



HAL
open science

Bayesian Optimisation for Quality Diversity Search With Coupled Descriptor Functions

Paul Kent, Adam Gaier, Jean-Baptiste Mouret, Juergen Branke

► **To cite this version:**

Paul Kent, Adam Gaier, Jean-Baptiste Mouret, Juergen Branke. Bayesian Optimisation for Quality Diversity Search With Coupled Descriptor Functions. IEEE Transactions on Evolutionary Computation, 2024, pp.1-1. 10.1109/TEVC.2024.3376733 . hal-04537563

HAL Id: hal-04537563

<https://inria.hal.science/hal-04537563>

Submitted on 8 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Bayesian Optimisation for Quality Diversity Search with coupled descriptor functions

Paul Kent, Adam Gaier, Jean-Baptiste Mouret and Juergen Branke

Abstract—Quality Diversity (QD) algorithms such as the Multi-Dimensional Archive of Phenotypic Elites (MAP-Elites) are a class of optimisation techniques that attempt to find many high performing points that all behave differently according to a user-defined behavioural metric.

In this paper we propose the Bayesian Optimisation of Elites (BOP-Elites) algorithm. Designed for problems with expensive coupled fitness and behaviour functions, it is able to return a QD solution-set with excellent performance already after a relatively small number of samples. BOP-Elites models both fitness and behavioural descriptors with Gaussian Process surrogate models and uses Bayesian Optimisation strategies for choosing points to evaluate in order to solve the quality-diversity problem. In addition, BOP-Elites produces high quality surrogate models which can be used after convergence to predict solutions with any behaviour in a continuous range.

An empirical comparison shows that BOP-Elites significantly outperforms other state-of-the-art algorithms without the need for problem-specific parameter tuning.

Index Terms—Quality-Diversity, Bayesian Optimisation

I. INTRODUCTION

Optimisation algorithms are ubiquitous in science, engineering, and research and typically attempt to find the best feasible solution for a given objective function. In recent years, a new class of optimisation problems has emerged, called Quality Diversity (QD) optimisation. In QD settings, alongside a solution’s objective performance, solutions exhibit some behavioural characteristics or *descriptor values* which map solutions on to a low dimensional *descriptor space*. These descriptors are selected because they are of interest to a decision maker, providing an informative way to distinguish between solutions. By requiring a search algorithm to return points well distributed in descriptor space, we develop a solution set that is meaningfully diverse and provides insights about the best possible objective performance depending on the point in descriptor space. This process of gaining information on the relationship between a solution’s descriptors and the best achievable objective value is referred to as *illumination* in QD literature [1].

One compelling argument for incorporating QD search in design tasks is that it enables the decision-maker to leverage their expertise when selecting the final solution. This approach makes it possible to consider criteria that are difficult to encode in an objective function, such as aesthetics or manufacturing ease. By using a QD search, the decision-maker can find a set

of high-quality starting designs that exhibit varied characteristics, allowing them to refine the design manually while taking into account their expert knowledge of additional factors that are challenging to quantify.

Interest in QD research has rapidly grown since the introduction of the Multi-Dimensional Archive of Phenotypic Elites (MAP-Elites) algorithm [1] that has wide reaching applications in high dimensional domains and exhibits excellent performance with relatively low computational cost.

Unfortunately, candidate solutions for many real-world problems can only be evaluated a few hundred times. For instance, Jiang et al. [2] described a proof-of-concept that leverages MAP-Elites to search for novel nanoparticles. They used a “robot chemist” to mix liquids autonomously and they evaluate both the features and the fitness of candidate solutions with a spectrometry analysis of the produced nanoparticles: Each of these automated chemistry experiments takes about 12 minutes which forced Jiang et al. to keep the total number of fitness function evaluations below 500 for each experiment (4 days). By contrast, most QD experiments are currently performed with a budget of at least 100,000 evaluations (often more, e.g., [1], [3], [4], [5], [6]). Similarly to chemistry, optimisation problems that involve fluid simulation [7] or real robots [8] are typically too expensive or too slow to use more than a thousand calls to the fitness function. The Surrogate Assisted Illumination (SAIL) algorithm [7] drastically improves sample efficiency by using a Gaussian Process (GP) surrogate model of the objective function. In contrast to MAP-Elites, which provides its final archive of observed points as a solution set, SAIL offers a prediction based on its surrogate model. Surrogate-Assisted Phenotypic Niching (SPHEN) [9] expands upon SAIL by additionally modelling expensive descriptor functions using GPs, thus making it suitable for use with expensive descriptors, or those that can only be evaluated at the same time as an expensive objective function.

In this study, we present BOP-Elites, a Bayesian optimisation (BO) algorithm tailored for Quality-Diversity problems¹. Like SPHEN, BOP-Elites employs GP surrogate models to represent both objective and descriptor landscapes. However, in contrast to prior research, our method applies a Bayesian Optimisation framework by continuously optimising an acquisition function using sequential point selection, which enables a principled and efficient search strategy with unparalleled sample efficiency. Furthermore, BOP-Elites utilises a structured archive for retaining high-performing observations and

Corresponding author Paul Kent. Email: paul.kent@warwick.ac.uk. Paul Kent is with University of Warwick, UK. Adam Gaier is with Autodesk, Germany. Jean-Baptiste Mouret is with Université de Lorraine, CNRS, Inria, France. Juergen Branke is with University of Warwick, UK.

¹A preliminary paper outlining the basic idea for BOP-Elites was released on arXiv in 2020 by the authors in [10]

ultimately delivers a solution set of observed points, akin to the MAP-Elites method.

Inspired by the SAIL methodology, we additionally explore the ‘illumination’ capabilities of BOP-Elites by evaluating the quality of the surrogate models at the end of the run. We further show that surrogate models built by BOP-Elites can be used to ‘upscale’ a solution archive, i.e., use the surrogates trained when building a lower resolution archive to fill a higher resolution archive with high performing predictions.

The key contributions of this paper are as follows:

- The first principled Bayesian Optimisation algorithm for Quality-Diversity Optimisation with expensive coupled objective and descriptor functions².
- An extension of the well known Expected Improvement [11] acquisition function for the QD case and, more generally, to the problem of optimising sub-regions of a search-space modelled with a single Gaussian Process.
- A method for ‘upscaling’ is proposed, using the surrogate models built while solving a QD problem with few regions for approximating a solution to a QD problem with many regions.
- A thorough experimental comparison is provided, showing BOP-Elites performance against the state-of-the-art QD algorithms.

II. PROBLEM DEFINITION

Given a box-constrained, d dimensional search domain $X \subset \mathbb{R}^d$, a real-valued objective function $f(x) : X \rightarrow \mathbb{R}$ and descriptor function $b(x) : X \rightarrow \mathbb{R}^m$. The value of $y = f(x)$ is the performance of the solution while $b(x)$ describes some high level characteristics in an m dimensional descriptor space. While descriptor functions themselves need not be bounded in principle, bounds mark areas of behavioural interest. For each descriptor b_i , human decision makers specify a-priori a set of N_i partitions within the dimension of each descriptor i . By subdividing each descriptor dimension, we form $|R| = \prod_{i=1}^m N_i$ regions in descriptor space with distinct combinations of behavioural qualities.

A simple example would be having two descriptor functions each with two partitions. Suppose we wish to manufacture a part made with a combination plastic polymer, our search domain is the space of different plastics, the performance metric is the life-span of the part under use and the descriptors are observed both during manufacturing and after testing. b_1 is the color of the resultant polymer and has two region labels ‘Mostly Blue’ and ‘Mostly Red’. b_2 measures the electrical conductivity of the part and has two labels ‘Low conductivity’ and ‘High conductivity’. This leads to $|R| = 4$ regions with the interpretable labels R_1 : ‘Mostly Blue, High Conductivity’, R_2 : ‘Mostly Blue, Low Conductivity’, R_3 : ‘Mostly Red, High Conductivity’ or R_4 : ‘Mostly Red, Low conductivity’.

To highlight the distinctions between Quality-Diversity and other optimisation methods, it is essential to recognize that there is no universally ‘better’ color, as preferences vary

depending on the decision-maker. Similarly, a manufacturer may desire a material with either conductive or insulating properties, depending on its intended application. The descriptors here are not treated as objectives to be optimized, but dimensions over which to generate diverse solutions.

In this paper, we explore two types of problems distinguished by the characteristic of the descriptor functions. In the simpler case, the descriptor functions can be evaluated independently of the objective and are assumed cheap to evaluate. We refer to this as a *decoupled* descriptor function. In the more general and more challenging case, the descriptor functions are either themselves expensive and black-box, having no analytical form, or coupled to an expensive objective and may only be observed at the same time. Either of the latter cases require modelling the descriptors, and for simplicity we refer to them both as *coupled* descriptor function case. The goal is to perform QD optimisation [12] in as few function evaluations as possible, that is, to find the set $S^* = \{e_1^*, \dots, e_{|R|}^*\}$ of optimal solutions within each achievable predefined region in descriptor space, with

$$e_r^* = \arg \max_x f(x) \quad s.t. \quad x \in X, \quad b(x) \rightarrow r,$$

where by “ $b \rightarrow r$ ” we mean that the descriptor values b place the solution in region r . Solution sets are evaluated using the Quality Diversity score (QDS) [1], which is the sum of performances of the obtained observations,

$$QDS = \sum_{r=1}^{|R|} f(e_r). \quad (1)$$

When an algorithm is unable to find any solution for a region, e_r may be empty for some r . In this case, it scores some minimum value, often 0, for all unobserved regions. An additional scoring metric, used in surrogate-assisted QD methods which build representative models of the objective function, is a *predicted* Quality Diversity Score. When an algorithm predicts the value of elite points \hat{e} , the quality of such predictions is the true QDS value of the predicted points:

$$QDS_p = \sum_{r=1}^{|R|} f'(\hat{e}_r) \quad (2)$$

$$f'(\hat{e}_r) = \begin{cases} f(\hat{e}_r) & \text{if } b(x) \rightarrow r \\ 0 & \text{otherwise} \end{cases}$$

In other words, if region membership is mispredicted, i.e., a solution is proposed with a predicted region membership which, when evaluated, turns out to be false, the value of this solution is set to 0, or a contextually meaningful lower bound.

III. RELATED WORK

A. Quality-Diversity Algorithms

QD has its origins in the field of evolutionary computation where, when searching for a single global optimum, maintaining genetic diversity is important to avoid premature convergence and to escape local optima [13]. In QD, however, we are not seeking diversity over the search space but searching

²While surrogate-assisted methods exist that implement BO acquisition functions, the point selection mechanism used by these algorithms deviates from BO methodology’

for diversity in descriptor space. In other words, QD wishes to solve many different optimisation problems simultaneously, where each problem is defined over a shared problem space and solved in a behaviourally diverse way.

QD algorithms such as Novelty Search with local competition (NSLC) [14] and MAP-Elites [1] return a *set* of high performing solutions. The solution set returned by MAP-Elites is an example of a *structured* archive, a grid of points where each bin in the grid stores a solution for a region in descriptor space. By contrast, NSLC returns an *unstructured* archive, a population of points of fixed size that sequentially compete, being replaced by points that improve diversity or performance of the population.

The QD problem is similar, but different to other classes of optimisation problems where evolutionary algorithms have been successful:

- In multimodal optimisation, one is looking for several local optima in the search space, but there is only the objective function (no descriptors that define separate partitions, the partitions are implicitly defined by the location of local optima).
- In multi-objective optimisation, one is looking for several solutions with different trade-offs in objectives. Objectives have to be optimised simultaneously, but there are no pre-defined partitions in descriptor space.
- In multi-task optimisation, several related optimisation problems have to be solved simultaneously. Our problem setting could be framed as multi-task optimisation, where each task is to find the best solution belonging to a particular region of descriptor space, i.e., all tasks have the same objective and search space but different complex constraints (descriptor regions).
- In constrained optimisation, there are usually just two categories, feasible and infeasible solutions, and we would like to find the best feasible solution. But identifying the best solution in a particular region of descriptor space could be seen as a constrained optimisation problem, where the region boundaries are the constraints.

QD is a vibrant field of research with recent applications in game level design, [15], urban design, [16], constrained optimisation [17], robotics [3], [18], [19], [20] and the design of nanomaterials [2]. There are many real world situations where producing a diverse set of high performing solutions may be desirable and with such rapid development in the field [21], the range of applications for such algorithms is likely to increase.

A variety of algorithms have been proposed for QD search, and [12] provides a good overview. In the following, the focus is on the algorithms most relevant to our work, namely MAP-Elites, SAIL, and SPHEN. Table I compares the use-cases and outputs from the various algorithms considered in this paper.

B. MAP-Elites

Many modern approaches to QD take place in the context of the MAP-Elites framework [1]. An initial set of points are evaluated on the objective and descriptor functions, the best point in each region is considered an 'elite' and is stored in

TABLE I
COMPARISON OF FEATURES OF QD ALGORITHMS COMPARED IN THIS PAPER, HIGH DIMENSIONALITY IMPLIES > 20 PARAMETERS.

Use-Cases	Map-Elites	SAIL	SPHEN	BOP-Elites
Coupled-Descriptors	✓		✓	✓
High dimensional X	✓			
Output	Map-Elites	SAIL	SPHEN	BOP-Elites
Observed archive	✓	✓	✓	✓
Prediction Map		✓	✓	✓

the archive. Elite points are randomly selected as parents from the archive and combined to produce children that are then evaluated. If a child outperforms the elite for its region in the archive, or is the first individual in that region, it becomes the new elite in that region. This relatively simple yet powerful approach provides good coverage over the regions and finds high-performing points when given a sufficient budget.

C. Bayesian Optimisation

Bayesian Optimisation (BO) is an optimisation approach for expensive black-box functions, see, e.g., [22] for an introduction. The general idea of BO is to start with some initial design points obtained e.g. by Latin Hypercube Sampling, then build a surrogate model of the objective function that not only allows to predict the quality of any solution candidate, but also provides a confidence measure for this prediction. This information is then iteratively used to decide which solution candidate to evaluate next, and to update the surrogate model. Point selection is performed by optimising a so-called *acquisition function*, a mathematically motivated heuristic that balances exploration and exploitation. While this means that an inner optimisation problem has to be solved in every iteration, this is only based on the quick-to-evaluate acquisition function, whereas BO exhibits a very good sample efficiency in terms of the often very expensive objective function.

The most widely used BO surrogate model, and the one implemented in BOP-Elites, is the Gaussian Process (GP) model [23]. Assuming the latent function $f(\cdot)$ is smooth and continuous, it may be modelled with a GP, meaning for any finite set of observed data pairs $\tau^n = \{(x_1, y_1), \dots, (x_n, y_n)\}$, $X_n = \{x_1, \dots, x_n\}$, $Y_n = \{y_1, \dots, y_n\}$ this function is modelled as a multivariate Gaussian random variable. Assuming a prior mean function $m(x)$, the function value $f(x)$ at an unobserved point can be predicted with posterior mean and variance

$$\mathbb{E}[f(x)|\tau^n] = \mu(x) = m(x) + K_*K^{-1}(Y - m(x)) \quad (3)$$

$$\text{var}(f(x)) = s_0^2 = K_{**}^\top - K_*K^{-1}K_* \quad (4)$$

where K is the kernel, or covariance, function. $K_* = K(x, X_n)$, $K^{-1} = K(X_n, X_n)^{-1}$ and $K_{**} = K(x, x)$. While many kernels exist, throughout this paper we use the Matérn 5/2 kernel in the BOP-Elites algorithm:

$$k(x, x') = \sigma_f^2 \sum_{d=1}^D \left[\left(1 + \frac{\sqrt{5}r}{\ell_d} + \frac{5r^2}{3\ell_d^2} \right) \exp\left(-\frac{\sqrt{5}r}{\ell_d}\right) \right] \quad (5)$$

The kernel has $D + 1$ free parameters which must be estimated from the data; D characteristic length-scales ℓ_d which encode the importance of the correlation in dimension d for the similarity in f and signal variance σ_f^2 which sets the maximum covariance for the process.

This derivation is for the modelling of deterministic smooth functions and therefore does not include a meaningful noise term but an adaptation to noisy functions is straightforward. GP regression is covered in depth in the influential book [23].

Key to BO algorithms is the acquisition function (sometimes called infill criterion), a calculation performed on the posterior distribution, which attempts to predict the value of sampling new points in the input space. Examples include upper confidence bound (UCB), probability of improvement (PI) [24], Expected Improvement (EI) [25], [11] and Knowledge Gradient (KG)[26]. In the following we present UCB and EI in more detail, as they will be used later on.

The UCB algorithm was first developed for the multi-armed bandit problem and implemented as a Bayesian acquisition function in [27]. It attempts to minimise regret by choosing to value points as the upper limit of a confidence bound using the posterior mean $\mu(x)$ and standard deviation s_0 defined as

$$UCB(x) = \mu(x) + \beta s_0. \quad (6)$$

The parameter β controls how highly to value reducing posterior uncertainty and therefore can be considered an exploration parameter.

Expected Improvement (EI), which we adapt for BOP-Elites, is a very popular acquisition function as it is both powerful and relatively simple to implement. Consider some candidate point x and $\mu(x)$ the posterior predicted valuation of x from our GP model. EI chooses the point for sampling that will maximise the improvement over our current best performing observation $\{x^*, y^*\}$, i.e., the point x that maximises

$$EI(x) = \mathbb{E}[\max(f(x) - y^*, 0)]. \quad (7)$$

This may be calculated in closed form [11], [25]

$$EI(x) = (\mu_0 - y^*)\Phi\left(\frac{\mu_0 - y^*}{s_0}\right) + s_0\phi\left(\frac{\mu_0 - y^*}{s_0}\right), \quad (8)$$

where Φ and ϕ are the standard cumulative normal and density functions, μ_0 and s_0 are the posterior mean prediction and standard deviation at x from the GP. EI suggests points that have a high probability of yielding improvement either because the surrogate model predicts an improvement with high certainty or because a point has high uncertainty that could yield a high positive value.

D. Surrogate Assisted Illumination (SAIL) algorithm

Surrogate-assisted approaches have shown promise in accelerating problems which are both high-dimensional and expensive [28], [29].

The Surrogate Assisted Illumination algorithm (SAIL) [7] is a QD algorithm that attempts to leverage the power of a GP surrogate model of the objective landscape. SAIL performs MAP-Elites using Upper Confidence Bound (UCB) as an objective function on the surrogate model. The underlying idea

is to attempt to build a good model of the objective function, then use it to predict elites. Once complete, SAIL outputs a *predicted* elite for each region, and a model of the objective function which may provide additional insights. SAIL does not model the descriptor space and instead assumes that solution descriptors can be quickly and accurately obtained, independent of objective function evaluations.

SAIL was the first approach to use surrogate models and ideas from BO to enhance the performance and increase sample efficiency of MAP-Elites [30]. While MAP-Elites is effective, it typically takes many observations to reach a high performing solution. SAIL uses the evolutionary algorithm approach of MAP-Elites on the cheap-to-evaluate surrogate model, with the UCB function evaluated on the GP as the objective function. This approach explores points with high objective performance and high variance and generates both a set of predicted elites and a GP that *illuminates* the relationship between performance and behaviour in a meaningful way. SAIL has been primarily applied as a tool for exploration in search spaces where simulations or experiments are expensive, such as aerodynamic design and materials science [31], [30], [32], [9], [33].

E. Surrogate-assisted Phenotypic Niching

The general framework introduced by SAIL for using surrogate models to assist QD has been adapted to predict descriptors along with objectives [9] in order to find solutions with specific behaviours and as a component in a larger generative design process that is iteratively applied to narrow in on promising design spaces [32], [28].

SPHEN builds upon SAIL by modelling the descriptor space with a separate GP surrogate model. SPHEN uses the posterior mean of the GP to predict the region in descriptor space a solution is likely belonging to, and accepts the prediction taking no account of the uncertainty in the model. This allows the algorithm to predict region membership and implement SAIL where the descriptors are expensive and coupled. SPHEN does not directly drive exploration in the descriptor/phenotype space, instead it updates its descriptor model indirectly through the observations gained using SAILs acquisition function over the objective.

F. Deep model based methods

The state of the art for QD optimisation includes deep neural network (DNN) based methods such as Deep Surrogate Assisted MAP-Elites (DSA-ME) [28], and Deep Surrogate Assisted Generation of Environments (DSAGE) [34]. These methods have applications in high dimensional, complex domains such as generating human-robot interaction scenarios [35] and optimising warehouse layout for multi-robot warehousing systems [36]. DNN methods typically require large datasets to perform well, with GP methods requiring an order of magnitude less. This makes Deep model based methods unsuitable for expensive domains and highlights a key benefit of the current work. Table II provides a comparative overview of dataset sizes and dimensions used in recent studies within this field, DNN based methods are highlighted gray.

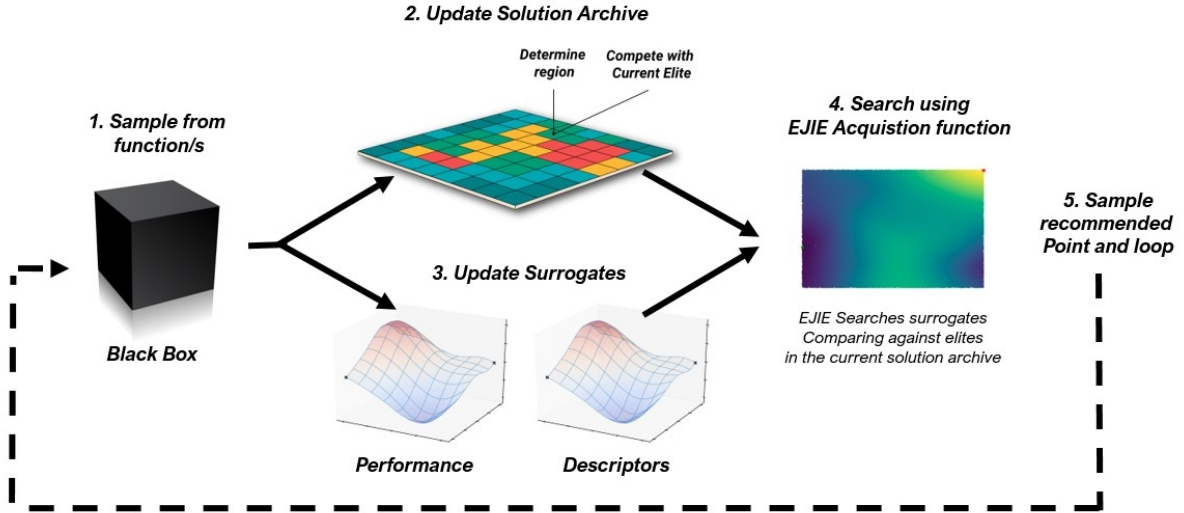


Fig. 1. A visualisation of the BOP-Elites algorithm

TABLE II
A COMPARISON OF PARAMETERS AND EVALUATIONS OF RECENT WORK

Approach	n_parameters	n_evals
SAIL [31] [30]	10, 16	1000
SPHEN [9]	16	1024
BOP-Elites	10	1250
DSA-ME [28]	180	10,000
SAG-HRI [35] ¹	1024	10,000
MRCD-LD [36] ²	432, 4224	10,000
DSAGE [34]	256, 896	100,000

G. Bayesian Optimisation for Quality-Diversity

While the SAIL algorithm utilises UCB, a BO acquisition function, to search for the best points to sample, its selection criterion departs from traditional BO methodology. In the acquisition step, SAIL fills an *acquisition map* with points that score highly on the UCB function before randomly selecting points from this acquisition map to evaluate on the true objective function. This selection process induces the possibility of randomly selecting a point from a region for which there is no real improvement to be found. Traditional EI, by contrast, attempts to compare the improvement a point provides to the objective function and selects the point for which there is the highest improvement.

In practice, EI often outperforms UCB in direct comparisons [37], [38] and while the performance of the UCB algorithm can be affected by parameter β , EI has no parameters to tune. These arguments provide compelling justification for creating a principled implementation of EI for QD.

IV. BOP-ELITES ALGORITHM

BOP-Elites is a model-based Bayesian optimisation algorithm for QD problems. It is designed to work with an expensive black-box objective function $f(x)$ and coupled descriptor

function $b(x)$ which we assume can be suitably modelled with a GP. BOP-Elites attempts to identify the single next point to sample that will provide the greatest expected improvement to the QDS. First, BOP-Elites builds GP surrogate models of y and, when the descriptor function is coupled to the objective function, each b_i . At each iteration, it will use its acquisition function to search the model for the next point to be sampled. After each function evaluation, we update our models and attempt to increase the value of our solution set. The algorithm proceeds until the search budget has been exhausted. The general flow of the algorithm is visualised in Fig. 1.

As seen in Algorithm 1, BOP-Elites is initialised with a set of n_0 points Sobol sampled [39], [40] from X . Following [11], we take that number to be 10 times the number of dimensions of the problem. The objective and descriptor functions are evaluated at each initial point and stored as a list of observations, the data $\mathcal{D} = \{x_i, y_i, b_i\}_{i=1}^n$. We store ‘elites’ $\hat{S} = \{e_1, \dots, e_{|R|}\}$ in an archive, as in MAP-Elites, this is updated whenever a new point is sampled that performs better than the existing elite in the corresponding region.

A. BOP-Elites Acquisition Function

1) *Predicting region membership*: We propose an extension to the expected improvement acquisition function tailored to quality-diversity search. The simplest approach would be to calculate the expected improvement for each region independently and take the highest one. However, this requires identification of the specific region to which a given point belongs, which may be computationally expensive.

This issue is critical since region membership — and hence, which elite we should compare against — must be predicted and comparison to the wrong elite will mis-value a point. BOP-Elites attempts to solve this issue by building surrogate models of the descriptor space and calculating the posterior probability $\mathbb{P}(x \in r | \mathcal{D})$ of a point belonging to region r .

As behavioural descriptors are evaluated by m descriptor functions, we fit m GPs using our observed data and make

¹Surrogate Assisted Generation of Human-Robot Interaction Scenarios

²Multi-Robot Coordination and Layout Design for Automated Warehousing

m predictions using the posterior mean positions $\hat{b}(x) = \{\hat{b}_1(x), \dots, \hat{b}_m(x)\}$ and variances $s(x) = \{s_1(x), \dots, s_m(x)\}$ along each of the descriptor dimensions using Eqns. 3 and 4. Regions are formed by intersecting partitions across descriptor dimensions and the probability of a given point x lying within a partition $\mathcal{P}_{j,r}$ in dimension j that forms part of region r is calculated as:

$$\mathbb{P}(b_j(x) \rightarrow \mathcal{P}_{j,r}) = \Phi\left(\frac{\hat{b}_j(x) - UB}{s_j(x)}\right) - \Phi\left(\frac{\hat{b}_j(x) - LB}{s_j(x)}\right), \quad (9)$$

where Φ is the posterior CDF of a uni-variate Gaussian and LB and UB are the lower and upper bounds for partition $\mathcal{P}_{j,r}$. Therefore, the likelihood of belonging to any given region is determined as the product of the individual partition probabilities. Assuming independence of the descriptors, we get the following:

$$\mathbb{P}(x \rightarrow r|\mathcal{D}) = \prod_j^m \mathbb{P}(b_j(x) \rightarrow \mathcal{P}_{j,r}) \quad (10)$$

2) *Calculating Expected Improvement when there is no elite*: A second issue implementing EI [11] arises as calculating EI_r , the Expected Improvement for a specific region, requires comparing our posterior mean performance of a point μ_0 to the value of the current elite $f(\hat{e}_r)$ in our solution archive.

$$EI_r(x) = \mathbb{E}[\max(\mu_0 - f(\hat{e}_r), 0)] \quad (11)$$

$$= (\mu_0 - f(\hat{e}_r))\Phi\left(\frac{\mu_0 - f(\hat{e}_r)}{s_0}\right) + s_0\phi\left(\frac{\mu_0 - f(\hat{e}_r)}{s_0}\right) \quad (12)$$

We propose to compare all regions, including those without an observed point. Given that an empty region contributes nothing to the QDS of our archive, it seems sensible to assume that the elite performance of an empty region is zero. This means that BOP-Elites will value empty regions highly and this will drive region-filling behaviour as BOP-Elites is an improvement-greedy strategy.

3) *Expected Joint Improvement of Elites*: Combining the above, we can calculate the region-specific EI which returns the expected improvement over the elite in region r weighted by the posterior probability that it is in competition with that elite.

The proposed Expected Joint Improvement of Elites (EJIE) acquisition function, outlined in Algorithm 2, takes the sum of these values for all regions to predict the improvement the point x will provide to our current solution archive.

$$EJIE(x) = \sum_{i=1}^{|R|} \mathbb{P}(x \rightarrow r_i|\mathcal{D})EI_{r_i}(x). \quad (13)$$

This is similar to the approach proposed for constrained Bayesian Optimisation (CBO) in [41] where the region boundaries are constraints. The key difference is that CBO is looking for a single optimal value given some constraints while QD

attempts to identify the optimal value for each of a set of non-overlapping constrained regions.

Eqn. 13 gives high valuations to points that are likely to provide large improvements to the QDS. The predefined regions force the algorithm to search in areas of the input domain that single-objective optimisation would ignore, naturally diversifying the solution set. The posterior models provided by BOP-Elites furthermore provide insightful illumination of the search space, offering the opportunity for prediction of behavioural quality and objective performance at any point in the input domain with quantifiable confidence bounds.

BOP-Elites can be easily adapted for use with decoupled descriptors, where descriptors are cheap to evaluate and can be queried at any time with no appreciable cost. In this case the region membership is known and the $\mathbb{P}(x \rightarrow r|\mathcal{D})$ in Eqn. 13 becomes a delta function for the given region.

In the following sub-section, we address difficulties with the EJIE acquisition function for QD problems and propose further adaptations to the algorithm.

B. Optimisation of the acquisition function

Optimisation of the acquisition function is performed using pattern search (or direct search) [42] over the acquisition function, with a number of restarts. This derivative free technique can be used to optimise acquisition functions which are not continuous everywhere [43], [44], which is the case for QD if descriptor values are known due to the way the acquisition function compares improvement over region-specific elites, making it only piece-wise continuous. The result from each optimisation run is added to a list of *candidate points* and the best point of these is evaluated on our true function. As the surrogate model of descriptor functions improves — and therefore posterior uncertainty over region membership reduces — the boundaries between regions become sharper and approximate the discontinuities of the instance above. The acquisition value, can jump dramatically at the boundary between regions, causing problems for continuous optimisers. As well as using direct search, and in order to avoid issues with getting stuck in local optima, BOP-Elites attempts to identify a large number of diverse starting points in each iteration.

C. Initial Diverse Points

Using Sobol Sampling over X and evaluating each solution on the descriptor surrogate with the posterior mean descriptor, we predict region membership for each point in the sample. The points are evaluated on the EJIE acquisition function - comparing only against the elite indicated in the region prediction - and then ranked. The best performing points predicted to be from unique regions will become the initial restart points for local optimisation along with a number of randomly selected points. The idea here is that we start each optimisation round from promising points well distributed in descriptor space. The optimisation process is detailed in the pseudo-code for Alg. 1 and Alg. 2.

D. Slow down of convergence with hard to reach regions

Some regions of the descriptor space may be hard to reach because they are very small. It makes sense for an algorithm to spend time seeking out these regions as they contribute both to the diversity of the solution set and the total QDS, but how can we tell the difference between such a region and one that is in fact unreachable?

Our experimental setup, using MAP-Elites style discrete regions, makes this a very difficult task as the GP surrogate gives non-zero probability to neighbouring descriptor values. As we get close to a discrete boundary in continuous space that probability will increase to nearly 0.5 of the posterior probability, even if the GP is well trained with indicative points. Even if the neighbouring region is not reachable, then the calculated expected improvement will be large leading to a huge over-valuation of a point on the boundary of reachable space.

In order to tackle this problem, we introduce a cut-off value to the probability which means that EJIE becomes

$$EJIE^+(x) = \frac{\sum_{i=1}^{|R|} \rho(x \rightarrow r_i) EI_{r_i}(x)}{\sum \rho} \quad (14)$$

$$\rho(x \in r_i) = \begin{cases} \mathbb{P}(x \rightarrow r_i | D) & \text{if } \mathbb{P}(x \rightarrow r_i | D) > \omega \\ 0 & \text{otherwise} \end{cases}.$$

The idea here is to ignore regions in our calculations that are too unlikely to be the region of the considered solution candidate. The cutoff value ω has an initial value of $\frac{1}{(\text{number of regions})}$ which should be easily surpassed by any region we wish to consider when our GP is initialised with a sufficient number of well distributed points. We expect our confidence in the model to increase as we take more observations, so we introduce

$$\omega = \frac{1}{2} \left(\frac{2}{R} \right)^\gamma, \quad \gamma = \sqrt{\frac{10d}{(\alpha - 2\beta + t)}} \quad (15)$$

where α is the mis-specification count, i.e., the number of times the algorithm incorrectly attributes more than 50 percent of the acquisition value to a single region which, when evaluated, is a misprediction in descriptor space. β is the over-specificity value, the number of times the algorithm is unable to find any expected improvement due to excluding too many regions. β therefore increases whenever the optimisation step returns no value in any *candidate points*. This quantity has the desirable property of being $\frac{1}{n}$ when the algorithm starts, due to the initial budget being $10d$, and as $t \rightarrow \infty, \gamma \rightarrow 0$ and therefore $\omega \rightarrow \frac{1}{2}$. See Algorithm 2 line 6-7.

E. Dealing with invalid points

For many real-world problems, there are regions of the search space that for various reasons do not produce a valid performance or descriptor evaluation. In this work we tried to take the approach that SAIL did, which is to assume that non-convergence is a cheap result and we can essentially treat such events as having no real cost. In this way, if BOP-Elites fails to receive a successful evaluation of a point, it will move on to the next candidate point, or search the acquisition function for an alternative.

This performed very poorly in our early experiments as BOP-Elites searches for points it predicts are high performing and unless it is able to reduce the predicted performance of a point it will continue to suggest it. We thus model validity directly via means of an extra classifying GP or, in our case, a Support Vector Machine (SVM). We chose the latter due to the experimental assumption that NaNs (a Not-a-Number error from the simulator), or invalid points, are cheap or free to evaluate and SVMs work well with a large number of data points, with low computational overhead.

In addition to classification we implement the Platt scaling method [45] to produce a probability estimate for feasibility $\mathbb{P}_{feas}(x)$. We weight our acquisition value by the probability of being valid. so we now have

$$EJIE(x)^{++} = EJIE(x)^+ \mathbb{P}_{valid}(x). \quad (16)$$

The modelling of validity becomes active when the first invalid point is found, otherwise BOP-Elites uses $EJIE(.)^+$ as default.

F. Extending BOP-Elites to work with many regions

While BOP-Elites returns a solution archive filled with evaluated solutions as the final output, SAIL, a surrogate *assisted* algorithm, returns a Prediction Map (PM) [31] as a final output, which is created by running MAP-Elites over the final posterior mean GP surrogate model. The idea then is to use the rich models generated by the algorithm for predictions in all regions, even when there are many more regions than evaluations in the budget. BOP-Elites is designed for coupled descriptor functions, in which case we additionally model descriptor functions. This introduces the possibility of region mis-prediction, i.e., predicting that a point maps to a certain region which, when evaluated on the true descriptor functions, turns out to be incorrect. In the case of mis-prediction a solution contributes no value to the QDS. When our surrogate models are not accurate, we expect a severe drop in performance due to mis-predictions. BOP-Elites will produce two results:

- 1) A solution archive made of observed points
- 2) A Prediction Map.

Prediction Maps: Throughout the optimisation process, the performance of the algorithm is reliant on the quality of the surrogate models. Analysing the quality of the underlying models can be useful in seeing how the algorithm is performing, and one way of doing so is through a PM. A PM uses the surrogate models to predict elites for all regions, including those not visited. PMs are created by performing MAP-Elites over the surrogate models where the value function is chosen to find the best points without exploration. In SAIL's original formulation, as the descriptor values are known, the performance measure used to generate the PM is simply the posterior mean of the objective GP and the recommended point for any region r should be the posterior mean maximal point for each region:

$$\hat{e}_r = \arg \max_x [\mu(x) | b(x) \rightarrow r]. \quad (17)$$

In the coupled descriptor case we must also predict values for the descriptor. The method implemented in the original SPHEN algorithm is to assume the descriptor GPs are correct and accept the region predicted by the posterior means. In this work we refine this value by acknowledging that neither SPHEN nor BOP-Elites actively control exploration of descriptor space, except by virtue of search for high objective performance. In this case we adapt Eqn. 17 as:

$$\hat{e}_r = \arg \max_x \left[\mu(x) \mathbb{P}(\hat{b}(x) \rightarrow r) | \hat{b}(x) \rightarrow r \right], \quad (18)$$

where we multiply the posterior predicted mean value by the probability of x being in the region and $\hat{b}(x) \rightarrow r$ is the region indicated by the mean of the posterior descriptor models.

G. Upscaling of archives

One interesting unexplored potential of the PM approach is that while a structured archive imposes a discretisation over the descriptor space, the models learned throughout the optimisation process are continuous. This allows us to use points collected during a coarsely discretised solution archive to build a rich model of the underlying functions and predict solutions for a much more finely discretised solution archive. We call this process ‘upscaling’.

A key difference between previous surrogate assisted methods and BOP-Elites is the way they balance exploration and exploitation. SPHEN uses a tunable parameter β to weight towards variance reduction, which leads to early model-building behaviour³. By contrast BOP-Elites is improvement greedy which means it fills regions starting from the highest performing regions first. As BOP-Elites takes many observations in a concentrated area, the underlying models can suffer until BOP-Elites gradually moves away from the higher performing regions. As SPHEN reduces posterior variance early, it leads to a good early model.

A good early model is desirable for BOP-Elites, as the quality of predictions and therefore performance of the algorithm depend on the model. In this case, we propose to utilise the region filling behaviour of BOP-Elites as a tool.

1) *Initial Upscaling*: As BOP-Elites is improvement greedy, when the number of regions far exceeds the search budget, BOP-Elites will focus on sampling in high-performing regions. While this will maximise the immediate return, it can lead to local clustering of evaluations, see Fig. 11A, and miss exploring nearly the entirety of the rest of the space, which is likely to affect model quality. As BOP-Elites is excellent at filling regions with high performing points, we first give it the task of filling a coarsely discretised solution archive, say a two dimensional descriptor space of 5^2 regions, [5, 5]. BOP-Elites will fill these regions, creating diversity over the descriptor space due to how spread out the regions are. Once BOP-Elites has either filled the regions or has spent a small budget, twice the number of regions, we increase the discretisation to the original resolution, leveraging the now improved model. This initial phase of BOP-elites, Algorithm 1 line 5-6, improves

the early prediction models (See Fig 12) and improves the surrogate models aiding the optimisation process throughout.

2) *Upscaling to higher resolution*: As the number of points a Gaussian Process model can feasibly handle is limited due to computational complexity and numerical issues, QD problems with higher resolution solution archives are infeasible for surrogate based methods to process naively. Whilst possible solutions may include sparse GPs, inducing points etc., in this work we take a simple approach to make predictions for higher resolution solution archives from optimised lower resolution solution processes. The idea here is to take the full observation set from a 25×25 solution (both the elite set and the observation set used in the optimisation) and make continuous predictions from the GP to fill a higher resolution grid. We present the results of these efforts in the following sections.

Algorithm 1 BOP-Elites Pseudo-code

Initialization:

- 1: Sample initial dataset $\mathcal{D}_0 = \{X_0, Y_0, \mathbf{b}_0\}$
- 2: Elite points in each region enter solution archive, S
- 3: Build Objective and Descriptor surrogate models $\hat{f}(\cdot), \hat{\mathbf{b}}(\cdot)$

Optimization Loop:

- 4: **while** $n < N_{max}$ **do**
 - 5: **if** S is full or $n > 2|R|_{init}$ **then**
 - 6: upscale archive
 - 7: Find X_d , diverse initial points ▷ by IV-C
 - 8: **for** x in X_d **do**
 - 9: compute $EJIE^+(x)$ ▷ Algorithm 2
 - 10: sample x_n at $\arg \max(EJIE^+(x))$
 - 11: **if** $f(x_n) > f(\hat{e}_{b(x_n)})$ **then**
 - 12: $\hat{e}_{b(x_n)} = x_n$
 - 13: update posterior models with $x_n, f(x_n), b(x_n)$
 - 14: $n = n + 1$
 - 15: **end while**
 - 16: **Return:** Elites
-

Algorithm 2 EJIE+ Acquisition Function

- 1: **Inputs:** $x, \hat{f}(\cdot), \hat{\mathbf{b}}(\cdot), \omega$
 - 2: $\hat{b}(x) \rightarrow \hat{r}_x$ ▷ predict region membership from posterior mean
 - 3: **for** r in R **do**
 - 4: $EI_r(x, \hat{f}_r(\cdot), \hat{e}_r)$ ▷ by Eqn. 11
 - 5: $\mathbb{P}_r = \mathbb{P}(x \in r|D)$ ▷ by Eqn. 10
 - 6: **if** $\mathbb{P}_r < \omega$ **then**
 - 7: $\mathbb{P}_r = 0$ ▷ remove region if probability $< \omega$ as Section IV-D
 - 8: Normalise probabilities
 - 9: $EJIE+ = \sum_r \mathbb{P}_r EI_r$
 - 10: **return** EJIE+
-

V. EVALUATION METHODOLOGY

We compare BOP-Elites against state-of-the-art QD algorithms on a number of benchmarks designed to simulate

³In this paper we present results where the β has been tuned, but incorrectly specified β can lead to poor results.

problem settings with interesting properties, including invalid solutions, different dimensionality, and either 1 or 2 descriptor functions. We measure performance by the QDS of the solution archive and when evaluating PMs we observe the true values of the recommended points, the QDS_p of the PM, attributing a default low valuation for contributions from misspecified points in the coupled descriptor case.

A. Benchmark algorithms

BOP-Elites is designed for problems where descriptor functions are coupled and assumed to be expensive. We compare against 3 baselines: Sobol sampling, MAP-Elites and the state of the art surrogate assisted method which is SAIL for the decoupled descriptor case and SPHEN when the descriptors are coupled functions.

Sobol sampling [39], [40] is a quasi-random number generation technique designed to sequentially generate a set of points that converge to uniformity over the unit hypercube as the number of samples t approaches infinity. While Sobol sampling is not a Quality-Diversity algorithm per se, retaining the best points found in each region can be employed to construct a valid solution archive.

We use MAP-Elites [3] with uniformly sub-partitioned descriptor dimensions. In our experiments we use Gaussian mutation with mutation step size $\sigma = 0.1$ and 50 children per generation, no crossover and the population is the set of elites in the archive.

For problems with decoupled descriptor functions, we compare against the SAIL algorithm [7] which builds a surrogate model of the objective with exactly the same specification as BOP-Elites' GPs using the BOTORCH [46] framework — see the next section - and runs MAP-Elites over those GP's with the same setup as standalone MAP-Elites, as above, except that it uses the UCB function as the objective. This function has a user-defined parameter β , and we have optimised it at $\beta = 3.7$ which performed well across all region numbers and best, on average, across domains.

SPHEN is an implementation of SAIL, with all the same settings, that additionally models the descriptor values with a surrogate model as described in Section III-E.

B. Experimental configuration

BOP-Elites uses a noise-free GP model implemented in BOTORCH using a Matérn 5/2 kernel with standardised objectives and normalised search inputs. The Matérn 5/2 kernel frequently serves as a default kernel in BO due to its computational efficiency and balance between capturing smoothness and complexity. We use the pattern search optimiser by PY-MOO in the search step with default settings and a maximum budget of 100 generations or 1000 evaluations. BOP-Elites has one parameter which may benefit from problem specific tuning; the number of restarts per iteration in the optimiser. Rather than fine-tuning this, we chose a naive setting of 10 restarts for the 10×10 , 25×25 and 50×50 solution archives, increasing this budget is almost certain to further improve the performance of BOP-Elites. We compare BOP-Elites with a small budget of observations against MAP-Elites at different

observation budgets. As MAP-Elites generates 50 children per generation each 1000 observations equate to 20 generations.

Our results compare BOP-Elites performance based on both, the observed solution archive and the value of the predicted map. We chose to limit the number of observations for BOP-Elites to 1000 for the 10×10 and 1250 for the 25×25 and 50×50 archives due to the computational time of making predictions from the Gaussian process at high observation count. 1250 is double the number of regions in the 25 by 25 structured archive and was sufficient to find a very strong solution. For the 50×50 archive, BOP-Elites does not have sufficient budget to create a good solution but we include these results to show a failure case which can be resolved by upscaling.

C. Benchmark functions

For testing, we use a range of artificial problems that simulate some of the challenges found in real-world problems. We also test the PARSEC airfoil optimisation as example of a close-to-real-world problem. The following provides a short summary of the test problems used, more details can be found in Appendix A

1) *Synthetic Gaussian Process problems*: We generate realisations of a single output, 10 dimensional input, Gaussian Process model with predefined lengthscale, for both objective and each descriptor function.

The benefit of generating functions from GPs is that we can create an infinity of unique functions quickly and efficiently without concern for model mismatch with the GP surrogate model. Lengthscales are chosen uniformly at random from 0.1 to 1 and algorithms are compared on functions generated from the same seeds.

2) *Mishras Bird function*: Mishras Bird function is a standard optimisation benchmark [47]. We define 2 simple descriptor functions: $B_1(x) = -x_1$, $B_2(x) = -x_2$.

Whilst being low dimensional, the problem exhibits an interesting property as there is a small region that vastly outperforms the rest of the domain in terms of function value.

3) *Robot Planar Arm*: This well-known QD benchmark problem [3] is a simulation of a planar robot arm with an $n = 4$ dimensional unit hypercube $[0, 1]^4$ as decision space and two descriptor functions.

This problem simulates the ability of a robot arm to reach feasible regions in a $2d$ plane while minimising the standard deviation of the position of the individual joints.

4) *Rosenbrock 6d*: The Rosenbrock function [48] is a relatively smooth function in 6 dimensions. We used 2 descriptor functions with the form

$$B_1(x) = \frac{1}{2}(x_1 + x_2), B_2(x) = (x_3 - 1)^2$$

5) *PARSEC: 2D Aerofoil optimisation problem*: This is a well known and well studied problem in engineering. We used the 10 dimensional PARSEC representation of the parameters [49]. This problem has also been the key benchmark of previous SAIL papers. The PARSEC domain features invalid regions, for which the adaptations in Section IV-E were implemented.

D. Coupled and decoupled functions

All benchmark functions may be evaluated either with coupled or decoupled descriptor functions. In the coupled case the behaviours may only be observed alongside the objective, which is assumed to be expensive. Decoupled descriptors are assumed to be cheap and able to be evaluated independent of the expensive objective.

VI. EMPIRICAL RESULTS

We present tables of summary statistics from 100 experiments for each algorithm. In the top rows of a results table we compare the performance of the evaluated solutions in the archive of algorithms after 1000 function evaluations in the 10x10 case or 1250 function evaluations for the 25x25 and 50x50 case. Besides the evaluated solution archives we generate PMs, which are generated as in Section IV-F and measured as the true value of the points suggested as in Eqn. 2. In the bottom rows we show the performance of MAP-Elites and Sobol sampling after 50,000 evaluations. Highlighted cells indicate the best value found among all solutions. Grey cells indicate the best mean results, statistically tested above a 99% confidence level with a t-test. In cases where the same algorithm generates both an observed and a predicted result, and neither of these results significantly outperforms the other, we only denote the result that exhibits the highest mean performance.

A. Coupled descriptors

1) *Synthetic Gaussian Process functions - coupled*: Table III contains results from experiments on synthetic GP functions with 1 generated descriptor function. As each random seed produces a unique objective and feature function, we normalise the performance (against the best score found by all algorithms) for each seed and take the average. BOP-Elites significantly outperforms all other algorithms even if those use much higher function evaluations. SPHEN appears to converge to a sub-optimal score and this is because it fails to fill all available regions, particularly in the 10 region case. Fig. 2 shows both SPHEN and BOP-Elites increase in QDS value quickly, with BOP-Elites improving quickest and converging at an optimal value. Synthetic GP benchmarks help to eliminate the problem of model mis-specification for surrogate based methods and this showcases the potential of BOP-Elites in ideal conditions. While SPHEN is also a surrogate based method it struggles to achieve optimal performance and examination of the returned values show that it sometimes struggles to reach all the regions.

2) *Mishras Bird function - coupled*: In Table IV and Fig. 3, we see strong early performance of BOP-Elites for both prediction map solutions (10x10 and 25x25). This is exactly as we would expect with a smooth and well behaved combination of objective/descriptor functions. BOP-Elites clearly outperforms other methods in the final stages, and the better points found by BOP-Elites enable an improvement in the PMs. We notice two things in the performance of the BOP-Elites PM: The error is lowest at 700 function evaluations before increasing again, we also notice an increase in the Standard Error of

TABLE III
FINAL QDS ON THE 10D SYNTHETIC BENCHMARK WITH COUPLED DESCRIPTOR FUNCTIONS. ALGORITHMS ON THE TOP USE 1000/1250 FUNCTION EVALUATIONS FOR RESPECTIVE ARCHIVE SIZES. ALGORITHMS AT THE BOTTOM USE 50K FUNCTION EVALUATIONS.

QDS:	1d. 10		1d. 25	
	Mean	S.E.	Mean	S.E.
BOP-Elites	1.00	0.000	1.00	0.0005
SPHEN	0.73	0.007	0.88	0.0052
BOP-Elites-PM	1.00	0.000	0.99	0.0023
SPHEN-PM	0.76	0.008	0.88	0.0052
MAP-Elites	0.63	0.011	0.72	0.0087
MAP-Elites 50K	0.93	0.005	0.97	0.0091
Sobol 50K	0.81	0.009	0.76	0.0103

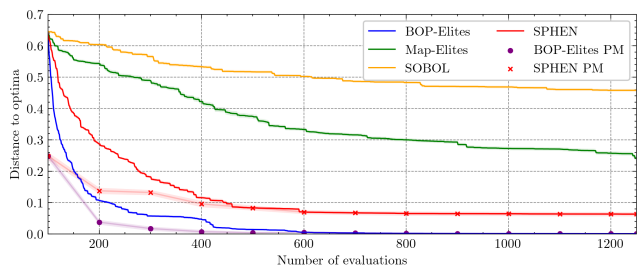


Fig. 2. Convergence of BOP-Elites in comparison to Map-Elites, SOBOL and SPHEN on the synthetic GP problem with a 1d 25 region archive. We normalise the performance of the algorithms for each seed before taking the mean over 100 experiments.

the PM performance. While the error is relatively small, this unusual behaviour is almost certainly due to the way BOP-Elites is focusing on visiting high performing points and not on reducing the uncertainty of the model, in particular in the descriptor predictions.

Another intriguing observation is the impressive early performance of SOBOL-sampling in this domain, surpassing that of SPHEN. This outcome can be primarily attributed to the smooth, flat behaviour across the function, with value distribution being even throughout the domain. As a result, effective performance can be achieved simply by covering the input space, an aspect that SOBOL excels at in two dimensions where the input also serves as the descriptor.

BOP-Elites demonstrated remarkable efficiency in achieving full coverage, finding a solution in each region, at both resolutions. Conversely, SPHEN's observation set achieves 95% coverage on average and this contributes to its lower archive score. This underperformance in region coverage is not an isolated issue, as it is likely to have a knock-on effect on the quality of the surrogate model, and subsequently, the performance metric score.

3) *Robotarm - coupled*: Table V summarises results on the robotarm benchmark problem. BOP-Elites returns the best solution archive for the 10x10 and 25x25 solution archive and much better than even MAP-Elites with 50k evaluations. Figure 5 shows BOP-Elites has excellent early performance that slows between 600 and 900 evaluations before accelerating. This is likely where BOP-Elites is refining the ω value used to focus its search and filter out unreachable regions. The excellent end performance indicates that BOP-

TABLE IV
FINAL QDS ON THE MISHRA 2D BENCHMARK WITH COUPLED
DESCRIPTOR FUNCTIONS.

QDS: Mishras Bird	10x10		25x25	
	Mean	S.E.	Mean	S.E.
BOP-Elites	13776.18	0.45	79973.29	1.186
SPHEN	13762.26	3.87	68587.21	237.02
BOP-Elites-PM	13776.55	0.28	79785.12	27.8
SPHEN-PM	13744.15	1.44	79552.95	0.81
MAP-Elites	11136.70	89.61	25473.58	196.45
MAP-Elites 50K	13666.59	1.63	79033.73	5.80
Sobol 50K	13710.03	1.08	79536.39	2.49

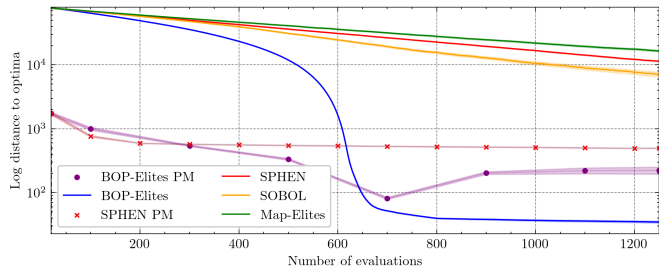


Fig. 3. Convergence of BOP-Elites in comparison to Map-Elites and SOBOL for the Mishra Bird Function with 2 descriptor dimensions and a $[25 \times 25]$ solution archive. BOP-Elites rapidly finds a near optimal solution.

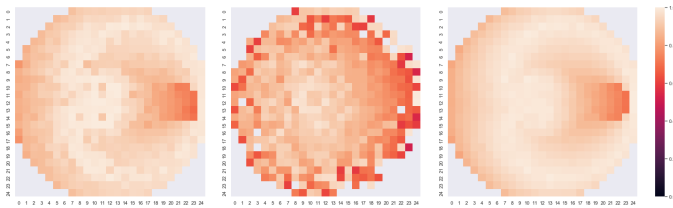


Fig. 4. These plots show an indicative result for a single solution archive of 25×25 , x and y axis are descriptor dimensions: (left) BOP-Elites finds a near-optimal solution and explores all regions after 1250 observations. (centre) After 10,000 observations, MAP-Elites is only starting to fill all regions and find high-quality solutions. (right) MAP-Elites, with 200k total observations, achieves a near-optimal solution set, surpassing BOP-Elites only after $\sim 100k$ points.

Elites is also a good choice for QD problems with unreachable regions. SPHEN doesn't require any adaptations to handle the unreachable regions in this domain and shows very good early performance, but falls behind BOP-Elites towards the end of the budget.

TABLE V
FINAL QDS ON THE ROBOT ARM 4D BENCHMARK WITH COUPLED
DESCRIPTOR FUNCTIONS.

QDS: robotarm	10x10		25x25	
	Mean	S.E.	Mean	S.E.
BOP-Elites	85.14	0.00	500.12	0.11
SPHEN	84.48	0.68	464.41	8.485
BOP-Elites-PM	84.91	0.02	502.30	0.14
SPHEN-PM	83.49	0.17	451.07	0.26
MAP-Elites	50.27	0.69	140.10	1.31
MAP-Elites 50K	84.15	0.02	493.15	0.40
Sobol 50K	81.21	0.08	267.07	1.15

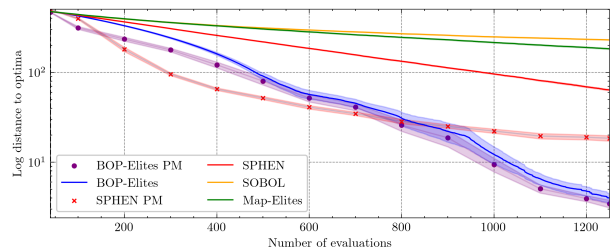


Fig. 5. Convergence on the Robot Planar Arm function with 2d coupled descriptor space over a $[25 \times 25]$ solution archive. A comparison between BOP-Elites and Map-Elites, SOBOL and SPHEN. BOP-Elites and SPHEN model both objective and descriptor values. Lines represent mean performance with the standard error around the mean.

TABLE VI
FINAL QDS ON THE ROSENBRACK 6D FUNCTION WITH COUPLED
DESCRIPTOR FUNCTIONS.

QDS: Rosenbrock6d	10x10		25x25	
	Mean	S.E.	Mean	S.E.
BOP-Elites	152837.88	0.04	903547.61	2.27
SPHEN	151488.48	518.90	822086.93	8011.94
BOP-Elites-PM	152340.82	48.27	903327.12	4.20
SPHEN-PM	152450.18	48.27	896067.23	3.57
MAP-Elites	99826.52	970.44	254849.92	2708.37
MAP-Elites 50K	151674.80	18.31	899302.84	233.07
Sobol 50K	106240.71	188.91	543038.19	344.11

4) *Rosenbrock 6d function - coupled*: Table VI and Fig. 6 shows a comparison of BOP-Elites, SPHEN, MAP-Elites and Sobol sampling for the Rosenbrock function with 6 input dimensions. We again see dominant performance for BOP-Elites and corresponding strong performance for the BOP-Elites PM, with an interesting dip in the PM error at around the 700 observations mark. At this point in the algorithm BOP-Elites changes its behaviours from exploration (filling regions) to exploitation (improving existing regions). As regions, by definition, represent diversity, the PM is now not improving as rapidly as the points found are no longer as useful in improving the surrogate model.

Additionally, the performance of the BOP-Elites Prediction Map now under-performs, on average, the observed solution map towards the end of the budget. This appears to be due to uncertainty over region membership at the boundaries leading to occasionally suggesting a point that ends up being in a region other than that predicted, leading to performance loss.

BOP-Elites consistently achieves perfect coverage, further solidifying its superiority in this domain. We notice a similar trend regarding the coverage of the SPHEN observation sets, which achieves only 93% coverage.

5) *PARSEC - coupled*: For the PARSEC domain, which contains invalid regions, Table VII shows once again that BOP-Elites outperforms the other algorithms. This is unsurprising as BOP-Elites is the only method capable of actively modelling the infeasible regions. While we assume there is no penalty for attempting to evaluate an invalid point, we note that, for the coupled descriptor problem, BOP-Elites attempts to make an average of 94 evaluations of invalid points, compared with 362 for the SPHEN algorithm from a run of 1250

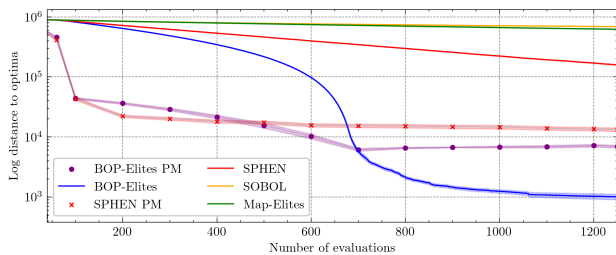


Fig. 6. Convergence of BOP-Elites in comparison to Map-Elites and SOBOL on the Rosenbrock 6d problem with a 2d coupled descriptor space and a $[25 \times 25]$ solution archive

TABLE VII
FINAL QDS ON THE PARSEC 10D DOMAIN WITH COUPLED DESCRIPTOR FUNCTIONS.

QDS: PARSEC	10x10		25x25	
	Mean	S.E.	Mean	S.E.
BOP-Elites	370.09	0.23	2206.10	1.96
SPHEN	361.21	1.63	1484.73	21.24
SPHEN - PM	369.15	0.61	1863.12	8.32
BOP-Elites - PM	370.08	0.17	2090.97	5.19
MAP-Elites	205.77	1.21	293.12	37.27
MAP-Elites 50k	333.54	2.21	2059.32	47.27
Sobol 50k	241.93	5.21	1341.07	34.21

points in the 25x25 case. BOP-Elites PM solutions underperform compared to the observed solutions. This is the result of invalid regions in the region prediction step leading to misclassification, which is due to the effect of invalid, unobservable, regions on model building. As this domain closely models a real-world optimisation task, these results provide an important confirmation of BOP-Elites' efficacy and ability to extend to problems with invalidity.

B. Decoupled Descriptors

For problems with decoupled descriptors, we compare against the SAIL algorithm rather than SPHEN. BOP-Elites is given the same budget as in the coupled descriptor case but for the 10x10 and 25x25 problems it often converges to a good solution early. PMs in this case are created with known descriptor functions.

1) *Mishra Bird function - decoupled*: Table VIII shows summary statistics for the Mishra Bird function with decoupled descriptors. BOP-Elites finds a near optimal solution

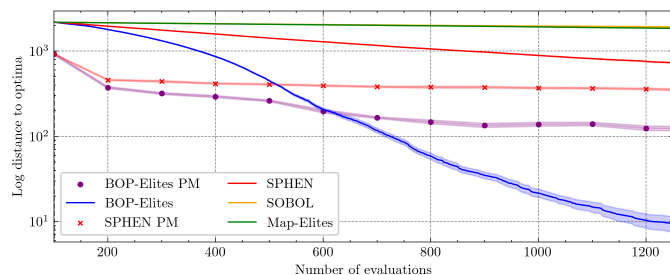


Fig. 7. Convergence of QDS for BOP-Elites in comparison to Map-Elites, SOBOL and SAIL on the PARSEC problem with 2 coupled descriptor functions, a $[25 \times 25]$ solution archive.

TABLE VIII
FINAL QDS ON THE MISHRA 2D FUNCTION WITH DECOUPLED DESCRIPTOR FUNCTIONS.

QDS: mishra problem	10x10		25x25	
	Mean	S.E..	Mean	S.E..
BOP-Elites	13778.69	0.04	80005.96	00.01
SAIL	13760.23	2.45	68587.44	51.98
BOP-Elites-PM	13778.68	0.01	79995.04	0.17
SAIL PM	13752.21	0.63	79561.09	1.21
MAP-Elites	11136.70	89.61	25473.58	196.45
MAP-Elites 50K	13666.59	1.63	79033.73	05.80
Sobol 50K	13710.03	1.08	79536.39	02.49

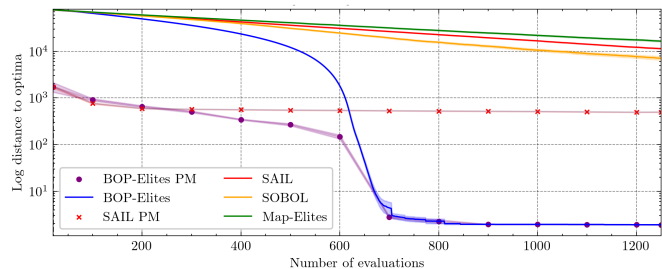


Fig. 8. Convergence of BOP-Elites in comparison to Map-Elites and SOBOL on the Mishra Bird function with 2 decoupled descriptor functions over a $[25 \times 25]$ solution archive

(comparably closer to the optimal than in the coupled case). Perhaps surprisingly SAIL does not appear to benefit much from knowledge of the descriptor functions, meaning the inability to keep up with BOP-Elites is likely related to SAILs point selection mechanism, which selects randomly from the acquisition map. BOP-Elites however produces a very different quality of PM which keeps step with the observed solution set as seen in Fig 8. This confirms that the drop in the coupled performance observed earlier is indeed due to errors in the descriptor models.

2) *Robotarm - decoupled*: In the decoupled descriptor case, BOP-Elites does not need to actively model the reachability of regions and this explains why we see such good performance. We do not see the slow down in performance as in the coupled case. As Table IX shows, in this domain the PMs perform slightly better than their fully evaluated counterparts and BOP-Elites performs best in all scenarios.

BOP-Elites once again achieves perfect coverage in this domain, while SAIL achieves 99% coverage on average with its observation set.

3) *PARSEC - decoupled*: Table X shows that BOP-Elites again outperforms the other algorithms. Unlike in the coupled case, the prediction maps for both BOP-Elites and SAIL perform well and rapidly achieve a result good QDS score in many regions. BOP-Elites attempts to make an average of 83 evaluations of invalid points, compared with 373 from the SAIL algorithm from a run of 1250 points in the decoupled case.

C. Upscaling Results

The following tables present results when performing 'Upscaling' by using the surrogate model from a completed BOP-

TABLE IX
FINAL QDS ON THE ROBOT ARM 4D FUNCTION WITH DECOUPLED DESCRIPTOR FUNCTIONS.

QDS: robotarm problem	10x10		25x25	
	Mean	S.E.	Mean	S.E.
BOP-Elites	85.17	0.001	504.30	0.020
SAIL	83.98	0.300	484.13	2.510
BOP-Elites-PM	85.17	0.001	505.10	0.005
SAIL - PM	84.42	0.216	499.39	0.120
MAP-Elites	50.27	0.690	140.10	1.310
MAP-Elites 50K	84.15	0.020	493.15	0.400
Sobol 50K	81.21	0.080	267.07	1.150

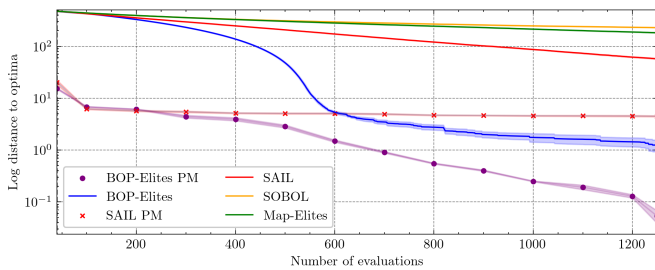


Fig. 9. Convergence of BOP-Elites in comparison to Map-Elites and SOBOL on the 4d Robot Planar arm problem with 2 decoupled descriptor functions and a $[25 \times 25]$ solution archive

TABLE X
FINAL QD SCORE ON THE PARSEC 10D DOMAIN WITH DECOUPLED DESCRIPTOR FUNCTIONS.

QD scores: PARSEC	10x10		25x25	
	Mean	S.E.	Mean	S.E.
BOP-Elites	371.01	0.21	2207.42	1.17
SAIL	361.21	1.63	1832.77	12.31
SAIL - PM	369.52	0.51	1875.06	1.79
BOP-Elites - PM	371.51	0.17	2198.79	1.30
MAP-Elites	205.77	1.21	293.12	37.27
MAP-Elites 50k	333.54	2.21	2059.32	47.27
Sobol 50k	241.93	5.21	1341.07	34.21

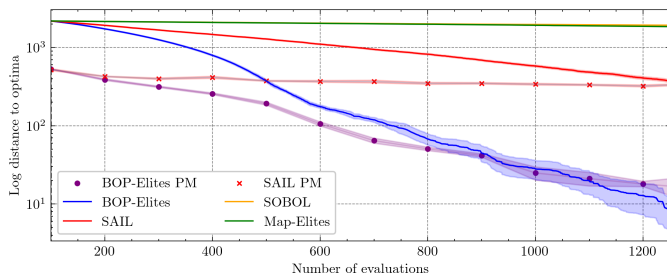


Fig. 10. Convergence of QDscores for BOP-Elites in comparison to Map-Elites, SOBOL and SAIL on the PARSEC problem with 2 white-box descriptor functions, a $[25 \times 25]$ solution archive.

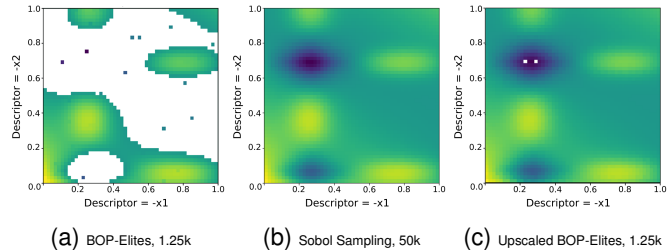


Fig. 11. Performance of a typical run of limited-budget BOP-Elites vs. Sobol sampling and upscaling on the Mishra's bird function.

Elites run optimising a 25×25 solution archive and predicting solutions for a 50×50 solution archive. All results in this section relate to coupled descriptor problems.

Recall that 'upscaling' is necessary because GP surrogate based methods cannot naively handle enough observations to fill a solution archive as large as 50×50 . However, the results in Table XI indicate that upscaling is a very effective method for extending surrogate based methods to high resolution problems. In each of the cases, both SPHEN and BOP-Elites provide competitive PMs, with BOP-Elites providing the best score of the PM methods. Interestingly, the Mishra Bird Function domain is again dominated by Sobol sampling methods for reasons discussed in the previous section.

Figure 11 compares a $[50 \times 50]$ solution archive from an indicative run of BOP-Elites, a Sobol Sampling run with a 50k budget and an upscaled BOP-Elites solution. Fig. 11(a) shows BOP-Elites is incapable of sampling each point of a 50×50 solution archive with its limited budget and therefore does not fill all the regions, but focuses on filling high performing areas first. In Fig. 11(b) Sobol sampling with 50k evaluations performs well in filling the regions with high performing solutions, and in Fig. 11(c) BOP-Elites Prediction Map upscaled from the 25×25 solution almost entirely fills the solution map trained on only 1.25k observations achieving 99% of the best observed solution archive and outperforming MAP-Elites with 50k observations. 2 missing regions appear in areas of low performance

TABLE XI
A COMPARISON OF PERFORMANCE ON MULTIPLE BENCHMARKS GIVEN A 50×50 SOLUTION ARCHIVE WITH THE ADDITION OF AN 'UPSCALED' 25×25 BOP-ELITES PM

QD score: 50x50	Mishra		Robot Arm		Rosenbrock 6d	
	Mean	S.E.	Mean	S.E.	Mean	S.E.
BOP-Elites	1.49e5	48.27	986.79	1.51	1.60e6	157.31
SAIL	1.30e5	771.27	897.66	10.07	1.18e6	1.31e4
SAIL - PM	3.07e5	22.71	1643.03	1.13	3.27e6	7.24e4
BOP-Elites - PM	3.05e5	38.99	1016.16	9.10	2.93e6	9.20e4
MAP-Elites	3.08e4	182.98	195.78	1.07	3.18e5	1.21e4
BOP-upscaled	3.08e5	51.99	1764.56	5.51	3.33e6	3.10e4
MAP-Elites 50k	3.02e5	152.1	1742.00	7.78	3.24e6	6.79e4
Sobol 50k	3.10e5	8.05	267.07	1.15	1.75e6	618.10

VII. CONCLUSION

In this work, we proposed BOP-Elites, a Bayesian Optimisation algorithm for quality diversity search, and formulated a

novel acquisition function, EJIE⁺. We show that BOP-Elites consistently and significantly outperforms MAP-Elites, SAIL and SPHEN in both the coupled and decoupled descriptor cases. We note that BOP-Elites is extremely sample efficient, requiring far fewer observations than QD methods that do not use surrogate models and outperforming all surrogate assisted methods at the same budget when using the QD-score to assess the solution archive. BOP-Elites consistently produces solution archives with objective performance approaching optimality where the search budget is sufficient to fill the solution archives - our experiments suggest a budget of twice the number of regions is sufficient in many cases - and additionally produces surrogate models that can be leveraged to make high quality Prediction Maps for solution archives with more regions, a process we call upscaling.

BOP-Elites is the first Bayesian Optimisation, rather than surrogate assisted, algorithm for QD problems and is able to model both the objective and descriptor values. Looking to the future, the BOP-Elites framework can be employed to test other acquisition functions such as Knowledge Gradient [26], Entropy Search [50], etc. BOP-Elites can be generalised to working on noisy problems by implementation of an adapted Noisy-Expected-Improvement [51], Knowledge Gradient, or other suitable acquisition functions.

BOP-Elites outperforms our comparison algorithms in all of the cases presented. These results include the PARSEC domain which exhibits invalid regions and the robot arm problem which features unreachable regions. BOP-Elites provides a rapid improvement to its solution archive in the early stages, filling the archive with high quality points drawn from high-performing regions of the search space. This behaviour encourages good coverage, filling empty regions, starting with the high performing regions first. This is likely to be a desirable trait of the algorithm, as in Fig. 9a, early stopping of the algorithm, or a smaller budget will still produce exploration of the high performing regions.

Our results show that BOP-Elites produces surrogate models that can ‘upscale’, i.e., surrogates built for a solution archive with few regions can be used to generate a solution archive with many regions. This may be an interesting area of further development if we were able to populate a coarse solution archive with few observations and then actively refine the solution archive in areas of interest, e.g., subdivide regions with high density or high performance, spending more observations in areas that prove the most useful.

In this work we use a uniform discretisation regime for the solution archive. However, when dealing with a higher number of behavioural dimensions, uniform discretisation typically leads to an impractical increase in the number of distinct regions. To mitigate this, Centroidal Voronoi Tessellation (CVT) [4] is often used to limit the number of regions. As a potential avenue for further investigation, BOP-Elites could use a CVT archive, and a Monte Carlo estimator could be implemented to estimate the probability of a solution belonging to a given region. Additionally, we assumed independence of the descriptor functions. Future work could explore whether modelling the descriptors with a joint GP and numerically integrating the joint distribution would yield

better performance when the behaviours are highly correlated.

BOP-Elites slows down computationally in the prediction step when the Gaussian Process is built with many points, but this could be mitigated by the use of sparse GP techniques or alternative surrogate models. Gaussian Processes are generally considered to be ill-suited to high dimensional problems (beyond 20 dimensions) but BOP-Elites could also utilise other surrogate models to extend to much higher dimensions.

In addition to its strong performance, we consider BOP-Elites a proof of concept for continuing exploration of Bayesian Optimisation for QD problems and hope this work shows that, while QD research tends to come from the field of evolutionary computation, it is an interesting and rich area of future research in future applications for surrogate based methods and Bayesian Optimisation in particular.

ACKNOWLEDGEMENTS

Paul Kent would like to acknowledge funding from EPSRC through grant EP/L015374/1. Jean-Baptiste Mouret has received funding from the European Union’s Horizon Europe Framework Programme under grant agreement No 101070596 and from the ANR in the France 2030 program through project PEPR O2R AS3 (ANR-22-EXOD-007).

REFERENCES

- [1] J.-B. Mouret and J. Clune, “Illuminating search spaces by mapping elites,” *arXiv preprint arXiv:1504.04909*, 2015.
- [2] Y. Jiang, D. Salley, A. Sharma, G. Keenan, M. Mullin, and L. Cronin, “An artificial intelligence enabled chemical synthesis robot for exploration and optimization of nanomaterials,” *Science Advances*, vol. 8, no. 40, p. eabo2626, 2022.
- [3] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, “Robots that can adapt like animals,” *Nature*, vol. 521, no. 7553, pp. 503–507, 2015.
- [4] V. Vassiliades, K. Chatzilygeroudis, and J.-B. Mouret, “Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 623–630, 2017.
- [5] A. Cully and J.-B. Mouret, “Behavioral repertoire learning in robotics,” in *Genetic and Evolutionary Computation Conference*, 2013, pp. 175–182.
- [6] J. K. Pugh, L. B. Soros, and K. O. Stanley, “Quality diversity: A new frontier for evolutionary computation,” *Frontiers in Robotics and AI*, vol. 3, p. 40, 2016.
- [7] A. Gaier, A. Asteroth, and J.-B. Mouret, “Data-efficient design exploration through surrogate-assisted illumination,” *Evolutionary Computation*, vol. 26, no. 3, pp. 381–410, 2018.
- [8] K. Chatzilygeroudis, V. Vassiliades, F. Stulp, S. Calinon, and J.-B. Mouret, “A survey on policy search algorithms for learning robot controllers in a handful of trials,” *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 328–347, 2019.
- [9] A. Hagg, D. Wilde, A. Asteroth, and T. Bäck, “Designing air flow with surrogate-assisted phenotypic niching,” in *International Conference on Parallel Problem Solving from Nature*. Springer, 2020, pp. 140–153.
- [10] P. Kent and J. Branke, “Bop-elites, a bayesian optimisation algorithm for quality-diversity search,” *arXiv preprint arXiv:2005.04320*, 2020.
- [11] D. R. Jones, M. Schonlau, and W. J. Welch, “Efficient global optimization of expensive black-box functions,” *Journal of Global optimization*, vol. 13, no. 4, p. 455, 1998.
- [12] K. Chatzilygeroudis, A. Cully, V. Vassiliades, and J.-B. Mouret, “Quality-diversity optimization: a novel branch of stochastic optimization,” in *Black Box Optimization, Machine Learning, and No-Free Lunch Theorems*. Springer, 2021, pp. 109–135.
- [13] R. K. Ursem, “Diversity-guided evolutionary algorithms,” in *International Conference on Parallel Problem Solving from Nature*. Springer, 2002, pp. 462–471.
- [14] J. Lehman and K. O. Stanley, “Evolving a diversity of virtual creatures through novelty search and local competition,” in *Genetic and Evolutionary Computation Conference*, 2011, pp. 211–218.

- [15] M. C. Fontaine, R. Liu, J. Togelius, A. K. Hoover, and S. Nikolaidis, "Illuminating mario scenes in the latent space of a generative adversarial network," in *AAAI Conference on Artificial Intelligence*, 2021.
- [16] T. Galanos, A. Liapis, G. N. Yannakakis, and R. Koenig, "Arch-elites: Quality-diversity for urban design," *arXiv preprint arXiv:2104.08774*, 2021.
- [17] S. Fioravanzo and G. Iacca, "Map-elites for constrained optimization," *Constraint Handling in Metaheuristics and Applications*, p. 151, 2021.
- [18] O. Nilsson and A. Cully, "Policy gradient assisted map-elites," in *Genetic and Evolutionary Computation Conference*, 2021.
- [19] G. Paolo, A. Coninx, S. Doncieux, and A. Laflaquière, "Sparse reward exploration via novelty search and emitters," in *Genetic and Evolutionary Computation Conference*, 2021.
- [20] J. Nordmoen, F. Veenstra, K. O. Ellefsen, and K. Glette, "Map-elites enables powerful stepping stones and diversity for modular robotics," *Frontiers in Robotics and AI*, vol. 8, 2021.
- [21] M. Fontaine and S. Nikolaidis, "Differentiable quality diversity," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10040–10052, 2021.
- [22] P. I. Frazier, "A tutorial on bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.
- [23] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*. MIT Press, 2005.
- [24] H. J. Kushner, "A new method of locating the maximum point of an arbitrary multiplex curve in the presence of noise," *Journal of Fluids Engineering*, vol. 86, no. 1, pp. 97–106, 1964.
- [25] J. Mockus, V. Tiesis, and A. Zilinskas, "The application of bayesian methods for seeking the extremum," *Towards Global Optimization*, vol. 2, no. 117-129, p. 2, 1978.
- [26] W. Scott, P. Frazier, and W. Powell, "The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 996–1026, 2011.
- [27] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv:0912.3995*, 2009.
- [28] Y. Zhang, M. C. Fontaine, A. K. Hoover, and S. Nikolaidis, "Deep surrogate assisted map-elites for automated hearthstone deckbuilding," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2022, pp. 158–167.
- [29] —, "Dsa-me: Deep surrogate assisted map-elites," in *ICLR Workshop on Agent Learning in Open-Endedness*, 2022.
- [30] A. Gaier, A. Asteroth, and J.-B. Mouret, "Data-efficient exploration, optimization, and modeling of diverse designs through surrogate-assisted illumination," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2017, pp. 99–106.
- [31] —, "Aerodynamic design exploration through surrogate-assisted illumination," in *18th AAAA/ISSMO multidisciplinary analysis and optimization conference*, 2017, p. 3330.
- [32] A. Hagg, A. Asteroth, and T. Bäck, "Prototype discovery using quality-diversity," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2018, pp. 500–511.
- [33] T. Sen, Y. Kawajiri, and M. J. Realff, "Integration of material and process design for kinetic adsorption separation," *Industrial & Engineering Chemistry Research*, vol. 60, no. 6, pp. 2536–2546, 2021.
- [34] V. Bhatt, B. Tjanaka, M. Fontaine, and S. Nikolaidis, "Deep surrogate assisted generation of environments," *Advances in Neural Information Processing Systems*, vol. 35, pp. 37762–37777, 2022.
- [35] V. Bhatt, H. Nemlekar, M. Fontaine, B. Tjanaka, H. Zhang, Y.-C. Hsu, and S. Nikolaidis, "Surrogate assisted generation of human-robot interaction scenarios," *arXiv preprint arXiv:2304.13787*, 2023.
- [36] Y. Zhang, M. C. Fontaine, V. Bhatt, S. Nikolaidis, and J. Li, "Multi-robot coordination and layout design for automated warehousing," *arXiv preprint arXiv:2305.06436*, 2023.
- [37] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.
- [38] G. De Ath, R. M. Everson, A. A. Rahat, and J. E. Fieldsend, "Greed is good: Exploration and exploitation trade-offs in bayesian optimisation," *ACM Transactions on Evolutionary Learning and Optimization*, vol. 1, no. 1, pp. 1–22, 2021.
- [39] I. M. Sobol, "Uniformly distributed sequences with an additional uniform property," *USSR Computational Mathematics and Mathematical Physics*, vol. 16, no. 5, pp. 236–242, 1976.
- [40] H. Chi, P. Beerli, D. W. Evans, and M. Mascagni, "On the scrambled sobol sequence," in *International conference on computational science*. Springer, 2005, pp. 775–782.
- [41] J. R. Gardner, M. J. Kusner, Z. E. Xu, K. Q. Weinberger, and J. P. Cunningham, "Bayesian optimization with inequality constraints," in *ICML*, vol. 2014, 2014, pp. 937–945.
- [42] R. Hooke and T. A. Jeeves, "'direct search' solution of numerical and statistical problems," *Journal of the ACM (JACM)*, vol. 8, no. 2, pp. 212–229, 1961.
- [43] D. Sha, K. Ozbay, and Y. Ding, "Applying bayesian optimization for calibration of transportation simulation models," *Transportation Research Record*, vol. 2674, no. 10, pp. 215–228, 2020.
- [44] Y. Mohsenzadeh and H. Sheikhzadeh, "Gaussian kernel width optimization for sparse bayesian learning," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 4, pp. 709–719, 2014.
- [45] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.
- [46] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy, "BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization," in *Advances in Neural Information Processing Systems 33*, 2020. [Online]. Available: <http://arxiv.org/abs/1910.06403>
- [47] S. K. Mishra, "Some new test functions for global optimization and performance of repulsive particle swarm method," *Available at SSRN 926132*, 2006.
- [48] H. Rosenbrock, "An automatic method for finding the greatest or least value of a function," *The computer journal*, vol. 3, no. 3, pp. 175–184, 1960.
- [49] H. Sobieczky, "Parametric airfoils and wings," in *Recent development of aerodynamic design methodologies*. Springer, 1999, pp. 71–87.
- [50] P. Hennig and C. J. Schuler, "Entropy search for information-efficient global optimization," *Journal of Machine Learning Research*, vol. 13, no. 6, 2012.
- [51] B. Letham, B. Karrer, G. Ottoni, and E. Bakshy, "Constrained bayesian optimization with noisy experiments," *Bayesian Analysis*, vol. 14, no. 2, 2019.
- [52] M. Drela, "Xfoil: An analysis and design system for low reynolds number airfoils," in *Low Reynolds Number Aerodynamics: Proceedings of the Conference Notre Dame, Indiana, USA, 5–7 June 1989*. Springer, 1989, pp. 1–12.
- [53] A. Gaier, A. Asteroth, and J.-B. Mouret, "Data-efficient design exploration through surrogate-assisted illumination," *Evolutionary computation*, vol. 26, no. 3, pp. 381–410, 2018.
- [54] J. Blank and K. Deb, "pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020.

APPENDIX

A. Test problems

Below we provide details on the various test problems used in our study.

1) *Synthetic Gaussian Process problems*: We generate a single output, 10 dimensional input, Gaussian Process model with a predefined lengthscale and take one realisation of the posterior Gaussian Process as a deterministic objective function for testing. We are able to generate both objective and one or more descriptor functions this way, combining them to create multidimensional descriptor spaces.

The benefit of generating functions from GPs is that we can create an infinity of unique functions quickly and efficiently without concern for model mismatch with the GP surrogate model. Lengthscales are chosen uniformly at random from 0.1 to 1 and algorithms are compared on functions generated from the same seeds.

2) *Mishras Bird function*: Mishras Bird function is an optimisation benchmark [47] with the following analytical form :]]]

$$\begin{aligned} f(x_1, x_2) &= A + B + C, \\ A &= \sin x_2 \exp(1 - \cos x_1^2), \\ B &= \cos x_1 \exp(1 - \sin x_2^2), \\ C &= (x_1 - x_2)^2, \\ &\text{with } -10 \leq x_1 \leq 0, -6.5 \leq x_2 \leq 0. \end{aligned}$$

We define 2 simple descriptor functions with Mishras bird function:

$$B_1(x) = -x_1, B_2(x) = -x_2$$

Whilst being low dimensional, the problem exhibits an interesting property as there is a small region that vastly outperforms the rest of the domain in terms of function value.

3) *Robot Planar Arm*: This well-known QD benchmark problem [3] is a simulation of a planar robot arm with an $n = 4$ dimensional unit hypercube $[0, 1]^4$ as the input and objective and descriptor values as follows:

$$y = 1 - \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (19)$$

$$B_1 = \frac{1}{2n} \sum_{i=1}^n \sin \left(\sum_{j=1}^i (2\pi x_j - \pi) \right) + 0.5 \quad (20)$$

$$B_2 = \frac{1}{2n} \sum_{i=1}^n \cos \left(\sum_{j=1}^i (2\pi x_j - \pi) \right) + 0.5 \quad (21)$$

where \bar{x} is the mean of x , $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$, the behaviour space is known to be $[B_1, B_2] \in [0, 1]^2$ and the objective function has values in the range $y \in [0, 1]$

This problem simulates the ability of a robot arm to reach feasible regions in a $2d$ plane while minimising the standard deviation of the position of the individual joints.

4) *Rosenbrock 6d*: The Rosenbrock function [48] is defined for even numbered dimensions, here we use 6 dimensions.

$$\begin{aligned} f(x) &= \sum_{i=1,3,5} 100 [(2x_i - 2x_{i+1}^2)^2 + (1 - 2x_{i+1}^2)] \\ &0 \leq x_i \leq 1, \forall i. \end{aligned}$$

We used 2 feature dimensions with the form

$$B_1(x) = \frac{1}{2}(x_1 + x_2), B_2(x) = (x_3 - 1)^2$$

Rosenbrock is a relatively smooth function but in 6 dimensions.

5) *PARSEC: 2D Aerofoil optimisation problem*: A 2D aerofoil design was modelled with the Xfoil simulation package [52] and the objective and features were derived from both the physical characteristics of the aerofoil and the simulation results. This is a well known and well studied problem in engineering and we used the 10 dimensional PARSEC representation of the parameters [49]. This problem provides an informative baseline as it forms a key evaluation from the

previous SAIL papers. The objective is calculated with penalty values ρ as:

$$\text{objective}(x) = (\mu_{drag}(x) + K\mu_{drag}(x)) \times \rho_{lift}(x) \times \rho_{area}(x)$$

and the descriptors are values taken from the parameterised representation called X_{up} and Z_{up} . For specifics on these values we guide readers to the SAIL literature [53].

B. Robustness study for the cutoff value

Although it is clear that the cutoff value approach is useful for problems with unreachable regions (such as the Robot Arm benchmark), we explore if this method has a detrimental effect on the performance of BOP-Elites when optimising other problems. Presented in Table XII is the average absolute difference between performance with and without the cutoff. As can be seen, a significant difference only emerges in the Robot Arm case, implying that this method is generalisable for use in cases without unreachable regions with no detrimental effect.

TABLE XII
DIFFERENCE IN AVERAGE PERFORMANCE WITH AND WITHOUT THE CUTOFF METHOD, CELLS IN GRAY INDICATE SIGNIFICANT DIFFERENCE AT THE 99% LEVEL USING A T-TEST

Average absolute difference QD scores:	10x10	25x25
Mishra Bird Function	0.02	1.31
Robot Arm	4.04	29.01
Rosenbrock 6D	0.09	1.93
Synthetic function 10D	0.00	0.00
PARSEC	0.03	1.41

C. Exploring the impact of early upscaling

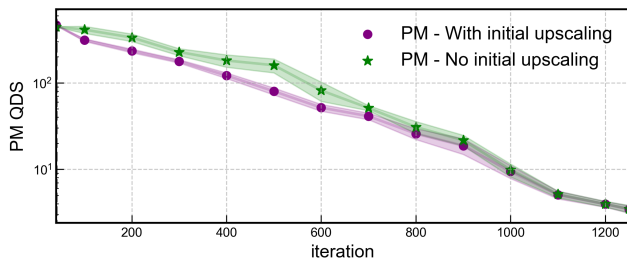
The performance of PMs serves as an indicator of model accuracy, and hence, BOP-Elites's predictions are contingent upon it. We performed a comparative analysis of the PMs' true objective performance, with and without early upscaling in the coupled descriptor case.

In all of our experiments, early upscaling improved, or had no negative effect on the PM. Fig. 12 demonstrates the efficacy of early upscaling in enhancing the surrogate models in BOP-Elites in 3 cases. We see an improvement to early PM QDS values. This can be attributed to the improvement in descriptor predictions due to exploration. This reduces the probability of mispredictions, which incur a zero QDS contribution.

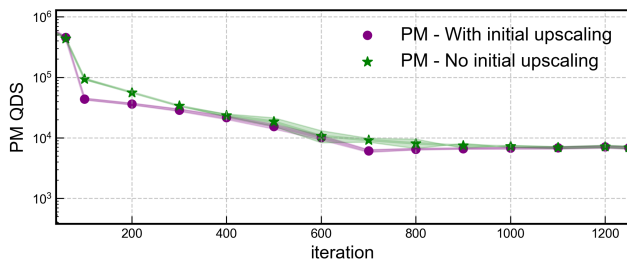
This improvement is sustained far longer in the Robot Arm domain, as illustrated in Fig. 12a. The Robot Arm domain has an interesting symmetrical property, the robot arm can reach high performing regions from both extremes in parameter space $(0, 0, 0, 0)$ and $(1, 1, 1, 1)$. Both of these points give identical descriptor values, corresponding to the arm's endpoint position, but they have been achieved by rotating the arm from opposite sides.

Our observations indicate that without early exploration, BOP-Elites can, with adversarial initial conditions, become overly focused on exploiting one high performing region, such as one side of the symmetry in the case of the RobotArm,

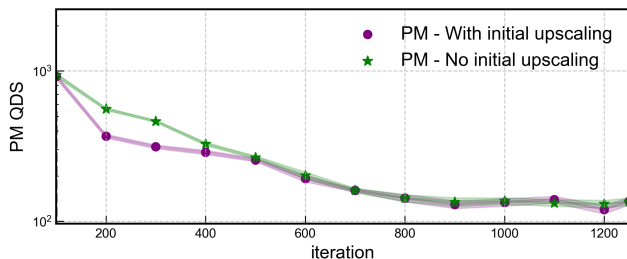
without sufficient exploration. In rare failure cases this *tunnel-vision* can lead to many iterations being spent without recognising the potential of an opposite, or hard to find, high-performing region. In our results, the algorithm is able to recover within the allowed budget, falling in line with the better performing version with initial upscaling. However, if the algorithm were stopped early, or a smaller budget given, it would be desirable to have a good balance between predictive model quality and objective performance across a diverse space. This forms a strong justification for the initial upscaling.



(a) Robot Arm problem



(b) Rosenbrock problem



(c) Parsec problem

Fig. 12. Comparison of Prediction Map performance with and without upscaling for various problems

D. Invalidity modelling

In this paper we implement an SVM model to predict the validity of a point in the search domain. The acquisition function becomes $EJIE^{++}$, actively modelling invalidity, only when an invalid point is found during an attempted evaluation. The PARSEC domain is the only benchmark we considered that features this invalidity.

Table XIII shows a comparison of the average QDS, the number of successful evaluations made n_e , and the number of invalid points sampled n_i , per run. A value of ∞ implies that the algorithm attempts to sample at an invalid point continuously, ending the run as a failure case.

TABLE XIII
MEAN PERFORMANCE WITH AND WITHOUT INVALIDITY MODELLING FOR THE PARSEC DOMAIN WITH COUPLED DESCRIPTORS.

	10x10			25x25		
	QDS	n_e	n_i	QDS	n_e	n_i
BOP-Elites ⁺⁺	370.09	1000	31	2206.10	1250	94
BOP-Elites ⁺	196.20	237	∞	1431.05	344	∞
SPHEN	361.21	1000	125	1484.73	1250	362

SPHEN, like SAIL, builds an acquisition map using the UCB acquisition function to find valuable points and then randomly samples points from this. The points in the acquisition map are well-distributed in behaviour space and SAIL has only a vanishing probability to select the same point twice in sequential iterations. BOP-Elites, by contrast, always chooses the point that is maximal under the $EJIE$ acquisition function. This means that if there arises no disincentive to select a point, it will continue to do so, leading to a complete failure of that algorithm, which we see here.

E. Complexity and runtime of BOP-Elites

Our BOP-Elites implementation uses the BOTORCH [46] package in Python, and does not use GPUs. The posterior variance prediction step is the most significant contributor to the overall computational complexity in the BOP-Elites algorithm, as the complexity grows as $\mathcal{O}(n^2)$, where n is the number of data points collected so far. The quadratic scaling arises from the need for kernel matrix multiplication, a process that is inherently computationally intensive as the number of samples increases. The posterior mean prediction step contributes $\mathcal{O}(n)$.

In the initialisation of an expensive coupled problem, we find diverse descriptor values of points in X by rejection sampling over a Sobol set sampled from the descriptor GPs. In order to achieve good diversity and limit the computational cost, this number is set to a maximum of 10,000 posterior samples, where the first $k = 5$ points in each region are kept, and the process stops when each region has k points. As BOTORCH is designed to benefit from batch sampling this step can be computed efficiently.

Pattern search is given a budget of 100 iterations. In each iteration it explores each input dimension. Therefore, each restart has a maximum number of calls to the GPs of $100d(m+1)$, to the m descriptor GPs and the GP for the fitness, with a maximum complexity of $\mathcal{O}(100d(m+1)(n+n^2))$ accounting for mean and variance predictions.

In the case of invalidity modelling with an SVM, the current algorithm implements an RBF kernel and therefore, in the prediction step, has a computational complexity of $s \times d$ for the number of support vectors identified s and input dimensions d .

Fig. 13 shows the actual wall clock time of the optimisation of the acquisition function for an indicative BOP-Elites run on 4 10x10 coupled benchmark problems. We can see that the observed time grows quadratically with the number of samples so far.

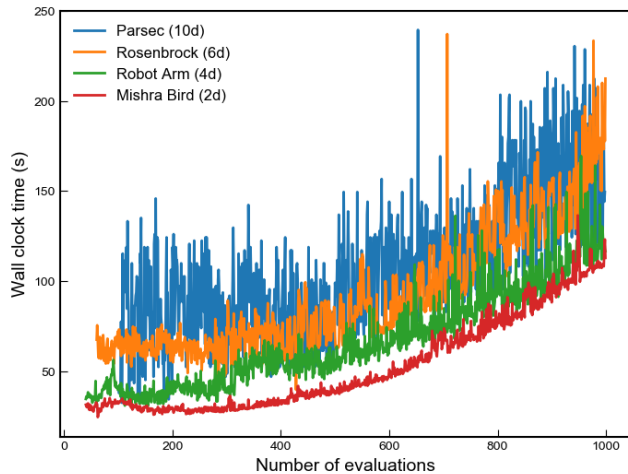


Fig. 13. Wall clock performance of an indicative run of the BOP-Elites acquisition function on 4 coupled benchmarks, performed on a 3.6GHz CPU with 62GB RAM.

While the raw wall clock time may be higher than other QD algorithms, BOP-Elites is designed for applications where evaluations are expensive. In domains where the actual objective function evaluations are costly (e.g., physical experiments,

expensive simulations), the time spent on model predictions is dwarfed by the real cost of the objective. Therefore the dominant factor in the overall optimisation time is the function evaluations themselves, not the computational overhead, or wall clock performance of the algorithm.

Furthermore, the implementation of BOP-Elites in this paper leaves room for several possible improvements with regard to wall clock performance.

- The pymoo [54] implementation of pattern search is sequential and cannot make use of parallel objective evaluations. Parallelising this step would cut run time by a factor of d .
- Our code does not currently run restarts in parallel, however, restarts would be straightforward to parallelise.
- While algorithms like SAIL select batches of points to evaluate each iteration, BOP-Elites uses a sequential point selection process, optimising several points but selecting only one to evaluate. Given BOP-Elites performs multiple restarts, it may be viable to select several points at once if the result from the restarts are sufficiently high performing and diverse from each other.
- Our implementation does not make use of the acceleration available through GPU's and this may reduce runtime.