac{\hat{\boldsymbol{m}}^{\setminus j}}{\sqrt{\hat{\boldsymbol{v}}^{\setminus j}}}$, then $\log(Z_j) = \log(\Phi(\alpha))$. For numerical robustness, if $a, b \in \mathbb{R}$, we apply the rule $\frac{a}{b} = \exp\left(\log(a) - \log(b)\right)$. Using these expressions, the log-derivatives are computed as follows:

$$\begin{aligned}\frac{\partial \log(Z_j)}{\partial \hat{\boldsymbol{m}}^{\setminus j}} &= \frac{\exp\{\log(N(\alpha)) - \log(Z_j)\}}{\sqrt{\hat{\boldsymbol{v}}^{\setminus j}}}\,,\\ \frac{\partial \log(Z_j)}{\partial \hat{\boldsymbol{v}}^{\setminus j}} &= -\frac{\exp\{\log(N(\alpha)) - \log(Z_j)\}\alpha}{2\hat{\boldsymbol{v}}^{\setminus j}}\,.\end{aligned} \tag{21}$$

Where $N(\cdot)$ represent the Gaussian probability density function. These expressions are valid for computing the first and second moments, but they do not present numerical robustness in all experiments. Since the lack of robustness of $\frac{\partial \log(Z_j)}{\partial \hat{\boldsymbol{v}}^{\setminus j}}$, we use the formula given by the Appendix A of the work by Opper [4], and use the second partial derivative $\frac{\partial^2 \log(Z_j)}{\partial [\hat{\boldsymbol{m}}^{\setminus j}]^2}$ rather than $\frac{\partial \log(Z_j)}{\partial \hat{\boldsymbol{v}}^{\setminus j}}$. This derivative is given by the following expression:

$$\frac{\partial^2 \log(Z_j)}{\partial [\hat{\boldsymbol{m}}^{\setminus j}]^2} = -\exp\{\log(N(\alpha)) - \log(Z_j)\}\frac{\alpha \exp\{\log(N(\alpha)) - \log(Z_j)\}}{\hat{\boldsymbol{v}}^{\setminus j}}\,. \tag{22}$$

Given these derivatives, in the next section it will be explained how to obtain the individual approximate factors $\hat{\boldsymbol{e}}_j$ and $\hat{\boldsymbol{f}}_j$.

4.1.3. Computation of the First and Second Moments for the Updates

We now have to compute the first and second moments for the mentioned updates. As the distributions are going to be Gaussian, which belongs to the exponential family, we know that the first and second moment of the Gaussian Distribution are given by:

$$\mathbb{E}[\boldsymbol{x}] = \boldsymbol{\mu}\,, \tag{23}$$

$$\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T\,, \tag{24}$$

In order to match the moments, we make the Kullback-Leibler divergence between both distributions zero. With the previous definition of Z_j and given the computed robust derivatives, the expressions that obtain the first and second moments that give the approximate factors $\hat{\boldsymbol{e}}_j$ that represents the variance and $\hat{\boldsymbol{f}}_j$ that represent the mean for the constraint $\boldsymbol{c}_j(\boldsymbol{X}^*)$ are given by the following expressions according to EP:

$$\hat{\boldsymbol{f}}_j = \frac{\frac{\partial \log(Z_j)}{\partial \hat{\boldsymbol{m}}^{\setminus j}} - \hat{\boldsymbol{m}}^{\setminus j} \frac{\partial^2 \log(Z_j)}{\partial [\hat{\boldsymbol{m}}^{\setminus j}]^2}}{1 + \frac{\partial^2 \log(Z_j)}{\partial [\hat{\boldsymbol{m}}^{\setminus j}]^2} \hat{\boldsymbol{v}}^{\setminus j}} \,,$$
$$\hat{\boldsymbol{e}}_j = -\frac{\frac{\partial^2 \log(Z_j)}{\partial [\hat{\boldsymbol{m}}^{\setminus j}]^2}}{1 + \frac{\partial^2 \log(Z_j)}{\partial [\hat{\boldsymbol{m}}^{\setminus j}]^2} \hat{\boldsymbol{v}}^{\setminus j}} \,. \tag{25}$$

In practice, the updates are not absolute, they are dumped as the section 5.2 of this supplementary material shows.

4.2. EP update operations for the $\Omega(\cdot,\cdot)$ factors

Recalling section 3, for the $\mathcal{M}$ Pareto Set points defined by the set $\mathcal{X}^*$ and the $\mathcal{N}$ input space observation points defined by the set $\hat{\mathcal{X}}$, for every pair of points $\boldsymbol{x}_i \in \hat{\mathcal{X}}$ and $\boldsymbol{x}_j \in \mathcal{X}^*$, the EP will generate K two-dimensional gaussian approximations for every objective $\boldsymbol{f}_k$ that will be defined for the pair observation and pareto set point by factors defined by mean $\hat{\boldsymbol{b}}_{ij}$ and variance $\hat{\boldsymbol{A}}_{ij}$ and for the pair of pareto set points by factors defined by mean $\hat{\boldsymbol{d}}_{ij}$ and variance $\hat{\boldsymbol{C}}_{ij}$. It will also define J one-dimensional gaussian approximations for every constraint c_j that will be defined for the pair observation and pareto set point by factors defined by mean $\hat{bc}_j$ and variance $\hat{ac}_j$ and for the pair of pareto set points by factors defined by mean $\hat{dc}_j$ and variance $\hat{cc}_j$. Computations are done for all the pairs of points from the sets $\mathcal{X}^*$ and $\hat{\mathcal{X}}$. The necessary operations for computing these factors are described in the following sections.

4.2.1. Computation of the Cavity Distribution

For the factors $\hat{ac}_j$, $\hat{bc}_j$, $\hat{cc}_j$ and $\hat{dc}_j$ that approximate the $\mathcal{J}$ one-dimensional gaussian approximations for every constraint c_j, the operations needed to extract the cavity distribution from the approximate distribution are the same ones as the ones described in Section

4.1.1. These operations are done for the observation points in $\hat{\mathcal{X}}$ for the factors $\hat{ac}_j$, $\hat{bc}_j$ and for the Pareto Set points in $\mathcal{X}^*$ for the factors $\hat{cc}_j$ and $\hat{dc}_j$. That is, obtaining the natural parameters of Ξ_j as in Eq. (16), substracting the natural parameters of the factor that is approximated, Eq. (17), and obtaining the gaussian parameters of the cavity that we define for a point $\boldsymbol{x}_i$, $m_{ij}^{\backslash b_j}$ and $v_{ij}^{\backslash a_j}$, as shown in Eq. (18).

Obtaining the cavity distribution for the factors $\hat{\boldsymbol{A}}_{ij}$, $\hat{\boldsymbol{b}}_{ij}$, $\hat{\boldsymbol{C}}_{ij}$ and $\hat{\boldsymbol{d}}_{ij}$ that approximate the $\mathcal{K}$ two-dimensional gaussian approximations for every objective $\boldsymbol{f}_k$ follow different expressions as in this case the Gaussian Distributions are bivariate for every pair of points considered.

In the first case, for the case of approximating a distribution that consider a point $\boldsymbol{x}_i$ belonging to the observations set $\hat{\mathcal{X}}$ and a point $\boldsymbol{x}_j$ from the pareto set $\mathcal{X}^*$, that is, the factors $\hat{\boldsymbol{A}}_{ij}$ and $\hat{\boldsymbol{b}}_{ij}$, it is necessary to obtain, for every objective k and each of the pair of points mentioned, the natural parameters $\boldsymbol{m}_{ij(nat)}^{k}$ and ${\boldsymbol{V}_{ij}^{k}}^{-1}$ of the Gaussian Process that models each of the $\mathcal{K}$ objectives $f(\cdot)_j$. These natural parameters are obtained by the following expressions:

$$
\begin{aligned}
\boldsymbol{m}_{ij(nat)}^{k} &= {\boldsymbol{V}_{ij}^{k}}^{-1} \boldsymbol{m}_{ij}^{k}\,, \\
{\boldsymbol{V}_{ij}^{k}}^{-1} &= (\boldsymbol{V}_{ij}^{k})^{-1}\,,
\end{aligned}
\tag{26}
$$

where $\boldsymbol{V}_{ij}^{k}$ is a 2x2 matrix that represent in the points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ the variance of the gaussian approximation of the objective k and $\boldsymbol{m}_{ij}^{k}$ is a vector that represent in the points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ the mean of the gaussian approximation of the objective k.

As in the constraints case, we now extract the cavity distribution that we define by the natural parameters $\boldsymbol{m}_{ijk(nat)}^{\backslash b}$ and $\boldsymbol{V}_{ijk(nat)}^{\backslash A}$, by substracting to the computed natural parameters $\boldsymbol{m}_{ij(nat)}^{k}$ and ${\boldsymbol{V}_{ij}^{k}}^{-1}$, computed in the previous step, the factors that we want to update $\boldsymbol{b}_{ij}^{k}$ and $\boldsymbol{A}_{ij}^{k}$. That is:

$$
\begin{aligned}
\boldsymbol{m}_{ijk(nat)}^{\backslash b} &= \boldsymbol{m}_{ij(nat)}^{k} - \boldsymbol{b}_{ij}^{k}\,, \\
\boldsymbol{V}_{ijk(nat)}^{\backslash A} &= {\boldsymbol{V}_{ij}^{k}}^{-1} - \boldsymbol{A}_{ij}^{k}\,.
\end{aligned}
\tag{27}
$$

For the bivariate gaussian distribution, the step of obtaining the gaussian parameters from the natural parameters is defined by the following expressions:

$$\begin{aligned} \boldsymbol{m}_{ijk}^{\backslash b} &= \boldsymbol{V}_{ijk}^{\backslash A}\, \boldsymbol{m}_{ijk(nat)}\,, \\ \boldsymbol{V}_{ijk}^{\backslash A} &= (\boldsymbol{V}_{ijk(nat)}^{\backslash A})^{-1}\,, \end{aligned} \tag{28}$$

where $\boldsymbol{V}_{ijk}^{\backslash A}$ is a 2x2 matrix with the variances of each of the points and the correlation between each of them and $\boldsymbol{m}_{ijk}^{\backslash b}$ is a two position vector that represent the means. In the case of the factors $\hat{\boldsymbol{C}}_{ij}$ and $\hat{\boldsymbol{d}}_{ij}$ that consider two Pareto Set points, the operations for extracting the cavity distribution are the same ones as in the previous case.

4.2.2. Computation of the Partial Derivatives of the Normalization Constant

In this section, the operations needed to compute the partial derivatives for all the $\hat{\boldsymbol{A}}_{ij}$, $\hat{\boldsymbol{b}}_{ij}$, $\hat{\boldsymbol{C}}_{ij}$, $\hat{\boldsymbol{d}}_{ij}$, $\hat{ac}_j$, $\hat{bc}_j$, $\hat{cc}_j$ and $\hat{dc}_j$ are described. These derivatives need previous computations in order to compute the normalization constant Z_Ω of the factor $\Omega(\cdot,\cdot)$ that we want to approximate. These computations are given by the following expressions, all of which depend upon terms computed in the previous section. The shown computations are the result of applying rules in order to be robust such as $a/b = \exp\{\log(a) - \log(b)\}$ and $ab = \exp\{\log(a) + \log(b)\}$.These operations are equivalent for the two points cases, but here, the necessary operations for computing the normalization constant Z_Ω are described for the case of the factors $\hat{\boldsymbol{A}}_{ij}$, $\hat{\boldsymbol{b}}_{ij}$, $\hat{ac}_j$ and $\hat{bc}_j$:

$$\boldsymbol{s}_k = \boldsymbol{V}_{ijk[0,0]}^{\backslash A} + \boldsymbol{V}_{ijk[1,1]}^{\backslash A} - 2\boldsymbol{V}_{ijk[0,1]}^{\backslash A}\,, \tag{29}$$

$$\boldsymbol{\alpha}_k = \frac{\boldsymbol{m}_{ijk[0]}^{\backslash b} - \boldsymbol{m}_{ijk[1]}^{\backslash b}}{\sqrt{s_k}}\,, \tag{30}$$

$$\boldsymbol{\beta}_j = \frac{m_{ij}^{\backslash b_j}}{\sqrt{v_{ij}^{\backslash a_j}}}\,, \tag{31}$$

$$\boldsymbol{\phi} = \Phi(\boldsymbol{\alpha})\,, \tag{32}$$

$$\tag{33}$$

where $\Phi(\cdot)$ represents the c.d.f of a Gaussian distribution,

$$\boldsymbol{\gamma} = \Phi(\boldsymbol{\beta})\,, \tag{34}$$

$$\zeta = 1 - \exp\{\sum_{k=1}^{K} \log(\boldsymbol{\phi}_k)\}\,, \tag{35}$$

$$\log(\eta) = \sum_{j=1}^{J} \log(\boldsymbol{\gamma}_j) + \log(\zeta)\,, \tag{36}$$

$$\lambda = 1 - \exp\{\sum_{j=1}^{J} \log(\boldsymbol{\gamma}_j)\}\,, \tag{37}$$

$$\tau = \max(\log(\eta), \log(\lambda))\,, \tag{38}$$

$$\log(Z_\Omega) = \log(\exp\{\log(\eta) - \tau\} + \exp\{\log(\lambda) - \tau\}) + \tau\,. \tag{39}$$

Having computed these terms, the log partial derivatives for the update of the factors that collaborate to the approximation of the objective variances $\hat{\boldsymbol{A}}_{ij}$ and the objective means $\hat{\boldsymbol{b}}_{ij}$ are given by the expressions:

$$\boldsymbol{\rho}_k = -\exp\{\log(\mathcal{N}(\boldsymbol{\alpha}_k))\} - \log(Z_\Omega) + \sum_{k=1}^{K}\{\log(\Phi(\boldsymbol{\alpha}_k))\} - \log(\Phi(\boldsymbol{\alpha}_k)) + \sum_{j=1}^{J}\{\log(\Phi(\boldsymbol{\beta}_j))\}\,, \tag{40}$$

$$\begin{aligned} \frac{\partial \log(Z_\Omega)}{\partial \boldsymbol{m}_{ijk}^{\backslash b}} &= \frac{\boldsymbol{\rho}_k}{\sqrt{\boldsymbol{s}_k}}\left[1, -1\right]\,, \\ \frac{\partial \log(Z_\Omega)}{\partial \boldsymbol{V}_{ijk}^{\backslash A}} &= -\frac{\boldsymbol{\rho}_k \boldsymbol{\alpha}_k}{2\boldsymbol{s}_k}[[1, -1], [-1,\ 1]]\,. \end{aligned} \tag{41}$$

Derivatives are computed for the two position vector mean and the 2x2 variance matrix, so they have the same structure, given by the [1, -1] and [[1, -1],[-1, 1]] expressions. The change in the sign appears due to the fact that the expression changes, whether it is the derivative of the mean of the observation point or the Pareto Set point or the derivative of the variance of one point or their correlation.

Alas, the derivative of the variance presents the same lack of robustness as in the constraint case shown in section 4.1.2. In order to ensure numerical robustness, we use the second partial derivative of the mean of the normalization constant instead of the first partial derivative of the variance for the further computation of the second moment. That is,

$$\frac{\partial^2 \log(Z_\Omega)}{\partial\big[\boldsymbol{m}_{ijk}^{\setminus b}\big]^2} = -\frac{\boldsymbol{\rho}_k}{\boldsymbol{s}_k}(\boldsymbol{\alpha}_k + \boldsymbol{\rho}_k)[[1, -1], [-1,\ 1]]\,. \tag{42}$$

For the log partial derivatives for the update of the factors that collaborate to the approximation of the constraint variances $\hat{ac}_j$ and the constraint means $\hat{bc}_j$, let $\boldsymbol{\omega}_j$ be defined as:

$$\begin{aligned}\boldsymbol{\omega}_j =& \exp\{\log(\mathcal{N}(\boldsymbol{\beta}_j))\} - \log(Z_\Omega) + \log(\zeta) + \sum_{j=1}^{J}(\log(\Phi(\boldsymbol{\beta}_j))) - \log(\Phi(\boldsymbol{\beta}_j)) - \exp\{\log(\mathcal{N}(\boldsymbol{\beta}_j))\}\,,\\ & - \log(Z_\Omega) + \sum_{j=1}^{J}(\log(\Phi(\boldsymbol{\beta}_j))) - \log(\Phi(\boldsymbol{\beta}_j))\,. \end{aligned} \tag{43}$$

Then, the robust log partial derivatives for the first and the second moments are given by the expressions:

$$\begin{aligned}\frac{\partial \log(Z_\Omega)}{\partial m_{ij}^{\setminus \boldsymbol{b}_j}} &= \frac{\boldsymbol{\omega}_j}{\sqrt{\boldsymbol{s}_j}}\,,\\ \frac{\partial^2 \log(Z_\Omega)}{\partial [m_{ij}^{\setminus \boldsymbol{b}_j}]^2} &= -\frac{\boldsymbol{\omega}_j}{\boldsymbol{s}_j}\ (\boldsymbol{\beta}_j + \boldsymbol{\omega}_j)\,. \end{aligned} \tag{44}$$

The expressions for the log partial derivatives of $\hat{\boldsymbol{C}}_{ij}$, $\hat{\boldsymbol{d}}_{ij}$, $\hat{cc}_j$ and $\hat{dc}_j$ are similar to the presented expressions in this section, but taking into account pairs of points belonging to the set $\mathcal{X}^*$.

4.2.3. Computation of the First and Second Moments for the Updates

Giving the expressions computed in the previous section, the first and second moments of the different Gaussian Distributions that approximate the factor $\Omega(\cdot,\cdot)$ can now be computed.

The expressions for computing the factors $\hat{\boldsymbol{A}}_{ij}$, $\hat{\boldsymbol{b}}_{ij}$, $\hat{\boldsymbol{C}}_{ij}$, $\hat{\boldsymbol{d}}_{ij}$ for each of the K objectives and the factors $\hat{a}c_j$, $\hat{b}c_j$, $\hat{c}c_j$ and $\hat{d}c_j$ for each of the J constraints are the following ones:

$$\hat{\boldsymbol{A}}_{ij}^{k} = \frac{\partial^2 \log(Z_\Omega)}{\partial \big[\boldsymbol{m}_{ijk}^{\backslash b}\big]^2} \; ((\boldsymbol{V}_{ijk}^{\backslash A} \frac{\partial^2 \log(Z_\Omega)}{\partial \big[\boldsymbol{m}_{ijk}^{\backslash b}\big]^2})^{-1} [[1,0],[0,1]]) \,, \tag{45}$$

$$\hat{\boldsymbol{b}}_{ij}^{k} = ((\frac{\partial \log(Z_\Omega)}{\partial \boldsymbol{m}_{ijk}^{\backslash b}} - \boldsymbol{m}_{ijk}^{\backslash b}) \frac{\partial^2 \log(Z_\Omega)}{\partial \big[\boldsymbol{m}_{ijk}^{\backslash b}\big]^2}) \; ((\boldsymbol{V}_{ijk}^{\backslash A} \frac{\partial^2 \log(Z_\Omega)}{\partial \big[\boldsymbol{m}_{ijk}^{\backslash b}\big]^2})^{-1} + [[1,0],[0,1]]) \,, \tag{46}$$

$$\hat{\boldsymbol{C}}_{ij}^{k} = \frac{\partial^2 \log(Z_\Omega)}{\partial \big[\boldsymbol{m}_{ijk}^{\backslash b}\big]^2} \; ((\boldsymbol{V}_{ijk}^{\backslash A} \frac{\partial^2 \log(Z_\Omega)}{\partial \big[\boldsymbol{m}_{ijk}^{\backslash b}\big]^2})^{-1} [[1,0],[0,1]]) \,, \tag{47}$$

$$\hat{\boldsymbol{d}}_{ij}^{k} = ((\frac{\partial \log(Z_\Omega)}{\partial \boldsymbol{m}_{ijk}^{\backslash b}} - \boldsymbol{m}_{ijk}^{\backslash b}) \frac{\partial^2 \log(Z_\Omega)}{\partial \big[\boldsymbol{m}_{ijk}^{\backslash b}\big]^2}) \; ((\boldsymbol{V}_{ijk}^{\backslash A} \frac{\partial^2 \log(Z_\Omega)}{\partial \big[\boldsymbol{m}_{ijk}^{\backslash b}\big]^2})^{-1} + [[1,0],[0,1]]) \,, \tag{48}$$

for the the rest of the factors, suppose that the index h refers to the points of the Pareto Set $\mathcal{X}^*$:

$$\hat{a}c_h^j = -\frac{\frac{\partial^2 \log(Z_\Omega)}{\partial [m_{ic}^{\backslash \boldsymbol{b}_j}]^2}}{1 + \frac{\partial^2 \log(Z_\Omega)}{\partial [m_{ic}^{\backslash \boldsymbol{b}_j}]^2} v_{ic}^{\backslash a_j}} \,, \tag{49}$$

$$\hat{b}c_h^j = \frac{\frac{\partial \log(Z_\Omega)}{\partial m_{ic}^{\backslash \boldsymbol{b}_j}} - m_{ic}^{\backslash b_j} \frac{\partial^2 \log(Z_\Omega)}{\partial [m_{ic}^{\backslash \boldsymbol{b}_j}]^2}}{1 + \frac{\partial^2 \log(Z_\Omega)}{\partial [m_{ic}^{\backslash \boldsymbol{b}_j}]^2} v_{ic}^{\backslash a_j}} \,, \tag{50}$$

$$\hat{c}c_h^j = -\frac{\frac{\partial^2 \log(Z_\Omega)}{\partial [m_{ic}^{\backslash \boldsymbol{b}_j}]^2}}{1 + \frac{\partial^2 \log(Z_\Omega)}{\partial [m_{ic}^{\backslash \boldsymbol{b}_j}]^2} v_{ic}^{\backslash a_j}} \,, \tag{51}$$

$$\hat{d}c_h^j = \frac{\frac{\partial \log(Z_\Omega)}{\partial m_{ic}^{\backslash \boldsymbol{b}_j}} - m_{ic}^{\backslash b_j} \frac{\partial^2 \log(Z_\Omega)}{\partial [m_{ic}^{\backslash \boldsymbol{b}_j}]^2}}{1 + \frac{\partial^2 \log(Z_\Omega)}{\partial [m_{ic}^{\backslash \boldsymbol{b}_j}]^2} v_{ic}^{\backslash a_j}} \,. \tag{52}$$

All these factors are then used to rebuild the means and the variances of the Gaussian Processes that model the K objectives and C constraints of a constrained multi-objective optimization problem, as will be shown in the following section. That is, C one-dimensional Gaussian Distributions for the constraint models and C one-dimensional Gaussian Distributions and K two-dimensional Gaussian Distributions for the objective models in each of the points in $\mathcal{X} = \{\mathcal{X}^* \cup \hat{\mathcal{X}} \cup \boldsymbol{x}\}$.

4.3. Reconstruction of the Conditional Predictive Distribution

In this section, we illustrate the way of obtaining a Conditional Predictive Distribution for every objective f_k and every constraint c_j, given a sampled Pareto Set $\mathcal{X}^* = \{\boldsymbol{x}_1^*, ..., \boldsymbol{x}_M^*\}$

of size M and a set of N input locations $\hat{\mathcal{X}} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ with corresponding observations of the k-th objective $\boldsymbol{y}_k$ and of the j-th constraint $\boldsymbol{y}_j$. For the following, it is assumed that we are given the EP approximate factors $\Phi(\cdot)$ and $\Omega(\cdot, \cdot)$, as an input for the next operations, which computation is explained in the previous section.

Recalling Eqs. 7, 8 and 9 of section 2, we want to obtain the J Conditional Predictive Distributions in the products of constraints and the K Conditional Predictive Distributions of the Gaussian Processes that model the objectives. The products presented in these factors are not a problem, due to the fact that the Gaussian Distributions are closed under the product operation, that is, the product of Gaussian Distributions is another Gaussian Distribution. These Conditional Predictive Distributions of the objectives and constraints are then used in Eq.(11) to build the final approximation.

Following the notation of section 4.1.1, let $\boldsymbol{\xi}_j$ and Ξ_j be the mean vector and variance matrix of the one-dimensional Gaussian Distributions of the $M + N$ points that generate the Gaussian Processes that model the constraints and let $\boldsymbol{m}_k$ and $\boldsymbol{V}_k$ be the mean vector and variance matrix of the two-dimensional Gaussian Distributions of the $M + N$ points that generate the Gaussian Processes that model the objectives. In order to update the constraint and objective distribution marginals, it is necessary to first follow the operations given by the equations 14 and 22, to obtain the natural parameters from the means and variances. Intuitively, as they are all natural parameters, these will be just sums taking into account that the matrices are formed first by the Pareto Set Points ,M, and then by the observations N. Univariate factors are added to the diagonal of these matrices, as they are not correlated with other points. Once the natural parameters are computed, the new means $\boldsymbol{\xi}_j$, $\boldsymbol{m}_k$ and variances Ξ_j, $\boldsymbol{V}_k$ marginals are updated from the EP factors $\hat{\boldsymbol{A}}_{ij}$, $\hat{\boldsymbol{b}}_{ij}$,

$\hat{\boldsymbol{C}}_{ij}$, $\hat{\boldsymbol{d}}_{ij}$, $\hat{e}_j$, $\hat{f}_j$, $\hat{ac}_j$, $\hat{bc}_j$, $\hat{cc}_j$ and $\hat{dc}_j$ by the following expressions:

$$
\begin{aligned}
&\boldsymbol{\Xi}^j_{ii} = \boldsymbol{\Xi}^j_{ii(old)} + \sum_{m=1}^{M} \hat{cc}^j_{mi} + \hat{e}^j_i && \text{for } \; i = 1, ..., M\,, \\
&\boldsymbol{\Xi}^c_{ii} = \boldsymbol{\Xi}^j_{ii(old)} + \sum_{m=1}^{M} \hat{ac}^j_{mi} && \text{for } \; i = M+1, ..., N+M\,, \\
&\boldsymbol{\xi}^c_i = \; \boldsymbol{\xi}^j_{i(old)} + \sum_{m=1}^{M} \hat{dc}^j_{mi} + \hat{f}^j_i && \text{for } \; i = 1, ..., M\,, \\
&\boldsymbol{\xi}^c_i = \; \boldsymbol{\xi}^j_{i(old)} + \sum_{m=1}^{M} \hat{bc}^j_{mi} && \text{for } \; i = M+1, ..., N+M\,, \\
&\boldsymbol{V}^k_{ii} = \boldsymbol{V}^k_{ii(old)} + \sum_{j=M+1}^{N} \hat{\boldsymbol{A}}^k_{ji[1,1]} + \sum_{j=1}^{M} \hat{\boldsymbol{C}}^k_{ij[0,0]} + \sum_{j=1}^{M} \hat{\boldsymbol{C}}^k_{ji[1,1]} && \text{for } \; i = 1, ..., M\,, \\
&\boldsymbol{V}^k_{ii} = \boldsymbol{V}^k_{ii(old)} + \sum_{j=1}^{M} \hat{\boldsymbol{A}}^k_{ij[0,0]} && \text{for } \; i = M+1, ..., N+M\,, \\
&\boldsymbol{V}^k_{ij} = \boldsymbol{V}^k_{ij(old)} + \boldsymbol{C}^k_{ij[0,1]} + {\boldsymbol{C}^k_{ij[1,0]}}^T && \text{for } \; i = 1, ..., M, \text{ and for } \; j = 1, ..., M\,, \\
&\boldsymbol{V}^k_{ij} = \boldsymbol{V}^k_{ij(old)} + \boldsymbol{A}^k_{ij[0,1]} && \text{for } \; i = M+1, ..., N, \text{ and for } \; j = 1, ..., M\,, \\
&\boldsymbol{V}^k_{ij} = \boldsymbol{V}^k_{ij(old)} + {\boldsymbol{A}^k_{ij[0,1]}}^T && \text{for } \; i = 1, ..., M, \text{ and for } \; j = M+1, ..., N\,, \\
&\boldsymbol{m}^k_i = \boldsymbol{m}^k_{i(old)} + \sum_{j=M+1}^{N+M} \hat{\boldsymbol{b}}^k_{ji[1]} + \sum_{j=1}^{M} \hat{\boldsymbol{d}}^k_{ij[0]} + \sum_{j=1}^{M} \hat{\boldsymbol{d}}^k_{ji[1]} && \text{for } \; i = 1, ..., M\,, \\
&\boldsymbol{m}^k_i = \boldsymbol{m}^k_{i(old)} + \sum_{j=1}^{M} \hat{\boldsymbol{b}}^k_{ij[0]} && \text{for } \; i = M+1, ..., N+M\,.
\end{aligned}
\tag{53}
$$

These natural parameters are then converted into Gaussian ones using the equations and 16 and 24. Once these operations are done the Gaussian Processes that model the objectives and constraints are updated from a full EP iteration.

4.4. The Conditional Predictive Distribution at a New Point

In this section, the computation of the conditional distributions for every model of the objectives f_k and the constraints c_j at a new candidate location $\boldsymbol{x}_{N+1}$ is explained. This requires that q($\boldsymbol{f}$,$\boldsymbol{c}$) is already computed approximated by the factors of EP. The interest lies in evaluating the conditional predictive variances for $f_k(\boldsymbol{x}_{N+1})$ and for $c_j(\boldsymbol{x}_{N+1})$ for every objective and constraint. With this variances, it is possible to compute the PESMOC acquisition function given by the Negative Differential Entropy w.r.t the Conditional Predictive Distribution and the Predictive Distribution:

$$\alpha(\mathbf{x}) \approx \sum_{j=1}^{J} \log v_j^{\mathrm{PD}}(\mathbf{x}) + \sum_{k=1}^{K} \log v_k^{\mathrm{PD}}(\mathbf{x}) - \frac{1}{M} \sum_{m=1}^{M} \Big[\sum_{j=1}^{J} \log v_j^{\mathrm{CPD}}(\mathbf{x}|\mathcal{X}^\star_{(m)}) + \sum_{k=1}^{K} \log v_k^{\mathrm{CPD}}(\mathbf{x}|\mathcal{X}^\star_{(m)}) \Big] . \tag{54}$$

In order to obtain these variances, we need to execute the Expectation Propagation algorithm once again, but as it is only a new observation point, it is not necessary to compute the $\Phi(\cdot)$ factor. The Conditional Predictive Distribution variances expressions that we need to evaluate in order to obtain these variances are:

$$p(f_k(\boldsymbol{x}_{N+1}))|\hat{\mathcal{X}}, \mathcal{X}^*, \boldsymbol{x}_{N+1}) \approx \int Z_k^{-1} q(\boldsymbol{f}_k, \boldsymbol{f}_k(\boldsymbol{x}_{N+1})) \prod_{\mathbf{x}^\star \in \mathcal{X}^\star} \Big(\Omega(\mathbf{x}_{N+1}, \mathbf{x}^\star) \Big) d\boldsymbol{f}_k, d\boldsymbol{c}_j \, , \tag{55}$$

$$p(c_j(\boldsymbol{x}_{N+1}))|\hat{\mathcal{X}}, \mathcal{X}^*, \boldsymbol{x}_{N+1}) \approx \int Z_j^{-1} q(\boldsymbol{c}_j, \boldsymbol{c}_j(\boldsymbol{x}_{N+1})) \prod_{\mathbf{x}^\star \in \mathcal{X}^\star} \Big(\Omega(\mathbf{x}_{N+1}, \mathbf{x}^\star) \Big) d\boldsymbol{f}_k, d\boldsymbol{c}_j \, , \tag{56}$$

where Z is a normalization constant and $q(\boldsymbol{f}, \boldsymbol{f}(\boldsymbol{x}_{N+1}))$ and $q(\boldsymbol{c}, \boldsymbol{c}(\boldsymbol{x}_{N+1}))$ are multivariate gaussian distributions that results by extending $q(\boldsymbol{f})$ and $q(\boldsymbol{c})$ with the new point $\boldsymbol{x}_{N+1}$. We have only taken into account the approximate factors that depend on f_k and c_j. The covariances between $\boldsymbol{f}_k$ and $f_k(\boldsymbol{x}_{N+1})$ and the covariances between $\boldsymbol{c}_j$ and $c_j(\boldsymbol{x}_{N+1})$ are obtained from the GP posteriors for f_k and c_j given the observed data. In the same way, the mean and the variance of $f_k(\boldsymbol{x}_{N+1})$ and $c_j(\boldsymbol{x}_{N+1})$ can be obtained. As all the factors are Gaussian and the product operation is closed the results are also Gaussian Distributions.

Let $\tilde{\boldsymbol{f}}_k = (f_k(\boldsymbol{x}_1^*), ..., f_k(\boldsymbol{x}_M^*), f_k(\boldsymbol{x}_{N+1}^*))^T$ and $\tilde{\boldsymbol{c}}_j = (c_j(\boldsymbol{x}_1^*), ..., c_j(\boldsymbol{x}_M^*), c_j(\boldsymbol{x}_{N+1}^*))^T$. As $\prod_{\mathbf{x}^\star \in \mathcal{X}^\star}$
$(\Omega(\mathbf{x}_{N+1}, \mathbf{x}^\star))$ does not depend on $f_k(\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$ nor in $c_j(\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$, we can marginalize these variables in Eq. (55) to obtain Gaussian Distributions for every objective k and every

constraint j in the point $\boldsymbol{x}_{N+1}$, proportional to:

$$\int q(\tilde{\boldsymbol{f}}_k) \prod_{\mathbf{x}^\star \in \mathcal{X}^\star} \left(\Omega(\mathbf{x}_{N+1}, \mathbf{x}^\star) \right) \prod_{i=1}^{M} df_k(\boldsymbol{x}_i^*) \,, \tag{57}$$

$$\int q(\tilde{\boldsymbol{c}}_j) \prod_{\mathbf{x}^\star \in \mathcal{X}^\star} \left(\Omega(\mathbf{x}_{N+1}, \mathbf{x}^\star) \right) \prod_{i=1}^{J} dc_j(\boldsymbol{x}_i^*) \,, \tag{58}$$

these expressions generate Gaussian Distributions, one for every objective and one for every constraint, at the point $\boldsymbol{x}_{N+1}$:

$$\int \mathcal{N}(\tilde{\boldsymbol{f}}_k | (\boldsymbol{V}^x)^{-1} \boldsymbol{m}^x, (\boldsymbol{V}^x)^{-1}) \prod_{i=1}^{M} df_k(\boldsymbol{x}_i^*) = \mathcal{N}(f_k(\boldsymbol{x}_{N+1}) | m_k, v_k) \,, \tag{59}$$

$$\int \mathcal{N}(\tilde{\boldsymbol{c}}_j | (\boldsymbol{\Xi}^x)^{-1} \boldsymbol{\xi}^x, (\boldsymbol{\Xi}^x)^{-1}) \prod_{j=1}^{J} dc_j(\boldsymbol{x}_i^*) = \mathcal{N}(c_j(\boldsymbol{x}_{N+1}) | \xi_j, v_j) \,, \tag{60}$$

where $\boldsymbol{m}^x$, $\boldsymbol{V}^x$, $\boldsymbol{\xi}^x$ and $\boldsymbol{\Xi}^x$ are the natural parameters of the Conditional Predictive Distributions for the objective k and the constraint j, which are Gaussian. The variances v_k and v_j of the Gaussian Approximations of the objectives and the constraints are the ones needed to the entropy computation in Eq.(54). These are given by the last diagonal of the natural parameter variance matrices $\boldsymbol{V}^x$ and $\boldsymbol{\Xi}^x$. As the means are not needed, no further details are here given in order to compute them.

Each entry in $\boldsymbol{\Xi}^x$ and $\boldsymbol{V}^x$, as they are natural parameters, is given by the reconstruction of the predictive distribution, which is the most expensive part:

$$\begin{aligned}
&\Xi^x_{i,j} = \Xi^c_{i,j} \quad \text{for} \quad 1 \leq i \leq M \quad \text{and} \quad 1 \leq j \leq M \,, \quad \text{and} \quad i \neq j \,, \\
&V^x_{i,j} = V^k_{i,j} \quad \text{for} \quad 1 \leq i \leq M \quad \text{and} \quad 1 \leq j \leq M \,, \quad \text{and} \quad i \neq j \,, \\
&V^x_{i,j} = \text{cov}(f_k(\mathbf{x}_{N+1}), f(\mathbf{x}^*_j)) + \tilde{c}_{N+1,j,k} \quad \text{for} \quad 1 \leq j \leq M \,, \quad \text{and} \quad i = M+1 \,, \\
&V^x_{j,i} = V^x_{i,j} \quad \text{for} \quad j \neq i \,, \quad \text{and} \quad 1 \leq i \,, j \leq M \,, \\
&V^x_{i,i} = V^k_{i,i} + \tilde{v}^*_{N+1,j,k} \,, \quad \text{for} \quad 1 \leq i \leq M \,, \\
&\Xi^x_{M+1,M+1} = \text{var}(c_j(\mathbf{x})) + \sum_{j=1}^{M} \tilde{s}_{N+1,j,c} \,, \\
&V^x_{M+1,M+1} = \text{var}(f_k(\mathbf{x})) + \sum_{j=1}^{M} \tilde{v}_{N+1,j,k} \,.
\end{aligned} \tag{61}$$

where $\tilde{v}_{N+1,j,k}$, $\tilde{v}^{*}_{N+1,j,k}$, $\tilde{c}_{N+1,j,k}$ and $\tilde{s}_{N+1,j,c}$ are the parameters of each of the M factors $\Omega(\mathbf{x}_{N+1}, \mathbf{x}^{\star})$, for $j = 1, \ldots, M$. The other terms, $\mathrm{var}(f_k(\mathbf{x}))$, $\mathrm{var}(c_j(\mathbf{x}))$ and $\mathrm{cov}(f_k(\mathbf{x}_{N+1}), f(\mathbf{x}^{*}_{j}))$ are the posterior variances of $f_k(\mathbf{x}_{N+1})$, $c_j(\mathbf{x}_{N+1})$ and the posterior covariance between $f_k(\mathbf{x}_{N+1})$ and $f_k(\mathbf{x}^{*}_{j})$.

The matrix $\mathbf{V}^{x}$, has a block structure in which only the last row and column depends on $\mathbf{x}_{N+1}$. This fact allows us to compute $v_k = (\mathbf{V}^{x})^{-1}_{M+1,M+1}$ with cost $\mathcal{O}(M^3)$ using the expressions for block matrix inversion. All these computations are carried out using the open-BLAS library for linear algebra operations.

Given the variances v_k and v_j, the only task remaining is adding the variance of the additive Gaussian noise ϵ^{k}_{N+1} and ϵ^{j}_{N+1} to obtain the final variance of the Gaussian approximations to the conditional predictive distributions of $y^{k}_{N+1} = f_k(\boldsymbol{x}_{N+1})$ and $y^{c}_{N+1} = c_j(\boldsymbol{x}_{N+1})$.

5. Final Gaussian approximation to the Conditional Predictive Distribution

5.1. Initialization and convergence of EP

When the EP algorithm computes the $\Phi(\cdot)$ and $\Omega(\cdot,\cdot)$ factors, it requires to set an initial value to all the factors that generates the Gaussians that approximate the $\Phi(\cdot)$ and $\Omega(\cdot,\cdot)$ factors. These factors, $\hat{\boldsymbol{A}}_{ij}$, $\hat{\boldsymbol{b}}_{ij}$, $\hat{\boldsymbol{C}}_{ij}$, $\hat{\boldsymbol{d}}_{ij}$, $\hat{e}_j$, $\hat{f}_j$, $\hat{ac}_j$, $\hat{bc}_j$, $\hat{cc}_j$ and $\hat{dc}_j$ are all set to be zero. The convergence criterion for stopping the EP algorithm updating the parameters is that the absolute change in all the cited parameters should be below 10^{-4}. Other criteria may be used.

5.2. Parallel EP Updates and Damping

The updates of every approximate factor $\hat{\boldsymbol{A}}_{ij}$, $\hat{\boldsymbol{b}}_{ij}$, $\hat{\boldsymbol{C}}_{ij}$, $\hat{\boldsymbol{d}}_{ij}$, $\hat{e}_j$, $\hat{f}_j$, $\hat{ac}_j$, $\hat{bc}_j$, $\hat{cc}_j$ and $\hat{dc}_j$ are executed in parallel as it is described in the work by Gerven [1]. The cavity distribution for each of the factors is computed and then the factors are updated afterwards. Once these operations are done the EP approximation is recomputed as it is described in the section 4.3.

In order to improve the convergence behaviour of EP we use the damping technique described in Minka & Lafferty [2]. We use this technique for all the approximate factors. Damping simply reduces the quantity that the factor changes in every update as a linear combination between the old parameters and the new parameters. That is, if we define the old parameters of the factor to be updated as u_{old}, the new parameters as u_{new} and the updated factor as u, then the update expression is:

$$u = \theta u_{new} + (1 - \theta) u_{old}\,. \tag{62}$$

Where θ is the damping factor whose initial value is set to be 0.5, this factor controls the amount of damping, if this value is set to be one then no damping is employed. This factor is multiplied by 0.99 at each iteration, reducing the amount of change in the approximate factors in every iteration of the Bayesian Optimization. An issue that happens during the optimization process is that some covariance matrices become non positive definite due to a high large step size, that is, a high value of θ. If this happens in any iteration, an inner loop executes again the update operation with $\theta_{new} = \theta_{old}\ /\ 2$ and the iteration is repeated. This inner loop is performed until the covariance matrices become non positive definite.

6. Sensitivity analysis of the sampled Pareto set size

In this section, we present the details of a sensitivity analysis of PESMOC with respect to the number of points included in each sample of the Pareto set $\mathcal{X}^\star_{(m)}$. We have considered the 6-dimensional synthetic problem described in the main manuscript. In this problem we have 6 black-boxes that are sampled from a GP prior. From these, 4 are objectives and 2 are constraints. We performed 100 experiments in which we evaluate 100 times the black-boxes and report average results. We consider different set sizes for $\mathcal{X}^\star_{(m)}$. Namely, 5, 10, 25, 50, 75 and 100. Figure 6 shows the average relative difference in log scale of the hyper-volume of the recommendation made w.r.t. the hypervolume of the actual soulution, at each iteration of the optimization process, for each Pareto set size. Furthermore, Table 1 shows the average time required to compute the next evaluation using PESMOC, for each different size of the Pareto set $\mathcal{X}^\star$ considered.

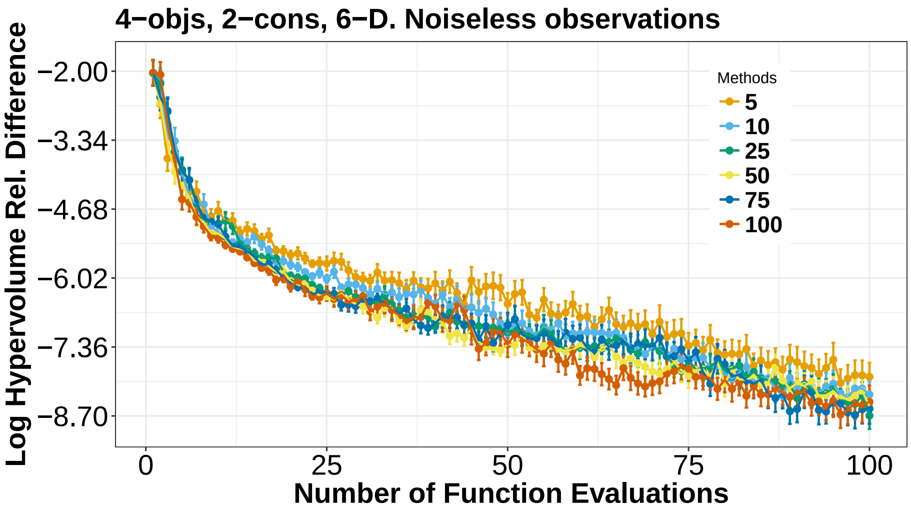

Figure 1: Performance of PESMOC in the synthetic experiment for different Pareto set sizes.

We observe that by including more points in the Pareto Set, in general, the reuslts of PESMOC improve, as expected. However, after including around 25 points the results saturate and no further improvement in performance is observed. In general one should include as many points as possible in each $\mathcal{X}^\star_{(m)}$. However, including more points in each sample $\mathcal{X}^\star_{(m)}$ also increases the computational cost, as described in Table 1. We observe that choosing 50 points shows a good trade-off between performance and computational time. In particular, this value gives good results and is not significantly more expensie than considering 25 points.

7. Sensitivity analysis of the number of Monte Carlo samples

In this section, we present the details of a sensitivity analysis of PESMOC with respect to the number of Monte Carlo samples considered for approximating the expectation required for evaluating the acquisition function. We have considered the 4-dimensional synthetic problem described in the main manuscript. In this problem we have 4 black-boxes that are sampled from a GP prior. From these, 2 are objectives and 2 are constraints. We performed 100 experiments in which we optimize the black-boxes and report average results. We consider a different number of Monte Carlo samples. Namely, 2, 5, 10, 20, 30, 50, 80

Table 1: Average time in seconds required to choose the next evaluation for each size of the Pareto Set sample $\mathcal{X}^{\star}_{(m)}$. These times do not include the GP fit, in contrast with the average times of the 4 dimensional problem scenario shown in Section 4.2 of the main manuscript.

	Time	
Size of $\mathcal{X}^{\star}_{(m)}$	**Mean**	**Standard Deviation**
5	95.98	10.67
10	99.05	11.53
25	109.48	10.92
50	133.02	11.68
75	159.90	15.34
100	190.25	25.71

and 100. Figure 7 shows the average relative difference in log scale of the hyper-volume of the recommendation made w.r.t. the hypervolume of the actual soulution, at each iteration of the optimization process, for each number of Monte Carlo samples.

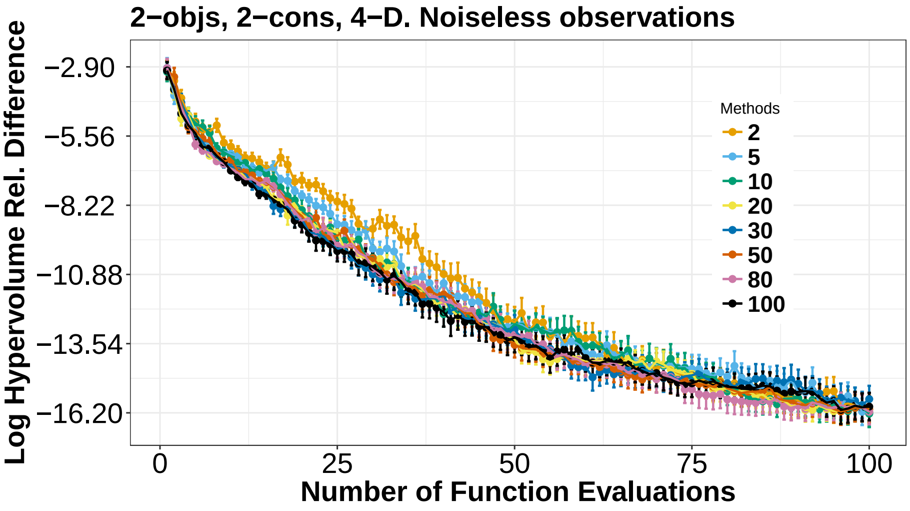

Figure 2: Performance of PESMOC in the synthetic experiment for different number of Monte Carlo samples M.

We observe that, initially, increasing the number of samples makes the reuslts of PESMOC improve, as expected. However, after considering around 10 samples, no further

Table 2: Average time in seconds per iteration as a function of M the number of Monte Carlo samples.

Number of MC samples	**Average Time in Seconds**	**Standard Deviation**
2	0.037462402153011	0.00449420307187503
5	0.044145710015292	0.0053619606855024
10	0.053984149718284	0.00240231706760828
20	0.075339632344249	0.00319199234053008
30	0.096420479202285	0.00432255645582367
50	0.13980684764385	0.0167542913437566
80	0.29173432607651	0.250478316827395
100	0.36593869268893	0.278688827201095

improvement in performance is observed. In general one should include as many number of Monte Carlo samples as possible. However, including more samples also increases the computational cost. Table 2 shows the average time per iteration as a function of M, the number of Monte Carlo samples. From these results, we conclude that that choosing 10 samples shows a good trade-off between performance and computational time.

8. Percentage of infeasible solutions in benchmark experiments

We show in the next figure the percentage of infeasible solutions for the different benchmark experiments shown in the main manuscript. The plots show the percentage of infeasible solutions for every iteration of Bayesian Optimization on these problems for every tested acquisition function. A higher percentage of infeasible solutions implies that the suggestions provided by the Bayesian Optimization criterion were not located in the feasible region and hence, are poorer that feasible ones. We remind that to be an infeasible point means that the probability of satisfying any of the constraints in that point was lower in at least one of them than 1 minus δ, typically set to 0.05 in these experiments. This fact may occur, for example for the PESMOC criteria, because the criterion has selected a point that may be close to the border of the feasible region and may be a good point in order to gain information about the Pareto Set, but unfortunately, is infeasible as it lies in the infeasible side of the border, same analogy can be applied for BMOO, which can suggest an infeasible point if it considers it good for optimizing the multiobjective problem. In the random case this does not apply, as the approach performs pure exploration. At the beggining, the infeasibility percentage of solutions is higher due to he fact that the shape of the constraints is unknown and all the criteria are basically exploring the space, hence, they suggest infeasible solutions with a high probability until the shape of the constraints is reasonably known so the approaches can take the constraints into account for suggesting new points. Then, the infeasibility percentage drops down as iterations are computed.

It is shown that, in average, both PESMOC approaches outperform the other alternatives of multi-objective constrained optimization in infeasibility percentage. The TNK problem has the particular feature that it contains an optimum near one of its constraints, that is the reason why the infeasibility grows around iteration 20 in all the approaches. This is because as the optimum is more located and the constraint is defined, the methods tend to search in that area, leaving unfeasible results. Once the constraint is defined, they do not search outside of the Feasible Space anymore, and hence, the unfeasible results dissapear. TwoBarTruss problem has the same nature as TNK, with the optimum lying in the frontier of the feasible and infeasible space. We, once again, see how PESMOC decoupled explores massively this area, giving lots of infeasible results in iterations 20 to 30. PESMOC decoupled, as it evaluates every black box separately, learns the shape of these constraints and, as the main manuscript states, it delivers a better solution that the other approaches in less evaluations once it knows the shape of these constraints and it is able to suggest points inside of the feasible space and close to the Pareto Set.

References

[1] T. Heskes M. V. Gerven, B. Cseke and R. Oostenveld. Bayesian source localization with the multivariate laplace prior. In *Advances in neural information processing systems*, pages 1901–1909, 2009.

[2] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In

Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence, pages 352–359. Morgan Kaufmann Publishers Inc., 2002.

[3] T. P. Minka. Expectation Propagation for Approximate Bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369, 2001.

[4] M. Opper and C. Archambeau. The Variational Gaussian Approximation Revisited. *Neural computation*, 21(3):786–792, 2009.

[5] C. E. Rasmussen. Gaussian Processes for Machine Learning. 2006.

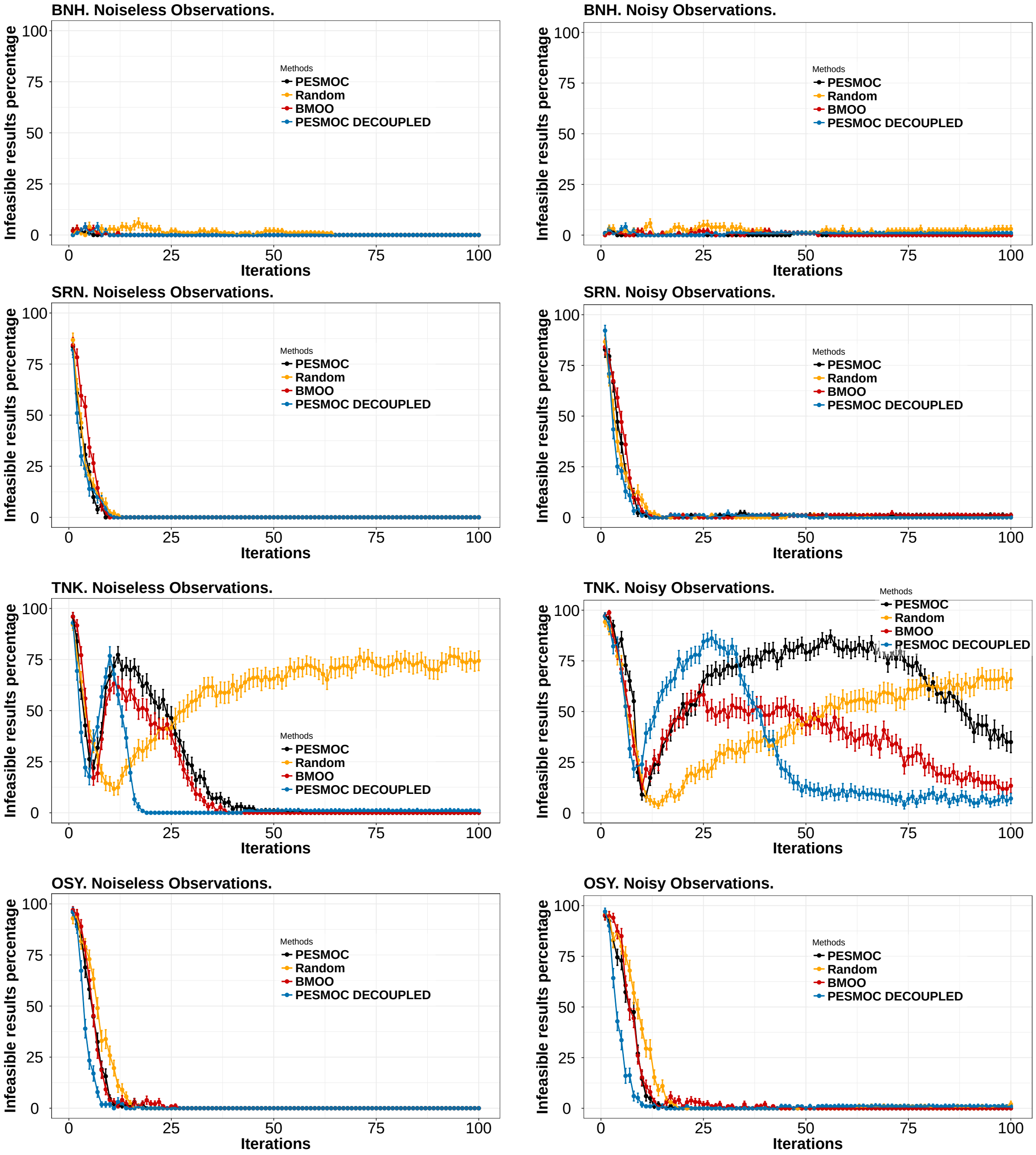

Figure 3: Percentage of infeasible results in every iteration of experiments BNH, SRN, TNK and OSY.

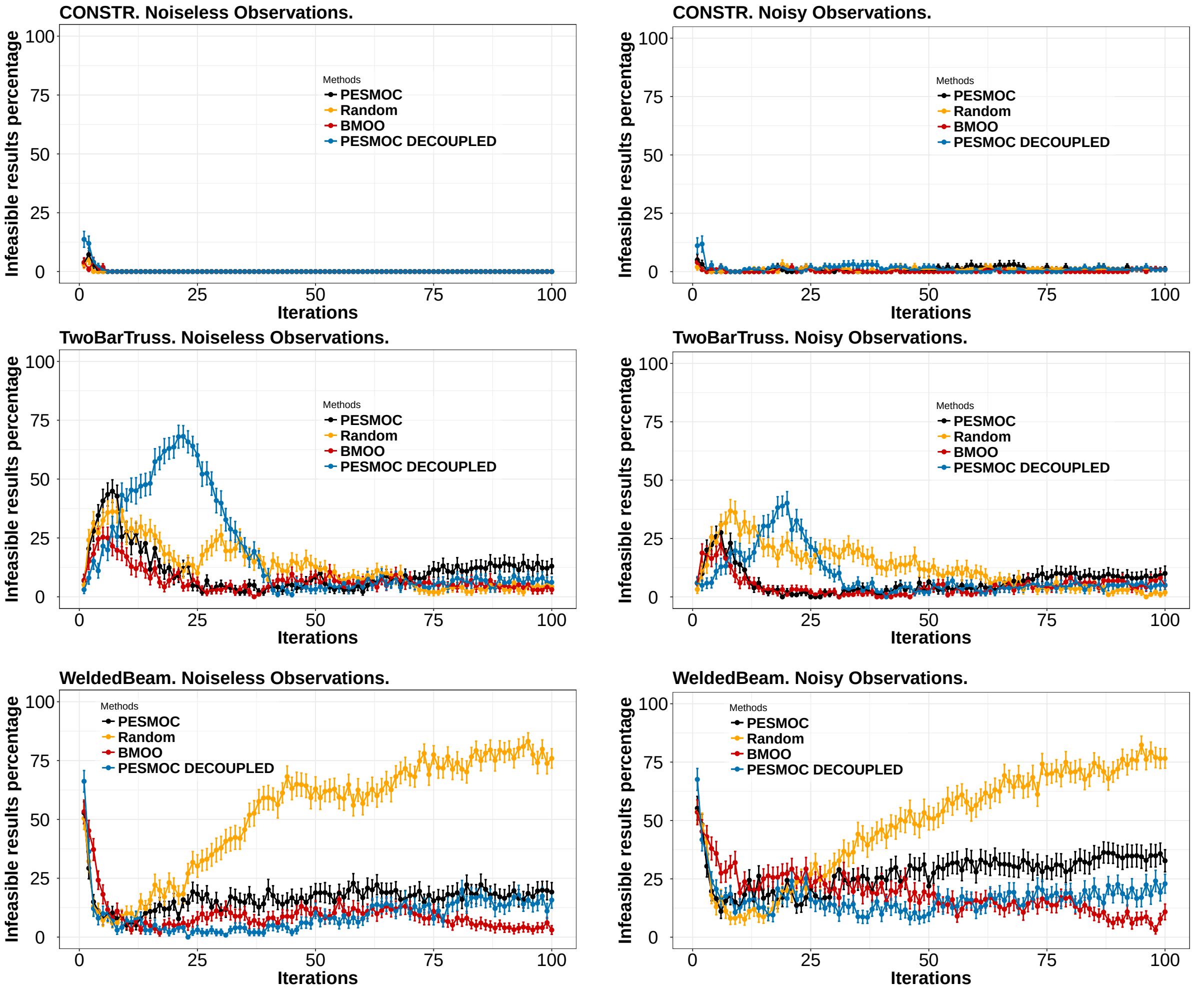

Figure 4: Percentage of infeasible results in every iteration of experiments CONSTR, TwoBarTruss and WeldedBeam.