



Personalized Bayesian optimization for noisy problems

Xilu Wang¹ · Yaochu Jin¹

Received: 2 October 2022 / Accepted: 18 February 2023 / Published online: 5 April 2023
© The Author(s) 2023

Abstract

In many real-world applications of interest, several related optimization tasks can be encountered, where each task is associated with a specific context or personalized information. Moreover, the amount of available data for each task may be highly limited due to the expensive cost involved. Although Bayesian optimization (BO) has emerged as a promising paradigm for handling black-box optimization problems, addressing such a sequence of optimization tasks can be intractable due to the cold start issues in BO. The key challenge is to speed up the optimization by leveraging the transferable information, while taking the personalization into consideration. In this paper, optimization problems with personalized variables are formally defined at first. Subsequently, a personalized evolutionary Bayesian algorithm is proposed to consider the personalized information and the measurement noise. Specifically, a contextual Gaussian process is used to jointly learn a surrogate model in different contexts with regard to the varying personalized parameter, and an evolutionary algorithm is tailored for optimizing an acquisition function for handling the presence of personalized information. Finally, we demonstrate the effectiveness of the proposed algorithm by testing it on widely used single- and multi-objective benchmark problems with personalized variables.

Keywords Personalized Bayesian optimization · Transfer learning · Noisy optimization · Expensive optimization

Introduction

Expensive black-box optimization problems are ubiquitous in real-world engineering and design optimization problems, where the problem is unknown besides the observed function values and the available number of function evaluations is severely limited. Bayesian optimization (BO) has emerged as an efficient method for optimizing black-box problems due to its data efficiency, which originates from a surrogate model approximating the true expensive objective functions [17]. Traditional BO typically focuses on a single task each time and starts a search from scratch assuming each task is isolated, which is called cold start [20]. However, several related tasks may be encountered in some real-world applications. As the standard BO does not consider task relatedness, a number of costly function evaluations are required to construct an efficient surrogate model even in cases several related tasks are encountered, making it inapplicable in practice [18].

Recently, transfer learning methods have been incorporated with BO to alleviate the cold start issue by leveraging the information from related tasks. A representative work is multi-task Gaussian processes [24], which is proposed to extend the GP to learn the inter-task similarities. Multi-task Gaussian processes have been successfully adopted various applications, such as hyperparameter optimization of machine learning models [20] and biomedical engineering [7]. The problem setting considered in this paper is motivated by problems in practice, such as optimizing a complex system subject to varying environmental conditions, advertising on different web pages, and developing personalized treatment for patients. In these problems, a sequence of related black-box optimization tasks should be addressed subjected to given context/personalized information in each round, which can be defined as expensive optimization with personalized/contextual information. For example, recently personalized medicine has been developed to shed light on the potential advantages of incorporating personal information, i.e., each person's unique clinical, environmental, and genetic information, into the treatment. Take the transcranial alternating current stimulation (tACS) [1] as an example; tACS utilizes an alternating current delivered via multiple electrodes placed on the scalp, which is capable of propa-

✉ Yaochu Jin
yaochu.jin@uni-bielefeld.de
Xilu Wang
xilu.wang@uni-bielefeld.de

¹ Faculty of Technology, Bielefeld University, Bielefeld 33619, Germany

gating through the scalp and modulating the activity of the underlying neurons. However, it is problematic to identify the optimal tACS parameters, i.e., the current and frequency, for different individuals, as tailoring the treatment based on tACS is financially costly and time-consuming and may be perturbed by noise. In each round, the personalized/contextual information is available for a patient, and the aim is to search for the optimal parameters for tACS simulations.

We formally define the considered optimization problems as follows:

$$\mathbf{x}^*(p) = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}, p), \quad (1)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_d)$ is the decision vector with d decision variables, \mathcal{X} denotes the decision space, p denotes a variable that indicates the personalized/contextual information, such as personal characteristics and contextual information from the environment, and f denotes the expensive black-box objective function, where each function evaluation requires to perform time- and resource-consuming simulations or physical experiments. Note that the black-box function f has no closed form, but can be evaluated at any location \mathbf{x} in the domain. As a result, the evaluation may contain measurement noise ε , which can be formulated as

$$y(\mathbf{x}, p) = f(\mathbf{x}, p) + \varepsilon, \quad (2)$$

where y denotes the noisy observation/output of the costly simulations or experiments. For instance, in tACS simulations, the decision variables include the alternating frequency and the current strength, and p is the personalized information of each patient. For this problem, the best inferred parameter combination differs along the value of p due to the individual difference. Hence, the optimal parameter $\mathbf{x}^*(p)$ is not defined globally, but specifically to variable p . Given a group of participants associated with their personal characteristic, the goal is to find the optimal tACS parameters for each individual. Moreover, the objective function f is black-box that can only be evaluated by the time-consuming tACS simulations.

In this work, we attempt to address the above-mentioned computationally expensive optimization problems with personalized information and measurement noise. To this end, a personalized Bayesian optimization algorithm (PBO-EA) with an evolutionary algorithm (EA) to solve the inner optimization in BO is proposed to take the personalized variable into consideration. More specifically, personalized Gaussian processes (PGPs) are employed to jointly learn surrogate models across different contexts by utilizing the personalized information. Hence, for a test data with a specific value of personalized variable, the PGP can provide predictions associated with a confidence level. For the given personalized information, a modified evolutionary algorithm is used

to optimize an acquisition function for identifying new samples. Both single-objective and multi-objective benchmark problems are modified to test the proposed method on the considered problem setting.

In the rest of the paper, Bayesian optimization for black-box global optimization problems, including Gaussian processes and acquisition functions, is introduced first. Following that, the proposed personalized evolutionary Bayesian optimization algorithm is presented. To validate the effectiveness of the proposed algorithm for expensive optimization problems with personalized variables and output noise, the single-objective and multi-objective benchmark suites are modified and the experimental results are summarized in “[Experimental studies](#)”. Finally, we draw a conclusion and present some lines of future work.

Bayesian optimization

In this section, Bayesian optimization [17] is briefly introduced, including its two key components, i.e., Gaussian processes and acquisition functions. Suppose we aim to optimize an expensive black-box function $f : \mathcal{X} \rightarrow \mathbb{R}$, where $\mathcal{X} \subset \mathbb{R}^d$ is a compact and convex set. We can access only the possible noisy evaluations $y = f(\mathbf{x}) + \varepsilon$ at any query point \mathbf{x} . Formally, the goal is to find the global optimum

$$\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}). \quad (3)$$

However, the limited evaluation budget resulting from the costly function evaluations makes it hard for an algorithm to converge to the global optimum. Bayesian optimization has been emerged as a popular methodology for global optimization of expensive black-box functions due to its high sample efficiency. A key component of Bayesian optimization is a surrogate model that trained by the observed data to approximate the true objective function. The surrogate model can replace the expensive evaluation by providing predictions on the queried locations. Bayesian optimization often adopts Gaussian processes (GPs) as the surrogate model, especially in the small data regime. Gaussian processes provide predictions associated with uncertainty estimates that is important to guide the global search. Afterwards, an acquisition function (AF) is carefully designed to balance the exploration and exploitation based on the predictions. Instead of optimizing the expensive objective function, the acquisition function is optimized to identify the next query point. In the following, more details of Gaussian processes and acquisition functions are presented.

Gaussian processes

The Gaussian process (GP) is characterized by its prior mean function $m(\cdot)$ and covariance function (or kernel matrix) $k(\cdot, \cdot)$ [16]. Consider a finite collection of data pairs (\mathbf{X}, \mathbf{y}) of the unknown function $y = f(\mathbf{X}) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$, where $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N]^T$, $\mathbf{y} = [y^1, y^2, \dots, y^N]^T$, we assume that the observed data are drawn from a multivariate Gaussian distribution with a Gaussian prior

$$y \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{4}$$

Therefore, we obtained the predictive distribution for y at a new observation \mathbf{x} that also follows a Gaussian distribution. Its mean μ and variance σ^2 are given by

$$\begin{aligned} \mu(\mathbf{x}) &= K(\mathbf{x}, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{y} \\ \sigma^2(\mathbf{x}) &= K(\mathbf{x}, \mathbf{x}) - K(\mathbf{x}, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I})^{-1} K(\mathbf{X}, \mathbf{x}), \end{aligned} \tag{5}$$

where $k(\mathbf{x}; \mathbf{X})$ presents a correlation vector between \mathbf{x} and each element \mathbf{x}^i in \mathbf{X} , $K(\mathbf{X}; \mathbf{X})$ is a covariance matrix whose element is calculated by the covariance (kernel) function. Commonly used covariance functions include the squared exponential and Matérn covariance functions.

Acquisition functions

Once the GP is constructed, the next step is to select the next query point, i.e., to select a new point to be evaluated using the expensive function. In Bayesian optimization, instead of optimizing the true objective function, an AF is optimized to identify new samples. Generally, optimizing the AF by evolutionary algorithms is called evolutionary Bayesian optimization (EBO). AFs typically incorporate both the mean and the variance of the GP prediction to achieve a trade-off between exploration and exploitation. A fruitful line of research has been done on designing acquisition functions, including expected improvement (EI) [11] and lower confidence bound (LCB) [8]. Moreover, some extensions of EI accounting for measurement noise have been proposed.

Let f^* denote the optimum obtained so far, and $\Phi(\cdot)$ and $\phi(\cdot)$ denote the normal cumulative distribution function (CDF), and probability density function (PDF) of the standard normal random variable, respectively. A commonly used acquisition function is called lower bound confidence (LCB) [19]. LCB is designed to balance the exploration and exploitation by combining the uncertainty with the predicted objective values

$$LCB(\mathbf{x}) = \mu(\mathbf{x}) - \kappa \sigma(\mathbf{x}), \tag{6}$$

where κ presents a trade-off constant. LCB implicitly prefers points whose predicted mean value μ is small and the corresponding standard deviation σ is large.

Alternatively, *expected improvement* (EI) [14] calculates the expected improvement with respect to f^*

$$\begin{aligned} EI(\mathbf{x}) &= \mathbb{E}[\max(0, f^* - f(\mathbf{x}))] \\ &= (f^* - \mu(\mathbf{x})) \Phi\left(\frac{f^* - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right) + \sigma(\mathbf{x}) \phi\left(\frac{f^* - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right), \end{aligned} \tag{7}$$

where \mathbb{E} denotes the expectation value, and Φ and ϕ are the Gaussian CDF and PDF, respectively.

For noisy observations, i.e., the objective function is subject to noise, the standard EI will face two key challenges [15]: (1) the current best solution is not well defined, and (2) the prediction uncertainty associated with the current best fitness values is not accounted for. Hence, a variant of EI for handling noisy optimization, known as Augmented EI (AEI) [10], is introduced

$$\begin{aligned} AEI(\mathbf{x}) &= \\ E[\max(\mu(\mathbf{x}^*) - f(\mathbf{x}), 0)] &\cdot \left(1 - \frac{\sigma_\epsilon}{\sqrt{\sigma^2(\mathbf{x}) + \sigma_\epsilon^2}}\right), \end{aligned} \tag{8}$$

where \mathbf{x}^* stands for the current ‘effective best solution’, which is determined as explained below, σ_ϵ^2 denotes the noise level. In Eq. (8), the expectation is conditional given the past data and given estimates of the correlation parameters.

To determine the effective best solution \mathbf{x}^* , we introduce a utility function, denoted as $Utility(\mathbf{x})$, to account for the uncertainty associated with the predicted objective values. In general, the form of the utility function may be selected according to the user’s preference. In AEI, the following formula is used:

$$Utility(\mathbf{x}) = -\mu(\mathbf{x}) - c\sigma(\mathbf{x}), \tag{9}$$

where c is a constant that can reflect the degree of risk aversion. We select $c = 1$ as our default, while implies a willingness to trade 1 unit of the predicted objective value for 1 unit of the standard deviation of prediction uncertainty. Consequently, the new sample is identified by maximizing $Utility$ function and $\mu(\mathbf{x}^*)$ denotes the prediction on \mathbf{x}^* provided by the surrogate model.

Similarly, in [9], another variant of EI criterion, called Profile EI (PEI), for contextual optimization is proposed. Recall that the standard EI measures the expected improvement on the current best location. However, for a given new context, the current best location may misguide the search for the unseen context. Hence, an alternative for the current best

location is introduced as follows:

$$T(\mathbf{x}) := \min (\max \mu(\mathbf{x}), \mathbf{x}^*), \quad (10)$$

where $m(x)$ is the mean value of x predicted by the personalized Gaussian process. Consequently, the PEI is

$$PEI(\mathbf{x}) \equiv E [\max (T(\mathbf{x}) - f(\mathbf{x}), 0)]. \quad (11)$$

Personalized evolutionary Bayesian optimization

Despite the abundance of applications of Bayesian optimization, a few methods have been proposed to investigate expensive black-box function with personalized variables and measurement noise. In this section, a personalized evolutionary Bayesian optimization is introduced. While the existing work mainly focused on single-objective optimization, we attempt to extend the proposed algorithm to multi-objective optimization problems with personalized variables. The pseudo code of the proposed algorithm for single/multi-objective problems is outlined in Algorithm 1.

Algorithm 1 Personalized Bayesian Optimization

Input: FE^{max} : the maximum number of the slow objective function evaluations;

Output: the optimal solution;;

- 1: Initialization: use the Latin Hypercube Sampling method [13] to sample N points $\mathbf{X} = \{\mathbf{x}_0^i\}_{i=1}^N$ to be evaluated on the true objective functions; set $D^0 = (\mathbf{X}, \mathbf{Y})$ and $FE = N$;
 - 2: //Construct surrogate models//
 - 3: Train a personalized GP for each objective function;
 - 4: **while** $FE \leq FE^{max}$ **do**
 - 5: **if** PBO-EA **then**
 - 6: Employ GA to optimize EI (Eq. 7) and select a new sample
 - 7: **end if**
 - 8: **if** PGP-MOEA **then**
 - 9: Employ an MOEA (we use RVEA) to optimize $mLCB$ (Eq. 13) and select u new samples
 - 10: **end if**
 - 11: //Update the Gaussian process//
 - 12: Add the newly evaluated data into the training dataset and re-train the surrogate model;
 - 13: **if** Optimize single-objective problems **then**
 - 14: Update $FE = FE + 1$;
 - 15: **end if**
 - 16: **if** Optimize multi-objective problems **then**
 - 17: Update $FE = FE + u$;
 - 18: **end if**
 - 19: **end while**
 - 20: Return the optimized solution.
-

Personalized Gaussian processes

The commonly used surrogate model in BO, Gaussian processes, does not take the personalized information into consideration. That is, the standard surrogate model approximates $f(\mathbf{x})$ instead of $f(\mathbf{x}, p)$. To model the objective function $f(\mathbf{x}, p)$ in presence of personalized variables, there are three possible methods: (1) we assume that each individual with a specific value of the personalized variable is not related to each other. Under this assumption, a single Gaussian process should be constructed for each individual and BO should be performed separately on each individual. Unfortunately, we only have very limited number of data for each individual, posing challenges for constructing separate surrogate models with good quality. (2) Alternatively, we can ignore the influence of the personalized variables on the objective function by assuming that all individuals with varying personal characteristics share a same objective function. Hence, we can reduce the optimization problem with personalized parameters to a standard optimization problem. However, this is not the case in practical problems. (3) we assume that these individuals with different personalized information share some similarities. Hence, we can learn a personalized Gaussian process by leveraging the similarity between different individuals. For the considered problem setting, the observed data for a number of individuals with different personal characteristics can be augmented together to train a single GP model, enhancing the estimate of the model parameters. To achieve this, a contextual GP [12] is employed in our work to jointly learn a personalized surrogate model, called a personalized GP (PGP).

The contextual GP is designed for contextual bandit problem where varying environmental conditions are considered. To leverage the contextual information, an additional kernel function is defined over the context space, which is used in conjunction with a kernel function over the input features. Such a composite kernel function in the contextual GP allows us to analysis different contexts within a single GP model. Motivated by the success of the contextual GP, we present a product kernel for the optimization with personalized parameters. This product kernel is constructed using kernels defined over the decision variables and the context (the personalized parameter). Hence, the covariance becomes

$$k(\{\mathbf{x}_i, p_i\}; \{\mathbf{x}_j, p_j\}) = k(\mathbf{x}_i, \mathbf{x}_j) \otimes k(p_i, p_j), \quad (12)$$

where $k(\mathbf{x}_i, \mathbf{x}_j)$ and $k(p_i, p_j)$ are the kernel over search space and the context, respectively. \otimes denotes Kronecker product. The core idea behind this product kernel is to describe the data-context pairs by constructing separate kernels on different variables. As a result, the observed data in one context can impact the predictions in another context as the model can capture the correlation between multiple

optimization tasks with varying contexts. In this work, the squared exponential kernel is used.

Optimization of acquisition functions

Having constructed the contextual Gaussian process, the next step is the design and optimization of an AF to search for the new sample conditional on the personalized parameter. In our work, while EI is adopted as the AF for single-objective problems with personalized variables, a variant of LCB is used in multi-objective optimization problems (MOPs). Consider a m -objective optimization problem with personalized variables, one personalized GP is constructed for each objective. Hence, the PGPs can predict mean and variance of a candidate solution \mathbf{x} on all objectives, denoted as $\sigma(\mathbf{x}) = \{\sigma_i(\mathbf{x})\}_{i=1}^m$ and $\mu(\mathbf{x}) = \{\mu_i(\mathbf{x})\}_{i=1}^m$, respectively. In this way, the original LCB shown in Eq. (6) can be extended to MOPs as follows:

$$mLCB(\mathbf{x}) = \mu(\mathbf{x}) - \kappa\sigma(\mathbf{x}), \quad (13)$$

where k is a parameter to manage the trade-off between exploration and exploitation. Note that $mLCB(\mathbf{x})$ is a vector with a length of m .

Generally, AFs are multi-modal, non-convex, and difficult to be optimized. While various kinds of optimization methods have been introduced to solve the inner optimization within the BO framework, there is a lack of studies exploring the optimization of acquisition functions in the presence of the personalized/contextual information and the measurement noise. In the considered problems, the personalized information for each individual is given before performing BO. Hence, the value for the personalized parameter is fixed when optimizing the acquisition function with the surrogate model.

Evolutionary algorithms (EAs), a class of population-based methods, have been successfully applied to various non-convex optimization problems [3]. EAs' efficiency originates from the fact that they are insensitive to local optima and do not require gradient information. While EAs have emerged as a powerful method to optimization AFs in BO [25], they are not applicable to the problem considered here as the personalized/contextual information is not considered. For the considered problem, the personalized information is given in each round of BO. Hence, the value of the personalized variable is fix when optimizing the AF. Without loss of generality, we suggest a standard genetic algorithm (GA) and a reference vector guided EA (RVEA) [2] as the single-objective and multi-objective optimizer for the acquisition function optimization given the personalized information, respectively. Specifically, in both GA and RVEA, a real number coding scheme, the simulated binary crossover [4], and the polynomial mutation [5] are employed.

Experimental studies

Test problems

We consider nine commonly used single-objective optimization problems [15,21], including six two-dimensional problems, i.e., Goldstein, Rosenbrock, Branin, Six-hump camel, Dropwave, Beale and Ackley function with ten decision variables, and Hartmann function with three and six decision variables, respectively. The DTLZ [6] test suite is used to test the performance of the proposed algorithm on MOPs. The modified counterparts for DTLZ1 and DTLZ3 by reducing the complexity to a reasonable level are denoted as DTLZ1a and DTLZ3a, respectively, and shown in the Supplementary material. As recommended in [6], the number of decision variables for the test instances is set to $n = m + K - 1$, where $K = 5$ is adopted for DTLZ1, $K = 10$ is used for DTLZ2 to DTLZ6 as well as DTLZ3a, and $K = 20$ is employed in DTLZ7. m represents the number of objectives. Here, we set $m = 3$.

Note that these benchmarks are designed without considering the personalized variables. Hence, to involve personalized variables into the objective function, we vary the parameters in the benchmark problems, so that different values for the personalized variable will generate different objective functions. For example, the original Rosenbrock function is

$$f(\mathbf{x}) = \left[\sum_{i=1}^{D-1} \left(100 (\bar{x}_{i+1} - \bar{x}_i^2)^2 + (1 - \bar{x}_i)^2 \right) \right], \quad (14)$$

where $\bar{x}_i = 15x_i - 5$, for all $i = 1, 2, 3, 4$. To generated a variant of Rosenbrock function for the considered problem, we treat a constant in Eq. (14) as the personalized variable p . The different value of p is calculated by multiplying the original value of the selected constant with values of samples s drawn from a normal distribution. Hence, the corresponding Rosenbrock function is formulated as

$$f(\mathbf{x}, p) = \left[\sum_{i=1}^{D-1} \left(p (\bar{x}_{i+1} - \bar{x}_i^2)^2 + (1 - \bar{x}_i)^2 \right) \right] \quad (15)$$

with $p = 100 * s$.

On the other hand, the observations in the real-word experiments may be noisy. To test the influence of the noise, we add a Gaussian noise $noise \sim N(0, \sigma_\epsilon^2)$ to the output of each benchmark problem. Varying σ_ϵ^2 of the distribution from 0 to 1 can change the level of noise. Hence, the final Rosenbrock function used in our experiments is

$$f(\mathbf{x}, p) = \left[\sum_{i=1}^3 \left(p (\bar{x}_{i+1} - \bar{x}_i^2)^2 + (1 - \bar{x}_i)^2 \right) \right] \times (1 + \sigma_\epsilon) \quad (16)$$

with $p = 100 * s$. In our experiments, we test each algorithm on the benchmark problems with different levels of noise, i.e., $\sigma_\varepsilon^2 = 0, 0.05, 0.1, 0.5$ and 1 . Moreover, ten personalized values are generated for each benchmark problem to test each algorithm on different optimization problems with varying personalized information.

Similar to personalized single-objective benchmark problems, we also modified the selected multi-objective test problems by treating a parameter in the original benchmark problem as the personalized variable.

Experimental settings

Hence, we tested a standard evolutionary algorithm (i.e., genetic algorithm), and a BFGS-based sequential quadratic programming (SQR) provided by an MATLAB function. We denote these two methods as PBO-EA and PBO-SQR, respectively. Note that the expected improvement (EI) is adopted as the acquisition function to identify new samples in both methods.

Based on the above analysis, we attempt to solve the optimization problems with the personalized variable from the perspective of the surrogate model, the design, and the optimization of the acquisition function. To validate each component in the proposed personalized Bayesian optimization framework, shown in Algorithm 1, the experiment include four major parts:

1. To validate the effectiveness of the personalized GP, the proposed PBO-EA is compared with the standard BO with the standard GP. While the standard GP ignores the personalized information, the contextual Gaussian process in PBO-EA attempts to jointly learn an approximation of the true objective function over different contexts.
2. A variant of PBO-EA is introduced to test the advantage of the use of GA. While PBO-EA use GA to optimize EI, the variant employs a BFGS-based sequential quadratic programming (SQR) provided by a MATLAB function, called PBO-SQR.
3. Different AFs are tested on the problems with personalized variable and measurement noise. In addition to EI, the two variants of EI for handling measurement noise, i.e., AEI and PEI, and LCB are adopted, denoted as PBO-AEI, PBO-PEI, and PBO-LCB, respectively.
4. To test the effectiveness of PGP on MOPs with personalized variables, we suggest to compare a PGP-assisted multi-objective evolutionary algorithm (MOEA) and a GP-assisted one, denoted as PGP-MOEA and GP-MOEA, respectively. In both PGP-MOEA and GP-MOEA, a commonly used MOEA (we use RVEA [2]) is performed to optimize *mLCB* to select u new samples. Parameter k in *mLCB* is set to 2 and u is set to 5.

We run each algorithm on each benchmark problem for ten independent times, and the Wilcoxon rank sum test [23] is calculated to compare the mean over ten independent runs obtained by PBO-EA and by the compared algorithms at a significance level of 0.05. Symbol “(+)” indicates that the proposed algorithm outperforms the compared algorithm statistically significantly, while “(−)” means that the compared algorithm performs better than PBO, and “(≈)” means that there is no significant difference between them. The mean and standard deviation (Std) of ten runs for the single-objective problems with different noise levels obtained by the algorithms are recorded in Tables 1, 2, 3, 4, 5 and Tables SI–SV in the Supplementary material. Note that we use notation ‘S’ to indicate tables and figures in the Supplementary material to avoid confusion. For MOPs, the hypervolume (HV) [22] metric is adopted to assess the performance of the algorithms. HV provides a combined information of the convergence and diversity of the obtained set of solutions. All HV values presented in this work are normalized to [0, 1]. Algorithms achieving a larger HV value are better. The results in terms of HV values obtained by PGP-MOEA and GP-MOEA are summarized in Table 6.

All algorithms are implemented in Matlab R2019a on an Intel Core i7 with 2.21 GHz CPU. The general parameter settings in the experiments are given as follows:

1. We initial ten personalized parameters. For each personalized value, the size of the initial training data is ten. Hence, there are $10 * 10$ initial data in total.
2. We randomly generate ten personalized values, i.e., $\mathbf{p} = [p_1, p_2, \dots, p_{10}]$, and the algorithm runs ten iterations for each personalized value. Hence, the maximum number of function evaluations $FE = 10 * 10 + 10 * 10$.
3. As ten personalized values (contexts) are considered for each benchmark problem, there are $10 * 12$ optimization tasks in total.
4. For evolutionary algorithm: the population size is set to 60 and the maximum number of generations is set to 20.
5. One new sample is selected for single-objective problems, and five new samples ($u = 5$) are selected for MOPs.

Experimental results

1. Comparison of surrogate models: The results obtained by the standard BO and PBO-EA on single-objective problems with different levels of noise are presented in Tables 1, 2, 3, 4, 5 and Tables SI–SV, respectively. We can see that the algorithm with the personalized GP, i.e., PBO-EA, shows better overall performance than the standard BO. More specifically, PBO-EA shows significantly better or similar performance than the standard BO on 74 out of 80 optimization tasks without noise, indicating the

Table 1 Statistical results obtained by PBO-EA, BO, PBO-SQR, PBO-LCB, PBO-AEI, and PBO-PEI on benchmark problems with $noiselevel = 0$

Problem	PBO-EA		BO		PBO-SQR		PBO-LCB		PBO-AEI		PBO-PEI	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Ackley	15.4	0.33	16.1	≈ 0.44	14.6	≈ 6.73	15.9	≈ 0.41	16.1	≈ 0.29	16.1	≈ 0.29
	8.37	0.19	8.43	≈ 0.24	7.10	≈ 2.30	8.21	≈ 0.36	8.43	≈ 0.29	8.43	≈ 0.29
	13.1	0.41	13.3	≈ 0.39	13.0	≈ 1.01	13.1	≈ 0.31	13.4	≈ 0.30	13.4	≈ 0.21
	6.11	0.12	5.97	≈ 0.15	5.73	≈ 0.65	6.25	≈ 0.24	6.19	≈ 0.34	6.19	≈ 0.22
	12.2	0.22	12.2	≈ 0.23	12.4	≈ 0.23	12.2	≈ 0.23	12.2	≈ 0.35	12.2	≈ 0.34
	11.9	0.36	11.5	≈ 0.35	12.1	≈ 0.27	12.2	≈ 0.35	11.6	≈ 0.21	12.2	≈ 0.21
	9.37	0.28	9.34	≈ 0.23	8.78	≈ 0.64	9.55	≈ 0.21	9.54	≈ 0.29	9.43	≈ 0.20
	16.1	0.21	16.7	≈ 0.33	16.8	≈ 0.41	16.6	≈ 0.31	16.6	≈ 0.25	16.3	≈ 0.41
	3.67	0.23	3.90	≈ 0.38	3.94	≈ 0.41	3.90	≈ 0.22	4.20	– 0.16	3.91	≈ 0.23
	13.9	0.37	14.3	≈ 0.33	14.1	≈ 0.21	14.2	≈ 0.13	14.2	≈ 0.25	14.3	≈ 0.3
Rosenbrock	0.82	0.19	6.23	+ 8.5	299	+ 279	3.6e3	+ 321	1.29	≈ 0.47	2.89	+ 0.41
	0.36	0.21	2.19	+ 2.2	89.6	+ 68.9	3.9e3	+ 4.09	0.67	+ 0.28	2.17	+ 0.5
	0.25	0.11	4.31	+ 12.9	71.0	+ 63.6	3.8e3	+ 4.23	0.39	+ 0.21	1.41	+ 0.39
	0.15	0.13	4.39	≈ 10.8	30.1	+ 12.1	3.7e3	+ 3.23	0.21	≈ 0.23	1.07	+ 0.49
	0.09	0.12	2.98	≈ 8.93	20.9	+ 15.3	3.9e3	+ 4.06	0.14	≈ 0.17	0.56	+ 0.23
	0.07	0.11	3.09	≈ 4.81	19.7	+ 17.3	3.8e3	+ 3.89	0.08	≈ 0.10	0.41	+ 0.41
	0.04	0.04	48.3	≈ 101	18.1	+ 16.4	3.8e3	+ 4.01	0.04	≈ 0.05	0.12	≈ 0.10
	0.10	0.05	45.6	≈ 129	31.5	+ 28.2	3.8e3	+ 4.30	0.09	≈ 0.07	0.06	≈ 0.19
	0.10	0.04	945	+ 1.4e3	68.4	+ 43.1	3.8e3	+ 3.91	0.12	≈ 0.06	0.17	≈ 0.19
	0.17	0.05	947	+ 1.5e3	209	+ 57.9	3.9e3	+ 3.39	0.39	≈ 0.32	0.21	≈ 0.11
Branin	0.69	0.21	3.09	+ 3.08	29.7	+ 21.5	97	+ 38.1	0.72	≈ 0.36	1.41	≈ 0.91
	1.61	2.29	8.09	+ 5.87	19.1	+ 6.01	89.0	+ 64.6	1.00	≈ 0.73	4.25	+ 3.71
	1.76	3.84	12.9	+ 5.41	19.2	+ 3.99	50.9	+ 57.4	4.09	+ 4.87	10.8	+ 6.70
	1.71	2.32	7.07	+ 6.41	15.2	+ 4.43	16.1	+ 14.3	6.87	+ 7.00	11.2	+ 4.81
	0.87	0.43	3.28	≈ 4.39	21.2	+ 20.0	19.8	+ 12.1	2.46	+ 2.47	3.78	+ 2.49
	0.89	0.35	13.2	+ 22.1	82.2	+ 50.1	21.1	+ 22.3	0.86	≈ 0.27	1.15	≈ 0.91
	0.79	0.23	14.3	+ 14.7	57.1	+ 44.1	24.6	+ 29.1	1.49	+ 0.38	4.01	+ 4.43
	0.68	0.36	34.6	+ 19.6	103	+ 69.9	56.1	+ 44.2	2.98	+ 4.37	12.2	+ 11.2
	3.03	4.56	13.8	+ 14.1	34.5	+ 38.1	32.2	+ 21.2	5.02	≈ 7.93	11.9	+ 4.76
	27.7	11.4	64.7	+ 65.0	69.7	+ 49.6	45.1	≈ 44.7	14.7	– 11.7	45.7	≈ 36.3
Harmann3	−4.21	0.21	−3.21	+ 0.70	−1.72	+ 0.63	−2.31	+ 0.87	−4.01	≈ 0.14	−3.78	≈ 0.31
	−4.19	0.31	−3.38	+ 0.81	−2.21	+ 0.89	−2.31	+ 1.04	−4.25	≈ 0.19	−4.13	≈ 0.21
	−3.64	0.24	−3.00	+ 0.67	−1.41	+ 0.80	−2.41	+ 0.65	−3.44	≈ 0.34	−3.34	≈ 0.41
	−3.79	0.13	−2.95	+ 1.1	−2.41	+ 0.84	−1.91	+ 1.02	−3.78	≈ 0.17	−3.65	+ 0.21
	−3.47	0.19	−2.86	≈ 1.91	−1.82	+ 0.72	−1.54	+ 0.94	−3.52	≈ 0.14	−3.52	≈ 0.22
	−4.78	0.28	−4.54	≈ 0.74	−3.36	+ 1.01	−1.75	+ 0.94	−4.92	≈ 0.14	−4.79	≈ 0.37
	−3.78	0.23	−3.26	+ 0.67	−2.21	+ 0.36	−1.79	+ 1.22	−3.84	≈ 0.21	−3.57	≈ 0.38
	−4.54	0.40	−3.51	+ 0.96	−2.84	+ 0.67	−1.44	+ 1.28	−4.55	≈ 0.30	−4.51	≈ 0.23
	−3.92	0.16	−2.93	+ 0.65	−2.16	+ 0.92	−0.25	+ 0.11	−3.85	≈ 0.21	−3.84	≈ 0.24
	+/≈/−			20/20/0		30/9/1		29/11/0		5/32/3		14/26/0

Table 2 Statistical results obtained by PBO-EA, BO, PBO-SQR, PBO-LCB, PBO-AEI, and PBO-PEI on benchmark problems with *noisesevel* = 0.05

Problem	PBO-EA		BO		PBO-SQR		PBO-LCB		PBO-AEI		PBO-PEI		
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	
Ackley	16.5	0.25	16.4	≈ 0.35	15.0	≈ 3.83	16.4	≈ 0.41	16.5	≈ 0.43	16.5	≈ 0.29	
	8.37	0.26	8.30	≈ 0.29	8.38	≈ 0.66	8.42	≈ 0.28	8.45	≈ 0.33	8.52	≈ 0.21	
	13.2	0.24	13.2	≈ 0.32	12.4	– 0.95	13.3	≈ 0.38	13.2	≈ 0.39	13.6	+ 0.29	
	6.14	0.26	6.14	≈ 0.23	6.16	≈ 0.21	5.89	≈ 0.16	6.26	≈ 0.14	6.24	≈ 0.18	
	12.3	0.32	12.3	≈ 0.22	12.6	≈ 0.31	12.3	≈ 0.26	12.4	≈ 0.18	12.3	≈ 0.38	
	12.6	0.32	12.4	≈ 0.29	12.6	≈ 0.33	12.5	≈ 0.25	12.6	≈ 0.18	12.6	≈ 0.13	
	9.41	0.27	9.52	≈ 0.22	9.30	≈ 0.54	9.45	≈ 0.12	9.32	≈ 0.32	9.58	+ 0.23	
	16.7	0.37	16.7	≈ 0.22	16.5	≈ 0.61	16.3	– 0.32	16.6	≈ 0.27	16.5	≈ 0.29	
	3.98	0.18	3.93	≈ 0.16	3.90	≈ 0.45	4.03	≈ 0.16	3.99	≈ 0.21	4.02	≈ 0.23	
	14.4	0.35	14.3	≈ 0.36	4.65	≈ 0.22	14.1	≈ 0.22	14.3	≈ 0.22	14.2	≈ 0.33	
Rosenbrock	1.06	0.27	1.65	≈ 1.78	356	+ 476	3.4e3	+ 1.0e3	1.75	+ 0.29	3.17	+ 0.41	
	0.43	0.17	3.35	+ 9.50	53.4	+ 67.6	3.5e3	+ 787	0.91	+ 0.21	2.06	+ 0.35	
	0.24	0.14	0.61	≈ 2.05	99.5	+ 123	3.6e3	+ 687	0.47	+ 0.12	1.54	+ 0.32	
	0.14	0.11	0.53	+ 1.42	42.8	+ 47.1	3.5e3	+ 9.1e2	0.26	+ 0.08	1.04	+ 0.24	
	0.09	0.06	0.08	≈ 0.14	20.2	+ 9.31	3.8e3	+ 4.01	0.11	≈ 0.06	0.68	+ 0.26	
	0.05	0.05	0.64	+ 1.83	29.4	+ 40.7	3.9e3	+ 4.10	0.06	≈ 0.08	0.36	+ 0.16	
	0.06	0.08	0.62	≈ 1.06	68.1	+ 113	3.8e3	+ 4.01	0.03	≈ 0.05	0.12	≈ 0.04	
	0.06	0.05	1.69	≈ 1.86	89.9	+ 173	3.9e3	+ 4.01	0.08	≈ 0.05	0.06	≈ 0.07	
	0.20	0.19	6.23	+ 10.3	250	+ 198	3.8e3	+ 4.11	0.15	≈ 0.04	0.19	≈ 0.17	
	0.24	0.21	7.74	+ 14.3	411	+ 191	3.8e3	+ 4.02	0.25	≈ 0.11	0.76	≈ 1.23	
Branin	0.61	0.21	1.26	≈ 0.82	23.6	+ 12.3	8.40	+ 4.32	0.88	≈ 0.51	1.78	+ 1.22	
	0.63	0.22	7.08	+ 5.71	21.2	+ 6.39	4.74	+ 5.11	3.21	≈ 3.26	2.98	+ 4.23	
	2.88	4.46	12.1	+ 6.04	14.9	+ 4.91	36.4	+ 22.8	8.09	+ 6.75	6.01	+ 4.41	
	3.37	4.68	6.19	≈ 6.33	15.6	+ 10.1	63.6	+ 43.6	9.72	+ 6.39	13.4	+ 5.40	
	2.81	4.20	4.24	≈ 4.98	41.0	+ 26.5	85.4	+ 93.4	4.38	+ 3.37	4.25	+ 2.31	
	1.20	0.81	9.00	≈ 13.1	49.0	+ 44.0	175	+ 117	0.87	≈ 0.25	1.56	≈ 1.58	
	0.95	0.38	9.01	+ 7.04	49.5	+ 33.1	66.3	+ 55.7	2.21	+ 1.29	2.70	≈ 5.21	
	1.09	1.35	29.6	+ 17.5	159	+ 91.8	247	+ 125	9.47	≈ 9.86	7.34	+ 9.42	
	3.08	4.63	7.44	+ 5.55	84	+ 79.5	51.0	+ 61.6	13.3	+ 12.8	13.4	+ 6.51	
	28.2	9.70	46.5	+ 33.1	151	+ 149	168	+ 159	40.7	≈ 28.3	48.6	≈ 27.6	
Hartmann3	-3.86	0.35	-3.42	+ 0.35	-2.15	+ 0.90	-2.49	+ 0.84	-4.01	≈ 0.17	-3.93	≈ 0.35	
	-4.15	0.30	-3.60	+ 0.25	-2.09	+ 0.64	-2.28	+ 1.14	-4.13	≈ 0.28	-4.16	≈ 0.22	
	-3.69	0.09	-3.12	≈ 0.41	-2.02	+ 0.90	-2.07	+ 0.83	-3.47	≈ 0.38	-3.25	+ 0.50	
	-3.88	0.12	-3.15	≈ 0.52	-1.95	+ 1.16	-2.12	+ 0.92	-3.73	≈ 0.24	-3.81	≈ 0.17	
	-3.61	0.21	-2.77	+ 0.93	-2.12	+ 0.85	-1.87	+ 1.04	-3.61	≈ 0.17	-3.49	≈ 0.33	
	-4.80	0.23	-3.64	+ 1.24	-2.65	+ 0.97	-2.01	+ 1.31	-5.10	– 0.12	-4.87	≈ 0.18	
	-3.86	0.41	-3.10	≈ 1.12	-2.59	+ 0.73	-1.83	+ 1.06	-3.86	≈ 0.27	-3.75	≈ 0.29	
	-4.63	0.23	-3.35	+ 1.21	-2.87	+ 0.86	-1.31	+ 1.31	-4.48	≈ 0.31	-4.23	≈ 0.42	
	-3.84	0.18	-3.18	≈ 1.21	-2.41	+ 0.84	-0.28	+ 0.06	-3.84	≈ 0.25	-3.79	≈ 0.18	
	+/≈/-			16/24/0			30/9/1		30/9/1			11/28/1	

Table 3 Statistical results obtained by PBO-EA, BO, PBO-SQR, PBO-LCB, PBO-AEI, and PBO-PEI on benchmark problems with *noisesevel* = 0.1

$\sigma_\epsilon^2=0.1$ Problem	PBO-EA		BO		PBO-SQR		PBO-LCB		PBO-AEI		PBO-PEI						
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std					
Ackley	16.4	0.31	13.6	≈	4.65	15.2	≈	0.45	16.3	≈	0.33	15.4	≈	0.34	14.0	≈	1.54
	8.23	0.28	8.19	≈	0.33	8.56	≈	0.30	8.63	≈	0.24	8.64	≈	0.37	7.13	≈	1.58
	13.2	0.25	13.3	≈	0.74	13.2	≈	0.16	13.3	≈	0.30	13.3	≈	0.45	12.1	≈	14.7
	6.27	0.24	6.01	≈	0.33	6.13	≈	0.14	6.12	≈	0.21	6.25	≈	0.19	6.03	≈	16.5
	12.1	0.23	12.2	≈	0.31	12.4	≈	0.28	11.7	≈	0.32	12.6	≈	0.28	26.7	+	6.41
	12.1	0.28	12.6	+	0.28	12.4	≈	0.30	12.4	≈	0.29	12.4	≈	0.33	17.9	+	9.13
	9.28	0.23	9.10	≈	0.51	9.48	≈	0.08	9.55	≈	0.20	9.60	≈	0.15	12.4	+	7.67
	16.4	0.34	16.6	≈	0.43	16.5	≈	0.26	16.6	≈	0.36	16.3	≈	0.28	22.4	+	10.4
	4.01	0.19	3.83	≈	0.67	4.07	≈	0.24	4.01	≈	0.14	4.14	≈	0.19	6.63	+	1.26
	14.5	0.34	14.3	≈	0.62	14.5	≈	0.18	14.1	≈	0.16	14.2	≈	0.19	14.0	≈	89.9
Rosenbrock	1.03	0.41	390	+	469	2.2e3	+	1.2e3	2.14	+	0.41	3.39	+	0.70	4.25	+	3.71
	0.44	0.23	95.4	+	102	1.8e3	+	1.7e3	1.28	+	0.39	2.21	+	0.37	11.8	+	6.72
	0.24	0.15	89.7	+	61.6	1.6e3	+	1.9e3	0.72	+	0.19	1.47	+	0.29	11.5	+	4.83
	0.14	0.10	23.2	+	23.2	1.3e3	+	1.7e3	0.36	+	0.14	0.99	+	0.23	3.36	+	2.49
	0.08	0.06	40.3	+	61.6	1.3e3	+	1.7e3	0.12	≈	0.09	0.81	+	0.38	1.15	≈	0.97
	0.11	0.08	10.0	+	14.1	1.2e3	+	1.7e3	0.07	≈	0.08	0.42	+	0.20	4.25	+	4.95
	0.05	0.05	7.01	+	7.22	1.4e3	+	1.7e3	0.03	≈	0.03	0.14	+	0.11	12.7	+	13.6
	0.07	0.04	83.8	+	112	1.3e3	+	1.7e3	0.09	≈	0.05	0.10	≈	0.14	13.1	+	5.05
	0.16	0.09	381	+	227	1.3e3	+	1.7e3	0.22	≈	0.15	0.14	≈	0.12	47.9	+	37.8
	0.28	0.13	547	+	218	1.3e3	+	914	0.39	≈	0.29	0.79	+	0.71	1.90	+	0.37
Branin	2.23	3.23	24	+	19.71	7.98	+	4.46	1.11	≈	0.53	1.64	≈	1.33	1.40	≈	0.9
	1.87	2.60	15.4	+	5.50	16.5	+	17.7	3.07	≈	3.90	6.10	+	4.02	4.25	+	3.7
	0.80	0.46	17.0	+	7.68	35.3	+	29.4	7.63	+	6.63	12.2	+	5.19	11.8	+	6.77
	2.89	5.23	18.5	+	16.9	81.8	+	68.1	9.15	+	7.20	13.3	+	5.14	11.6	+	4.88
	2.16	4.79	20.6	+	19.0	64.4	+	33.6	2.33	+	1.17	2.84	+	1.21	3.36	+	2.49
	1.07	0.48	100	+	54.5	167	+	72.7	1.05	≈	0.92	1.21	≈	0.71	1.15	≈	0.98
	1.92	2.69	41.8	+	29.9	50.9	+	45.6	4.19	+	6.08	3.60	+	2.56	4.25	+	4.97
	6.43	14.5	104	+	87.4	245	+	97.1	8.42	+	10.4	14.8	+	11.0	11.7	+	11.3
	3.78	4.56	68.2	+	83.1	65.8	+	71.8	9.08	+	7.71	13.9	+	8.74	12.7	+	5.01
	32.1	14.5	169	+	124	157	+	98.9	32.6	≈	15.1	47.1	+	18.2	47.1	≈	37.1
Hartmann3	-3.89	0.24	-2.89	≈	1.01	-2.78	+	0.78	-3.76	≈	0.28	-3.83	≈	0.37	-3.91	≈	0.33
	-4.23	0.28	-2.23	≈	1.01	-1.84	+	0.81	-3.76	≈	0.33	-3.71	≈	0.48	-4.19	≈	0.24
	-3.48	0.32	-1.28	+	0.43	-1.98	+	1.08	-3.48	≈	0.33	-3.49	≈	0.12	-3.29	≈	0.37
	-3.86	0.18	-1.69	+	0.89	-2.15	+	1.09	-3.74	≈	0.24	-3.79	≈	0.39	-3.70	+	0.29
	-3.47	0.31	-1.53	+	0.45	-1.49	+	0.87	-3.59	≈	0.20	-3.37	≈	0.29	-3.56	≈	0.25
	-4.78	0.33	-3.24	+	0.53	-2.21	+	1.34	-4.65	≈	0.54	-4.89	≈	0.43	-4.80	≈	0.38
	-3.82	0.23	-2.61	≈	0.80	-1.56	+	1.13	-3.82	≈	0.33	-3.85	≈	0.21	-3.71	≈	0.39
	-4.59	0.15	-2.89	≈	0.74	-1.75	+	1.39	-4.43	≈	0.46	-4.61	≈	0.16	-4.60	≈	0.26
-3.82	0.26	-2.12	≈	0.75	-0.27	+	0.13	-3.82	≈	0.24	-3.66	≈	0.26	-3.84	≈	0.28	
1/0/-1				27/13/0			30/10/0				10/30/0			16/24/0			22/18/0

Table 4 Statistical results obtained by PBO-EA, BO, PBO-SQR, PBO-LCB, PBO-AEI, and PBO-PEI on benchmark problems with *noisesevel* = 0.5

Problem	PBO-EA		BO		PBO-SQR		PBO-LCB		PBO-AEI		PBO-PEI						
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std					
Ackley	16.6	0.31	16.6	≈	0.53	10.4	–	6.50	16.7	≈	0.36	16.7	≈	0.30	16.5	≈	0.32
	8.35	0.23	8.54	≈	0.19	7.37	≈	2.87	8.54	≈	0.22	8.54	+	0.23	8.63	≈	0.23
	13.2	0.28	13.1	≈	0.22	12.6	≈	1.21	13.5	≈	0.21	13.4	≈	0.34	13.3	≈	0.33
	6.28	0.19	6.29	≈	0.18	6.11	≈	0.33	6.17	≈	0.23	6.33	≈	0.17	6.27	≈	0.19
	12.1	0.23	12.4	≈	0.22	12.6	≈	0.23	12.5	≈	0.25	12.6	≈	0.20	12.5	≈	0.23
	12.5	0.21	12.3	≈	0.22	12.6	≈	0.41	12.5	≈	0.32	12.7	≈	0.30	14.3	+	0.23
	9.43	0.23	9.62	≈	0.24	9.53	≈	0.59	9.58	≈	0.22	9.51	≈	0.23	9.63	≈	0.17
	16.2	0.21	16.9	+	0.22	16.3	≈	0.61	16.4	≈	0.42	16.6	≈	0.26	16.2	≈	0.32
	3.87	0.28	3.88	≈	0.27	4.08	≈	0.58	4.02	+	0.18	4.01	≈	0.13	4.02	≈	0.21
	14.6	0.21	14.6	≈	0.19	14.6	≈	0.50	14.6	≈	0.23	14.5	≈	0.27	14.5	≈	0.23
Rosenbrock	56.9	303	5.31	–	6.42	88.2	≈	134	1.2e3	+	642	55.2	≈	103	62.1	≈	124
	9.54	16.9	9.39	≈	9.47	363	≈	894	946	+	986	32.0	+	48.1	141	+	179
	16.0	116	12.6	≈	26.7	214	+	431	343	+	483	15.5	≈	26.7	10.7	≈	10.6
	4.27	73.2	3.81	≈	3.56	13.4	+	13.2	212	+	315	46.1	+	112	6.24	+	10.0
	11.5	30.7	3.73	–	4.30	651	+	999	8.77	–	7.02	9.32	≈	16.5	18.4	+	26.7
	21.7	169	23.1	≈	56.4	997	+	998	63.0	+	95.8	45.6	+	72.7	74.1	+	121
	2.72	2.65	12.1	+	16.2	63.2	+	121	587	+	767	123	+	203	7.43	+	19.1
	23.9	531	30.0	≈	67.2	9.19	–	10.1	812	+	756	54.6	≈	145	95.2	+	234
	20.1	19.9	12.4	≈	24.5	18.5	≈	27.6	2.3e3	+	1.2e3	51.2	+	98.9	298	+	389
	256	387	305	+	876	201	≈	564	2.0e3	+	1.0e3	795	+	1.1e3	402	+	813
Branin	7.21	5.56	32.3	+	53.1	59.6	+	67.3	34.2	+	37.2	5.10	≈	4.34	11.1	+	10.5
	12.1	24.2	11.8	≈	12.5	39.9	+	57.1	58.4	+	63.4	23.6	+	20.4	21.7	≈	20.4
	22.6	21.0	10.9	–	14.4	80.3	+	116	87.7	+	67.8	32.5	+	18.5	27.7	+	19.6
	22.1	23.2	5.87	–	3.87	33.8	+	19.0	141	+	97.8	36.7	≈	38.00	17.5	≈	12.2
	21.4	20.9	15.4	≈	18.9	25.2	≈	17.4	153	+	96.9	20.0	≈	36.8	27.3	≈	48.4
	29.8	28.4	26.3	≈	32.3	87.0	+	130	235	+	164	15.6	–	24.1	56.9	+	90.2
	8.91	9.00	24.8	+	41.7	54.3	+	55.8	75.4	+	62.0	35.1	+	53.7	32.9	+	57.3
	12.6	10.2	8.79	≈	8.28	105	+	141	214	+	118	73.5	+	81.2	32.0	+	23.0
	35.3	29.1	19.2	–	24.0	55.6	+	55.2	104	+	72.92	16.9	–	16.0	32.1	≈	59.1
	38.6	31.1	16.3	–	19.4	19.6	–	13.1	119	+	44.1	65.0	+	50.4	45.0	+	28.6
Harmann3	–3.19	0.67	–1.53	+	0.96	–1.97	+	0.55	–2.11	+	0.81	–2.98	≈	0.52	–3.11	≈	0.68
	–3.74	0.75	–1.63	+	1.14	–1.44	+	0.49	–2.24	+	1.28	–3.37	≈	0.65	–3.69	≈	0.69
	–3.22	0.31	–1.51	+	1.12	–1.56	+	0.27	–2.03	+	1.01	–3.03	≈	0.66	–3.23	≈	0.23
	–3.21	0.58	–2.37	+	1.18	–1.57	+	0.43	–1.99	+	0.73	–3.57	≈	0.41	–3.12	≈	0.64
	–2.99	0.42	–2.19	+	1.13	–1.29	+	0.55	–1.89	+	0.87	–3.36	≈	0.33	–2.86	≈	0.49
	–3.45	1.23	–3.17	≈	1.55	–2.34	+	0.56	–2.39	+	1.18	–4.12	≈	0.50	–3.80	≈	0.81
	–3.43	0.76	–2.35	+	1.01	–1.69	+	0.41	–1.54	+	0.91	–3.81	≈	0.37	–2.85	+	0.73
	–3.89	0.81	–2.65	+	1.49	–1.82	+	0.48	–2.31	+	1.40	–4.34	≈	0.61	–4.23	≈	0.77
	–2.80	0.92	–2.43	≈	1.26	–1.87	+	0.44	–1.09	+	0.75	–3.13	≈	0.73	–3.19	≈	1.03
	–2.45	1.12	–2.31	≈	1.41	–2.04	≈	0.45	–1.22	+	1.29	–3.43	–		–3.69	–	0.72
1/0/–1					12/22/6			22/15/3			30/9/1			12/26/2			16/23/1

Table 5 Statistical results obtained by PBO-EA, BO, PBO-SQR, PBO-LCB, PBO-AEI, and PBO-PEI on benchmark problems with *noiselevel* = 1

Problem	PBO-EA		BO		PBO-SQR		PBO-LCB		PBO-AEI		PBO-PEI		
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	
Ackley	14.8	0.31	16.1	≈ 0.34	15.4	≈ 2.51	12.3	≈ 0.47	16.5	≈ 0.52	16.2	≈ 0.29	
	8.32	0.18	8.36	≈ 0.24	7.07	≈ 2.21	8.49	≈ 0.28	8.62	≈ 0.24	8.45	≈ 0.21	
	12.1	0.27	13.4	≈ 0.29	13.4	≈ 1.39	13.4	≈ 0.22	13.7	≈ 0.25	13.43	≈ 0.42	
	6.07	0.28	6.01	≈ 0.21	5.21	≈ 1.32	6.11	≈ 0.20	6.28	≈ 0.26	6.23	≈ 0.11	
	11.2	0.23	12.2	≈ 0.42	9.24	– 3.12	12.7	≈ 0.28	12.9	≈ 0.32	12.6	≈ 0.28	
	12.1	0.34	13.1	≈ 0.29	9.87	– 2.31	12.3	≈ 0.22	12.7	≈ 0.31	12.7	≈ 0.19	
	9.43	0.12	9.52	≈ 0.12	7.93	≈ 1.92	9.69	≈ 0.15	9.50	≈ 0.18	9.54	≈ 0.37	
	16.5	0.29	16.5	≈ 0.56	13.9	≈ 4.81	16.6	≈ 0.25	16.6	≈ 0.28	16.6	≈ 0.16	
	4.15	0.20	4.10	≈ 0.15	3.22	≈ 1.03	4.05	≈ 0.20	3.99	≈ 0.27	4.10	≈ 0.26	
	14.5	0.23	14.5	≈ 0.35	12.6	≈ 3.15	14.5	≈ 0.25	14.4	≈ 0.22	14.5	≈ 0.24	
	Rosenbrock	116	96.1	956	+ 768	967	+ 786	1.2e3	+ 865	701	+ 568	445	+ 498
		193	278	181	≈ 176	1.3e3	+ 1.1e3	854	+ 753	488	≈ 832	490	≈ 293
		121	109	334	+ 423	821	+ 1.2e3	542	+ 454	434	+ 601	205	≈ 184
		119	109	131	≈ 176	1.2e3	+ 1.5e3	323	+ 276	212	+ 123	124	≈ 123
	123	112	189	+ 214	1.2e3	+ 1.3e3	476	+ 346	287	+ 251	119	≈ 125	
	143	287	149	≈ 207	423	+ 687	621	+ 625	276	+ 213	241	+ 179	
	117	187	178	+ 256	397	+ 798	897	+ 798	214	+ 423	345	+ 313	
	176	512	134	≈ 176	578	+ 821	1.4e3	+ 821	501	+ 423	156	≈ 157	
	556	780	512	≈ 1.1e3	387	– 865	2.5e3	+ 787	921	+ 723	1.0e3	+ 613	
	612	761	474	– 1.0e3	1.0e3	+ 1.1e3	2.1e3	+ 1.0e3	1.0e3	+ 765	1.1e3	+ 731	
Branin	31.9	19.6	81.1	+ 61.7	90.4	+ 59.5	75.1	+ 54.4	48.1	≈ 33.4	28.1	≈ 21.8	
	36.3	34.4	109	+ 123	131	+ 91.1	121	+ 63.3	31.2	≈ 13.4	34.4	≈ 17.4	
	37.1	24.8	93.0	+ 75.4	92.4	+ 87.1	183	+ 67.12	32.2	≈ 17.9	34.8	≈ 28.9	
	73.0	95.1	129	≈ 126	190	+ 156	221	+ 102	51.5	≈ 27.6	43.5	– 42.7	
	82.3	91.3	126	≈ 81.6	134	+ 137	212	+ 110	87.2	≈ 77.0	50.8	≈ 54.9	
	102	95.3	71.4	≈ 81.4	180	+ 188	328	+ 126	88.3	≈ 70.5	86.7	≈ 53.0	
	68.1	45.1	52.8	≈ 33.1	96.9	+ 91.4	126	+ 58.5	68.8	≈ 64.5	81.1	+ 44.92	
	102	82	76.0	– 58.3	141	+ 154	283	+ 95.1	87.5	≈ 73.2	135	+ 98.3	
	45.4	36.4	31.2	≈ 20.3	84.7	+ 82.2	132	+ 70.5	80.4	+ 72.1	91.0	+ 70.3	
	91	88	93.2	≈ 70.3	193	+ 128	250	+ 139	90.3	≈ 53.6	146	+ 115	
Hartmann3	–3.23	0.56	–1.25	+ 1.13	–1.53	+ 0.59	–1.97	+ 0.75	–3.53	≈ 0.68	–2.63	+ 0.84	
	–3.61	0.95	–1.36	+ 1.44	–1.73	+ 0.45	–1.95	+ 1.00	–3.60	≈ 0.52	–2.81	+ 0.62	
	–2.41	0.92	–1.15	+ 1.22	–1.28	+ 0.61	–2.19	≈ 0.95	–2.99	≈ 0.45	–2.42	≈ 0.87	
	–2.97	0.76	–1.17	+ 1.14	–1.32	+ 0.55	–2.24	≈ 1.21	–3.01	≈ 0.79	–2.84	≈ 0.76	
	–2.99	0.68	–0.89	+ 0.76	–1.32	+ 0.91	–1.36	+ 1.09	–2.65	≈ 0.72	–2.34	≈ 0.74	
	–4.56	1.32	–1.23	+ 1.16	–2.31	+ 0.78	–2.32	+ 1.28	–3.44	≈ 1.08	–3.37	≈ 1.29	
	–2.59	0.84	–1.15	+ 0.99	–1.79	≈ 0.63	–2.27	≈ 1.31	–2.68	≈ 0.81	–2.79	≈ 0.83	
	–3.85	1.00	–1.28	+ 0.99	–2.40	+ 0.66	–2.61	+ 1.32	–3.70	≈ 0.66	–3.77	≈ 0.85	
	–2.77	0.99	–1.26	+ 1.02	–1.54	+ 0.47	–1.38	+ 1.05	–3.03	≈ 0.92	–2.68	≈ 0.70	
	–2.77	1.13	–1.19	+ 0.94	–2.02	≈ 0.32	–2.02	≈ 1.56	–2.63	≈ 0.81	–3.19	≈ 0.81	
+ / ≈ / –			17/21/2		27/10/3		26/14/0		10/30/0		11/28/1		

benefits of the use of personalized information for modelling the objective function. However, the performance of both BO and PBO-EA will degrade with the increase of the noise level.

2. Comparison of optimization methods: We can see that PBO-EA always shows better performance than PBO-SQR, indicating the effectiveness of EAs for optimizing the acquisition function. A possible explanation is that

EAs are population-based optimization methods and do not require the gradient of the objective function, allowing them to get out of local optima. However, it is interesting to see that PBO-SQR shows better performance than PBO-EA on Ackley function. The reason may be that SQR methods are suitable for solving Ackley function, because Ackley function is characterized by a nearly flat outer region, and a large hole at the centre, which follows the assumption of SQR method.

3. Comparison of the acquisition functions: We tested four acquisition functions within the PBO framework, i.e., EI, LCB, AEI, and PEI. According to the results in Tables 1, 2, 3, 4, 5 and Tables SI–SV, we can observe that the EI shows better performance than the others on problems with different levels of noise. Among all the tested acquisition functions, LCB shows the worst performance on most test instances and the level of noise significantly degrades its performance. It is interesting to see that when the noise level is large, i.e., $\sigma_\varepsilon^2 = 1$, EI, PEI, and AEI show similar performance. A possible explanation is that AEI involves a penalty to account for the noise variance of the next evaluation. Specially, AEI penalizes data points with a small prediction variance and therefore enhances exploration. The small noise level may impact the surrogate model slightly, and the enhanced exploration may hinder slow convergence, which is not desired when a very limited number of fitness evaluations are available. By contrast, when heavy noise may lead to poor surrogate models, both PEI, AEI, and EI cannot evaluate the candidate solutions correctly, resulting in similar performance.
4. Comparison on MOPs with personalized variables: According to the statistical results in terms of the mean and std HV values in Table 6, it is clear that PGP-MOEA significantly outperforms GP-MOEA on 38 out of 54 optimization tasks resulting from varying values of the personalized variable. Moreover, they show similar performance with regard to convergence and diversity indicated by HV values on the remaining ones. This observation indicates that personalized GPs can provide better predictions than the standard GPs in the considered problem, which further enhances the optimization performance of PGP-MOEA. This confirms the effectiveness of the personalized GP for handling optimization problems where personalization is taken into consideration.

To gain a deeper insight into the performance achieved by each algorithm, we plot the change of the mean and the variance of the best objective values over the number of contexts (the value of the personalized variable). As shown in Fig. 1, PBO-EA can achieve better performance on the selected test instances for different personalized values. Specifically,

while the best value found by the standard BO on different personalized values varies a lot, PBO can achieve a more stable performance with the change of the personalized variable, indicating the benefits of PGPs. Compared with the SQR method, the EA shows more stable and better results, especially for Beale function and Rosenbrock function. This reveals the advantage of EAs for optimization of the acquisition function that are multi-model and complex. By comparing the different acquisition functions, we can see that the EI acquisition function shows promising performance compared with the other acquisition functions. Interestingly, both AEI and PEI achieve competitive performance on all the test instances, especially when the noise level increases. Similarly, we plot the non-dominated solution set with the median HV value among 20 runs obtained by PGP-MOEA and GP-MOEA on DTLZ2 and DTLZ1a, respectively. As shown in Fig. 2 and Fig. S1, we can see that PGP-MOEA can achieve a better performance in terms of convergence and diversity.

Conclusion

In this paper, we consider solving expensive optimization problems with personalized decision variables and observation noise, which cannot be efficiently solved by standard Bayesian optimization methods. We consider both single-objective and multi-objective optimization, and introduce the corresponding test benchmark problems. Then, we propose a personalized Bayesian optimization algorithm by jointly learning surrogate models over different contexts and accounting for the observation noise. More specifically, a composite kernel is introduced to measure the similarity of decision variables and personalized variables simultaneously, making it possible to transfer knowledge between different individuals. To reduce the impact of noise on the optimization, we test different kinds of acquisition functions and it turns out that EI always achieves competitive performance with different noise levels.

The proposed algorithm is tested on sets of widely used benchmark problems for different personalized information. Our experimental results demonstrate that the proposed algorithm achieves significantly better performance than the standard Bayesian optimization methods on most test instances studied in this work. Comparisons are also carried out to investigate the effectiveness of the contextual Gaussian process and acquisition functions used in the proposed algorithm. The empirical results confirm that the Gaussian processes considering the personalized information result in the good performance of the proposed algorithm.

The research on optimization problems involving personalized/contextual information is still in its infancy and demands for further investigations. On the one hand, it is chal-

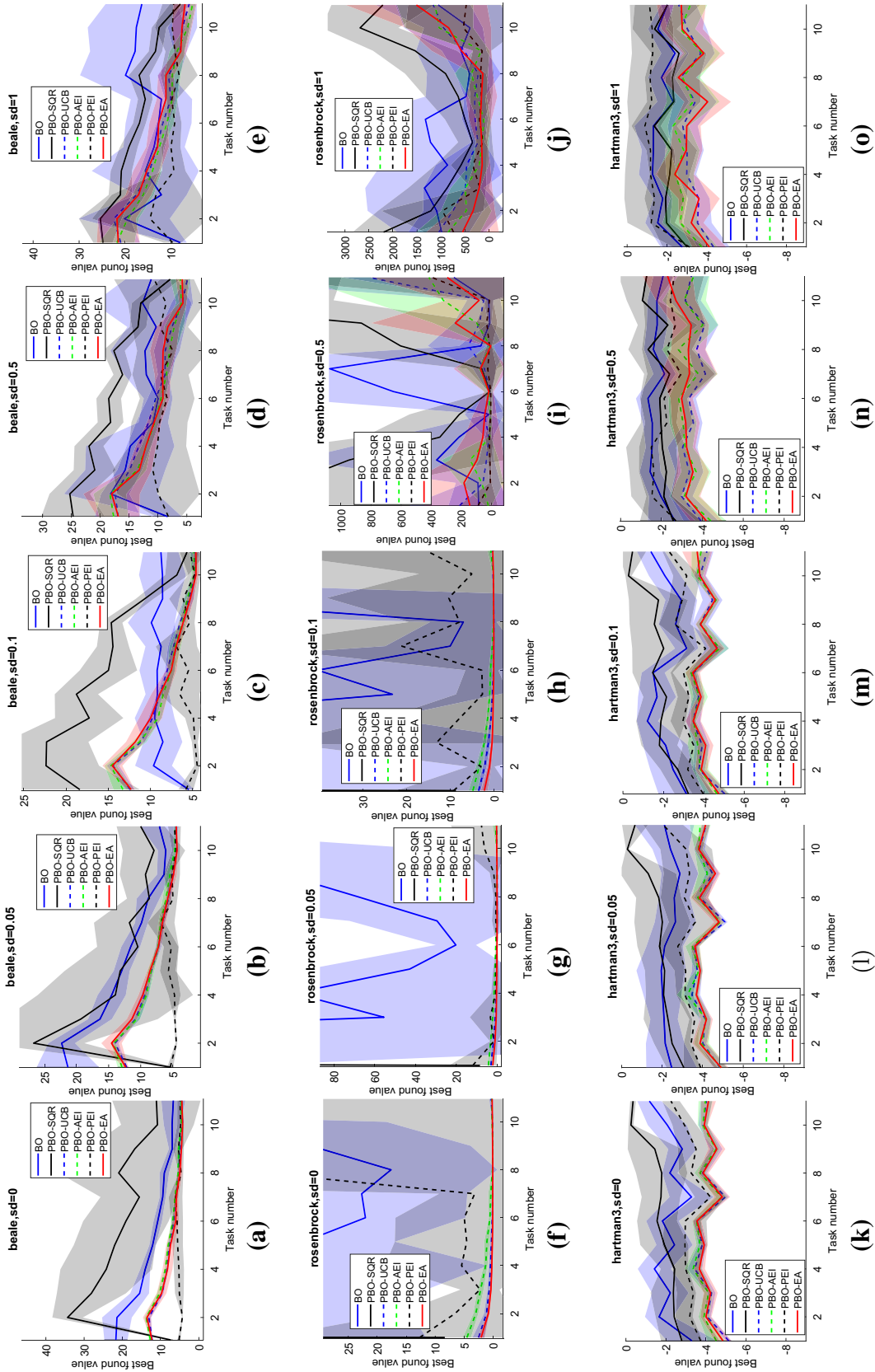


Fig. 1 The mean and variance of the optima obtained by PBO-EA, BO, PBO-SQR, PBO-LCB, PBO-AEI, and PBO-PEI for some benchmark problems with different noise levels $\sigma_\epsilon^2 = 0, 0.05, 0.1, 0.5, 1$

Table 6 Statistical results obtained by PGP-MOEA and GP-MOEA with the same number of real function evaluations

Algorithm	Problems	DTLZ1						DTLZ2						DTLZ3					
		p_1	p_2	p_3	p_4	p_5	p_6	p_1	p_2	p_3	p_4	p_5	p_6	p_1	p_2	p_3	p_4	p_5	p_6
PGP-MOEA	Mean	0.54	0.42	0.45	0.45	0.46	0.43	0.63	0.38	0.35	0.30	0.23	0.25	0.49	0.43	0.42	0.41	0.42	0.47
	Std	0.13	0.29	0.29	0.24	0.27	0.28	0.06	0.25	0.23	0.22	0.21	0.22	0.09	0.22	0.14	0.15	0.17	0.20
	Mean	+	+	+	+	+	+	+	+	+	+	≈	+	+	+	+	+	+	+
	Std	0.08	0.20	0.25	0.20	0.23	0.22	0.06	0.05	0.04	0.08	0.07	0.07	0.10	0.10	0.16	0.16	0.17	0.13
GP-MOEA	Mean	0.41	0.40	0.40	0.37	0.38	0.38	0.56	0.59	0.57	0.55	0.58	0.57	0.19	0.19	0.18	0.19	0.21	0.23
	Std	0.14	0.16	0.17	0.15	0.17	0.17	0.12	0.10	0.11	0.10	0.08	0.10	0.03	0.03	0.03	0.03	0.05	0.03
	Mean	≈	≈	≈	≈	≈	≈	+	+	+	+	+	+	+	+	+	+	+	+
	Std	0.11	0.13	0.12	0.12	0.13	0.13	0.15	0.13	0.10	0.17	0.08	0.08	0.04	0.02	0.02	0.03	0.03	0.03
PGP-MOEA	Mean	0.02	0.01	0.02	0.02	0.02	0.02	0.51	0.44	0.41	0.56	0.51	0.47	0.46	0.32	0.38	0.41	0.40	0.38
	Std	0.01	0.00	0.01	0.01	0.01	0.01	0.33	0.28	0.27	0.30	0.24	0.35	0.29	0.31	0.35	0.33	0.30	
	Mean	≈	≈	≈	≈	≈	≈	+	+	≈	+	+	+	≈	+	+	+	+	
	Std	0.00	0.00	0.00	0.01	0.00	0.01	0.19	0.24	0.22	0.28	0.26	0.26	0.15	0.10	0.13	0.13	0.12	0.09

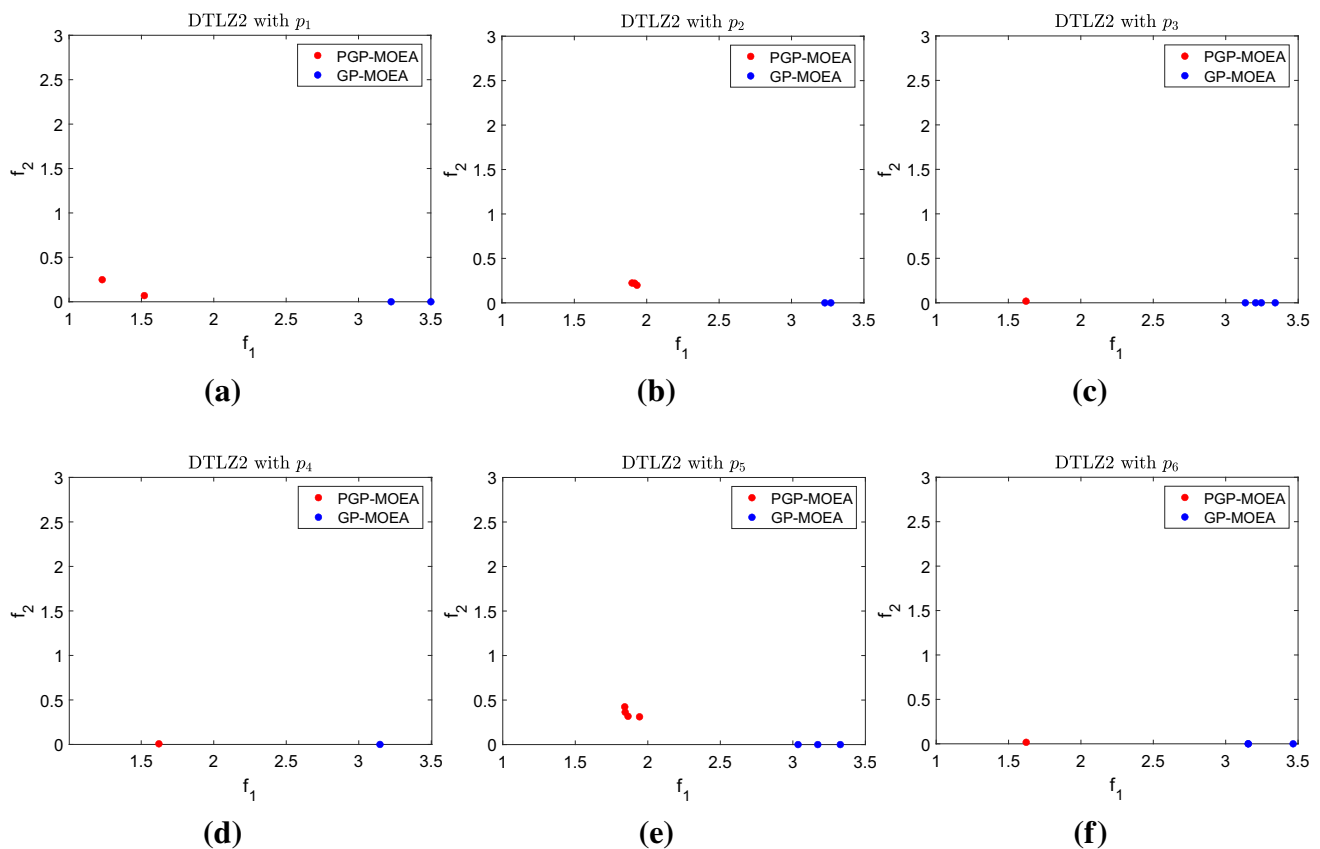


Fig. 2 The optimal solutions associated with median HV values obtained by PGP-MOEA and GP-MOEA on DTLZ2 for six different personalized values

lenging to construct effective surrogate models over various contexts, as only a very limited number of data are available in each context. On the other hand, for a given value for the personalized variable, the search of new samples to be evaluated by the true expensive objective functions should be further investigated, since optimization of the acquisition function will become increasingly challenging. Moreover, the assumptions that optimization problems are noise-free may hardly hold in practice, rendering poor optimization performance. Hence, it is interesting yet challenging to investigate noise-handling methods in Bayesian optimization of real-world problems.

Acknowledgements This work was supported by an Alexander von Humboldt Professorship for Artificial Intelligence endowed to Y. Jin by the German Federal Ministry of Education and Research.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest All authors declare that there are no financial and personal relationships with other people or organizations that could inappropriately influence (bias) our work.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Antal A, Paulus W (2013) Transcranial alternating current stimulation (tACS). *Front Hum Neurosci* 7:317
2. Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 20(5):773–791
3. Dipankar D, Zbigniew M (2013) *Evolutionary algorithms in engineering applications*. Springer Science & Business Media
4. Deb K, Beyer H-G (2001) Self-adaptive genetic algorithms with simulated binary crossover. *Evol Comput* 9(2):197–221

5. Deb Kalyanmoy, Deb Debayan et al (2014) Analysing mutation schemes for real-parameter genetic algorithms. *Int J Artif Intell Soft Comput* 4(1):1–28
6. Deb Kalyanmoy, Thiele Lothar, Laumanns Marco, Zitzler Eckart (2002) Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 1, pages 825–830. IEEE
7. Robert Dürichen, Pimentel Marco AF, Lei Clifton, Achim Schweikard, Clifton David A (2014) Multitask Gaussian processes for multivariate physiological time-series analysis. *IEEE Trans Biomed Eng* 62(1):314–322
8. Emmerich Michael TM, Giannakoglou Kyriakos C, Boris Naujoks (2006) Single-and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Trans Evol Comput* 10(4):421–439
9. Ginsbourger David, Baccou Jean, Chevalier Clément, Perales Frédéric, Garland Nicolas, Monerie Yann (2014) Bayesian adaptive reconstruction of profile optima and optimizers. *SIAM/ASA J Uncertainty Quantification* 2(1):490–510
10. Deng Huang, Allen Theodore T, Notz William I, Ning Zeng (2006) Global optimization of stochastic black-box systems via sequential Kriging meta-models. *J Global Opt* 34(3):441–466
11. Jones Donald R, Matthias Schonlau, Welch William J (1998) Efficient global optimization of expensive black-box functions. *J Global Opt* 13(4):455–492
12. Krause Andreas, Ong Cheng Soon (2011) Contextual Gaussian process bandit optimization. In *Nips*, pages 2447–2455
13. McKay Michael D, Beckman Richard J, Conover William J (2000) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 42(1):55–61
14. Moćkus Jonas (1975) On Bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, pages 400–404. Springer
15. Picheny Victor, Wagner Tobias, Ginsbourger David (2013) A benchmark of Kriging-based infill criteria for noisy optimization. *Struct Multidisciplinary Opt* 48(3):607–626
16. Rasmussen Carl Edward (2003) Gaussian processes in machine learning. In *Summer school on Machine Learning*, pages 63–71. Springer
17. Bobak Shahriari, Kevin Swersky, Ziyu Wang, Adams Ryan P, Nando De Freitas (2015) Taking the human out of the loop: a review of Bayesian optimization. *Proc IEEE* 104(1):148–175
18. Snoek Jasper, Larochelle Hugo, Adams Ryan P (2012) Practical Bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25
19. Srinivas Niranjana, Krause Andreas, Kakade Sham M, Seeger Matthias (2010) Gaussian process optimization in the bandit setting: No regret and experimental design. pages 1015–1022
20. Swersky Kevin, Snoek Jasper, Adams Ryan P (2013) Multi-task Bayesian optimization. *Advances in neural information processing systems*, 26
21. Anh Tran, Mike Eldred, Tim Wildey, Scott McCann, Jing Sun, Visintainer Robert J (2022) aphBO-2GP-3B: a budgeted asynchronous parallel multi-acquisition functions for constrained Bayesian optimization on high-performing computing architecture. *Struct Multidisciplinary Opt* 65(4):1–45
22. While Lyndon, Hingston Philip, Barone Luigi, Huband Simon (2006) A faster algorithm for calculating hypervolume. *IEEE Trans Evol Comput* 10(1):29–38
23. Wilcoxon Frank, Katti SK, Wilcox Roberta A (1963) Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test, volume 1. American Cyanamid Pearl River (NY)
24. Williams Chris, Bonilla Edwin V, Chai Kian M (2007) Multi-task Gaussian process prediction. *Advances in Neural Information Processing Systems*, pages 153–160
25. Zhang Qingfu, Liu Wudong, Tsang Edward, Virginas Botond (2009) Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Trans Evol Comput* 14(3):456–474

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.