Multi-Objective Bayesian Optimization over High-Dimensional Search Spaces (Supplementary Material)

Samuel Daulton[*,1,2] **David Eriksson**[*,2] **Maximillian Balandat**[2] **Eytan Bakshy**[2]

[*]Equal contribution
[1]University of Oxford, Oxford, UK
[2]Meta, Menlo Park, USA

A DETAILS ON BATCH SELECTION

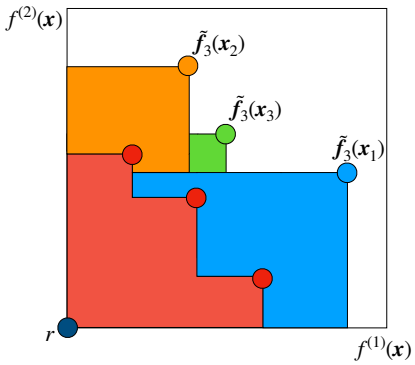

Figure 1: A visualization of our batch selection using HVI with $q = 4$. The red points represent the current PF. Blue, orange, and green points show the function values for the 3 selected points under the next posterior sample. To select the 4th point, the HVI of each candidate is evaluated jointly with the red, blue, orange, and green points.

As discussed in Section 2, over-exploration can be an issue in high-dimensional BO because there is typically high uncertainty on the boundary of the search space, which often results in over-exploration. This is particularly problematic when using continuous optimization routines to find the maximizer of the acquisition function since the global optimum of the acquisition function will often be on the boundary, see Oh et al. [2018] for a discussion on the "boundary issue" in BO. While the use of trust regions alleviates this issue, this boundary issue can still be problematic, especially when the trust regions are large.

To mitigate this issue of over-exploration, we use a discrete set of candidates by perturbing randomly sampled Pareto optimal points within a trust region by replacing only a small subset of the dimensions with quasi-random values from a scrambled Sobol sequence. This is similar to the approach used by Eriksson and Poloczek [2021] which proved crucial for good performance on high-dimensional problems. In addition, we also decrease the perturbation probability p_n as the optimization progresses, which Regis and Shoemaker [2013] found to improve optimization performance. The perturbation probability p_n is set according to the following schedule:

$$p_n = p_0 \left[1 - 0.5 \frac{\log n'}{\log b} \right],$$

where n_0 is the number of initial points, n_f is the total evaluation budget, $p_0 = \min\{\frac{20}{d}, 1\}$, $b = n_f - n_0$, and $n' = \min\{\max\{n - n_0, 1\}, b\}$.

Given a discrete set of candidates, MORBO draws samples from the joint posterior over the function values for the candidates in this set and the previously selected candidates in the current batch, and selects the candidate with maximum HVI across

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).

the joint samples. This procedure is repeated to build the entire batch.[1] Using standard Cholesky-based approaches, exact posterior sampling has complexity that is cubic with respect to the number of test points and therefore is only feasible for relatively small discrete sets.

A.1 RFFS FOR FAST POSTERIOR SAMPLING

While asymptotically faster approximations than exact sampling exist; see Pleiss et al. [2020] for a comprehensive review, these methods still limit the candidate set to be of modest size (albeit larger), which may not do an adequate job of covering a the entire input space. Among the alternatives to exact posterior sampling, we consider using Random Fourier Features (RFFs) [Rahimi and Recht, 2007], which provide a deterministic approximation of a GP function sample as a linear combination of Fourier basis functions. This approach has empirically been shown to perform well with Thompson sampling for multi-objective optimization [Bradford et al., 2018]. The RFF samples are cheap to evaluate and which enables using much larger discrete sets of candidates since the joint posterior over the discrete set does not need to be computed. Furthermore, the RFF samples are differentiable with respect to the new candidate $\boldsymbol{x}$, and HVI is differentiable with respect to $\boldsymbol{x}$ using cached box decompositions [Daulton et al., 2021], so we can use second-order gradient optimization methods to maximize HVI under the RFF samples.

We tried to optimize these RFF samples using a gradient based optimizer, but found that many parameters ended up on the boundary, which led to over-exploration and poor BO performance. In an attempt to address this over-exploration issue, we instead consider continuous optimization over axis-aligned subspaces which is a continuous analogue of the discrete perturbation procedure described in the previous section. Specifically, we generate a discrete set of candidates points by perturbing random subsets of dimensions according to p_n, as in the exact sampling case. Then, we take the top 5 initial points with the maximum HVI under the RFF sample. For each of these best initial points we optimize only over the perturbed dimensions using a gradient based optimizer.

Figure 2 shows that the RFF approximation with continuous optimization over axis-aligned subspaces works well on for $D = 10$ on the DTLZ2 function, but the performance degrades as the dimensionality increases. Thus, the performance of MORBO can likely be improved on low-dimensional problems by using continuous optimization; we used exact sampling on a discrete set for all experiments in the paper for consistency. We also see that as the dimensionality increases, using RFFs over a discrete set achieves better performance than using continuous optimization. In high-dimensional search spaces, we find that exact posterior sampling over a discrete set achieves better performance than using RFFs, which we hypothesize is due to the quality of the RFF approximations degrading in higher dimensions. Indeed, as shown in Figure 2, optimization performance using RFFs improves if we use more basis functions on higher dimensional problems (4096 works better than 1024).

B ADDITIONAL DETAILS OF CONSTRAINT HANDLING IN MORBO

If there are feasible points, the center is selected as the point with maximum HVC across the feasible Pareto frontier. If there are no feasible points, the center is selected to be the point with minimum total constraint violation (the sum of the constraint violations). A TR's success counter is incremented if the TR center was feasible and the candidates generated from this TR improved the feasible hypervolume or if the TR center was infeasible and a candidate generated from this TR has lower total constraint violation than the TR center.

C PROOFS

Lemma 4.1. *Let $\boldsymbol{f} \in [0, B]^M$, and assume that MORBO only considers a newly evaluated sample to be an improvement (for updating the corresponding TR's success and failure counters) if it increases the HV by at least $\delta \in \mathbb{R}^+$ and assume that success counter threshold $\tau_{succ} = \infty$.[2] Then each TR will only evaluate a finite number of samples.*

Proof. First, note that The hypervolume of the true Pareto frontier $\mathcal{P}^*$ is bounded. Without loss of generality, if the reference point $\boldsymbol{r} = \mathbf{0}$, then the $\text{HV}(\mathcal{P}^*) \leq B^M$. Suppose that a trust region evaluates an infinite number of samples. Then, the trust

[1] In the case that the candidate point does not satisfy that satisfy all outcome constraints under the sampled GP function, the acquisition value is set to be the negative constraint violation.

[2] As stated in Appendix D, we use $\tau_{\text{succ}} = \infty$ in all of our experiments.

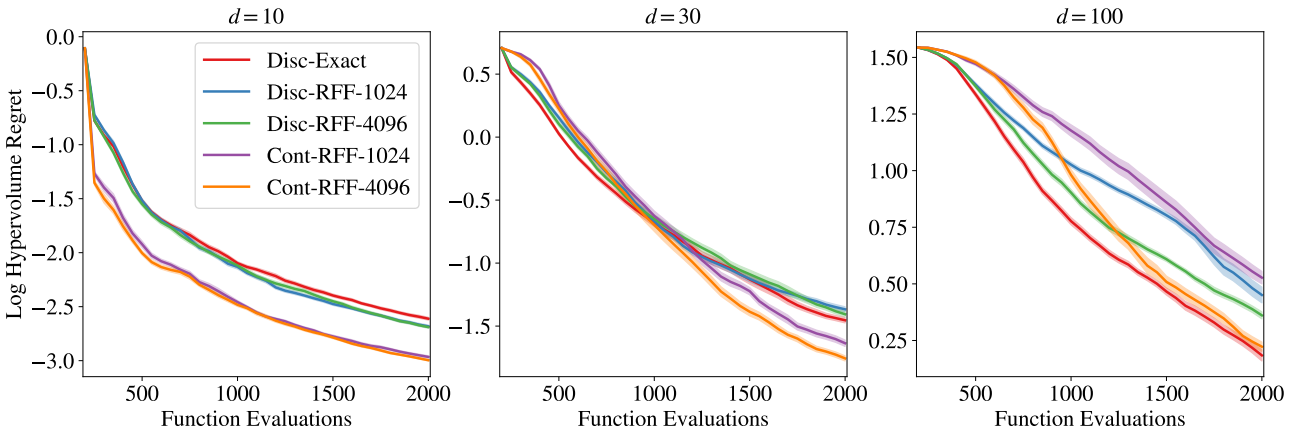

Figure 2: Optimization performance under various Thompson sampling approaches on DTLZ2 test problems with 2 objectives and various input dimensions $d \in \{10, 30, 100\}$. Disc-Exact uses exact samples from the joint posterior over a discrete set of 4096 points. Disc-RFF-1024 and Disc-RFF-4096 evaluate approximate sample paths (RFFs) over a discrete set of 4096 points with 1024 and 4096 basis functions, respectively. Cont-RFF-1024 and Cont-RFF-4096 use L-BFGS-B with exact gradients to optimize RFF draws along a subset of the dimensions (see in Appendix A.1 for details) using 1024 and 4096 basis functions, respectively.

region has not had $1 + \log_2 L_{\text{init}} - \log_2 L_{\text{min}}$ streaks of τ_{fail} consecutive failures. Hence, the trust region has increased the hypervolume of the Pareto frontier over the previously evaluated designs by at least δ infinitely many times. Hence, the hypervolume over the previously evaluated designs is infinite. This is a contradiction. □

Theorem 4.1. *Let $\boldsymbol{f} \in [0, B]^M$ for $B > 0$ and let each component $f^{(m)}$ for $m = 1, ..., M$ follow a Gaussian distribution with marginal variances $\sigma \leq 1$ and independent observation noise $\epsilon_m \sim \mathcal{N}(0, \sigma_m^2)$ such that $\sigma_m^2 \leq \sigma^2 \leq 1$. Let $\mathcal{P}_t$ denote the Pareto frontier over $\boldsymbol{f}(X_t)$, where X_t is the set of TR re-initialization points after t TRs have been restarted. Suppose further that the conditions of Lemma 4.1 hold. Then, the cumulative hypervolume regret R_T of MORBO after T restarts is bounded by:*

$$R_T \leq M^2 (\sqrt{2e\pi} B/2)^M \sqrt{d \gamma_T T \ln(T)}.$$

Proof. From Lemma 4.1, we have that each trust region will only evaluate a finite number of samples. Hence, as the number of evaluations goes to infinity, MORBO will terminate and select new initial center points for trust regions an infinite number of times. Our regret bound is in terms of the number of restart points.

Our proof follows that of Zhang and Golovin [2020, Theorem 8], but the final form of our bound holds for arbitrary B. Note that lines 13-19 in Algorithm 1 correspond to Paria et al. [2020, Algorithm 1] using Thompson sampling, where the only evaluations are the $t - 1$ restart points. From Paria et al. [2020, Theorem 1], the scalarized Bayes regret of Paria et al. [2020, Algorithm 1] using L–Lipschitz scalarizations is $O\big(LMd^{\frac{1}{2}}[\gamma_T T \ln(T)]^{\frac{1}{2}}\big)$. Since a hypervolume scalarization $s_{\boldsymbol{\lambda}}[\boldsymbol{y}]$ is $\mathcal{O}(B^M M^{1+M/2})$–Lipschitz [Zhang and Golovin, 2020, Lemma 6], we have that $L \leq B^M M^{1+M/2}$. From Zhang and Golovin [2020, Proof of Theorem 8], the hypervolume regret can be expressed by scaling the scalarized Bayes regret by a constant $c_M = \frac{\pi^{\frac{M}{2}}}{2^M \Gamma(\frac{M}{2}+1)}$ that depends on the number of objectives. Hence, we can bound the hypervolume regret as:

$$R_T = \sum_{t=1}^{T} \text{HV}(\mathcal{P}^*) - \text{HV}(\mathcal{P}_t) \leq c_M LMd^{\frac{1}{2}}[\gamma_T T \ln(T)]^{\frac{1}{2}}.$$

Note that

$$c_M L \leq B^M M^{1+M/2} \frac{\pi^{\frac{M}{2}}}{2^M \Gamma(\frac{M}{2}+1)}$$

From Li and Chen [2007, Theorem 1], $\Gamma(x) > \frac{x^{x-\gamma}}{e^{x-1}}$, where $\gamma \approx 0.577$ is the Euler-Mascheroni constant. So,

$$\Gamma(M/2+1) > \frac{(M/2+1)^{(M/2+1-\gamma)}}{e^{(M/2)}} > \frac{M^{(M/2)}}{2e^{(M/2)}}.$$

Hence,

$$\frac{1}{\Gamma(\frac{M}{2}+1)} < \frac{(2e)^{(M/2)}}{M^{(M/2)}}.$$

So,

$$\begin{aligned} c_M L &\leq B^M M^{1+M/2} \frac{\pi^{\frac{M}{2}}}{2^M \Gamma(\frac{M}{2}+1)} \\ &\leq B^M M \frac{(2e\pi)^{\frac{M}{2}}}{2^M} \\ &\leq M\big(\sqrt{2e\pi}B/2\big)^M. \end{aligned}$$

So the cumulative regret bound is

$$\begin{aligned} R_T &\leq c_M L M d^{\frac{1}{2}} [\gamma_T T \ln(T)]^{\frac{1}{2}} \\ &\leq M^2 (\sqrt{2e\pi}B/2)^M d^{\frac{1}{2}} [\gamma_T T \ln(T)]^{\frac{1}{2}}. \end{aligned}$$

□

D DETAILS ON EXPERIMENTS

D.1 ALGORITHMIC DETAILS

For MORBO, we use 5 trust regions, which we observed was a robust choice in Figure 4. Following [Eriksson et al., 2019], we set $L_{\text{init}} = 0.8$, $L_{\max} = 1.6$, and use a minimum length of $L_{\min} = 0.01$. We use 4096 discrete points for optimizing HVI for the vehicle safety and welded beam problems, 2048 discrete points on the trajectory planning and optical design problems, and 512 discrete points on the Mazda problem. Note that while the number of discrete points should ideally be chosen as large as possible, it offers a way to control the computational overhead of MORBO; we used a smaller value for the Mazda problem due to the fact that we need to sample from a total of 56 GP models in each trust region as there are 54 black-box constraints. We use an independent GP with a a constant mean function and a Matérn-$5/2$ kernel with automatic relevance detection (ARD) and fit the GP hyperparameters by maximizing the marginal log-likelihood (the same model is used for all BO baselines).

When fitting a model for MORBO, we include the data within a hypercube around the trust region center with edgelength $2L$. In the case that there are less than $N_m := \min\{250, 2d\}$ points within that region, we include the N_m closest points to the trust region center for model fitting. The success streak tolerance is set to be infinity, which prevents the trust region from expanding; we find this leads to good optimization performance when data is shared across trust regions. For qNEHVI and qParEGO, we use 128 quasi-MC samples and for TS-TCH, we optimize RFFs with 500 Fourier basis functions. All three methods are optimized using L-BFGS-B with 20 random restarts. For DGEMO, TSEMO, and MOEA/D-EGO, we use the default settings in the open-source implementation at `https://github.com/yunshengtian/DGEMO/tree/master`. Similarly, we use the default settings for NSGA-II the Platypus package (`https://github.com/Project-Platypus/Platypus`). We encode the reference point as a black-box constraint to provide this information to NSGA-II.

D.1.1 LaMOO in High-Dimensional Search Spaces

For LaMOO methods, leverage the implementation of LaMOO available at `https://drive.google.com/drive/folders/1CMdg5iBdbKe3nkboIjiS998rnBEV09EB?usp=sharing`. We set the exploration parameter C_p dynamically using the heuristic proposed by Zhao et al. [2021] to be 10% of the hypervolume of the current Pareto frontier over the previously evaluated designs. We follow Zhao et al. [2021] and set the minimum leaf sample size to be 10.

Zhao et al. [2021] propose to use qEHVI with LaMOO, but we opt to use qNEHVI instead since it is capable of scaling to the batch size of $q = 50$ used in many of our experiments. We refer to this method as LaMOO-qNEHVI. We note that qNEHVI is mathematically equivalent to qEHVI on noiseless problems. The authors propose using rejection sampling to ensure samples come from the "good" region. For high-dimensional search spaces, the acceptance probability is low for

uniform random samples from the global design space, and therefore, rejection sampling is prohibitively slow. Rejection sampling is used 1) to select starting points for multi-start L-BFGS-B and within the L-BFGS-B routine to enforce that samples are within the "good" region. We contacted the authors about computational issues with this approach, and the authors recommended to use rejection sampling for selecting starting points, and then to simply run L-BFGS-B from these "good" starting points across the global search space. With this approach, the resulting candidates may not (and often are not) within the "good" region, and LaMOO-qNEHVI is simply an initialization heuristic for optimizing qNEHVI, but this approach does speed up candidate generation quite a bit. Nevertheless, even using rejection sampling to generate starting points for L-BFGS-B can be (and is on our problems) prohibitively expensive in high-dimensional search spaces. Hence, we limit the rejection sampling by only considering $120,000$ design points before beginning L-BFGS-B with the most promising designs (whether or not they are in the "good" region). This makes LaMOO-qNEHVI feasible to run our our high-dimensional problems.

For LaMOO-CMA-ES, we use $q = 5$ rather than $q = 1$ on vehicle safety, as $q = 1$ is not supported.

D.2 SYNTHETIC PROBLEMS

The reference points for all problems are given in Table 1. We multiply the objectives (and reference points) for all synthetic problems by -1 and maximize the resulting objectives.

PROBLEM	REFERENCE POINT
DTLZ2	[6, 6]
DTLZ3	$[10^3]^M$
DTLZ5	$[10]^M$
DTLZ7	$[15]^M$
VEHICLE SAFETY	[1698.55, 11.21, 0.29]
WELDED BEAM	[40, 0.015]
MW7	[1.2, 1.2]

Table 1: The reference points for each synthetic benchmark problem.

DTLZ: We consider the 2-objective DTLZ2 problem with various input dimensions $d \in \{10, 30, 100\}$. We also use 2-objective and 4-objective variants of DTLZ3, DTLZ5, and DTLZ7 with $d = 100$. The DTLZ problems are standard test problems from the multi-objective optimization literature. Mathematical formulas for the objectives in each problem are given in Deb et al. [2002].

MW7: For a second test problem from the multi-objective optimization literature, we consider a MW7 problem with 2 objectives, 2 constraints, and $d = 10$ parameters. See Ma and Wang [2019] for details.

Welded Beam: The welded beam problem [Ray and Liew, 2002] is a structural design problem with $d = 4$ input parameters controlling the size of the beam where the goal is to minimize 2 objectives (cost and end deflection) subject to 4 constraints. More details are given in Tanabe and Ishibuchi [2020].

Vehicle Safety: The vehicle safety problem is a 3-objective problem with $d = 5$ parameters controlling the widths of different components of the vehicle's frame. The goal is to minimize mass (which is correlated with fuel economy), toe-box intrusion (vehicle damage), and acceleration in a full-frontal collision (passenger injury). See Tanabe and Ishibuchi [2020] for additional details.

D.3 TRAJECTORY PLANNING

For the trajectory planning, we consider a trajectory specified by 30 design points that starts at the pre-specified starting location. Given the 30 design points, we fit a B-spline with interpolation and integrate over this B-spline to compute the final reward using the same domain as in Wang et al. [2018]. Rather than directly optimizing the locations of the design points, we optimize the difference (step) between two consecutive design points, each one constrained to be in the domain $[0, 0.05] \times [0, 0.05]$. We use a reference point of [0, 0.5], which means that we want a reward larger than 0 and a distance

that is no more than 0.5 from the target location $[0.95, 0.95]$. Since we maximize both objectives, we optimize the distance metric and the corresponding reference point value by -1.

D.4 OPTICAL DESIGN

In order to obtain precise estimates of the optimization performance at reasonable computational cost, we conduct our evaluation on a neural network surrogate model of the optical system rather than on the actual physics simulator. The surrogate model was constructed from a dataset of $101{,}000$ optical designs and resulting display images to provide an accurate representation of the real problem. The surrogate model is a neural network with a convolutional autoencoder architecture. The model was trained using $80{,}000$ training examples and minimizing MSE (averaged over images, pixels, and RGB color channels) on a validation set of $20{,}000$ examples. A total of $1{,}000$ examples were held-out for final evaluation.

D.5 MAZDA VEHICLE DESIGN PROBLEM

We follow the suggestions by Kohira et al. [2018] and use the reference point $[1.1, 0]$ and optimize the normalized objectives $\tilde{f}_1 = f_1 - 2$ and $\tilde{f}_2 = f_2/74$ corresponding to the total mass and number of common gauge parts, respectively. Additionally, an initial feasible point is provided with objective values $f_1 = 3.003$ and $f_2 = 35$, corresponding to an initial hypervolume of ≈ 0.046 for the normalized objectives. This initial solution is given to all algorithms. We limit the number of points used for model fitting to only include the $2{,}000$ points closest to the trust region center in case there are more than $2{,}000$ in the larger hypercube with side length $2L$. Still, for each iteration MORBO using 5 trust regions fits a total of 56×5 GP models, a scale far out of reach for any other multi-objective BO method.

E COMPLEXITY IMPROVEMENTS FROM LOCAL MODELING

The differences in model fitting time can be even more profound. To illustrate this, consider a situation in which a total of N data points have been collected by n_{TR} trust regions. Suppose for simplicity that each TR has the same number of observations (under some abuse of nomenclature we use TR to refer to the modeling domain of a TR in this section). Let η denote the average number of trust regions that a data point is part of. Then the number of points in each TR is $\eta N / n_{\text{TR}}$. Assuming cubic time complexity for model fitting (i.e. $O(N^3)$ if we used a single global model), the total time complexity of fitting all n_{TR} models in the individual TRs is $O\big(n_{\text{TR}}(\eta N / n_{\text{TR}})^3\big) = O\big(\eta^3 N^3 / n_{\text{TR}}^2\big)$. This will lead to asymptotic speedups of order $O\big(n_{\text{TR}}^2 / \eta^3\big)$ when using local modeling. Typically, as the optimization progresses and the trust regions shrink, η becomes quite small (e.g. $\eta < 1$)[3]. We validate this claim empirically in the lower right subplot in Figure 3, which shows that η becomes less than 1 on the all problems considered as the optimization progresses. In Figure 3 we illustrate some additional information from the trust regions to better understand the role of data-sharing and local modeling in MORBO. Thus, the speedup relative to fitting a single global model can be multiple orders of magnitude.

E.1 MODEL FITTING TIMES

Empirically, we verify this speedup in Figure 4. This can also be seen in the results in Tables 3 and 2. While candidate generation is fast for TSEMO, the model fitting causes a significant overhead with almost an hour being spent on model fitting after collecting $2{,}000$ evaluations on the trajectory planning problem. This is significantly longer than for MORBO, which only requires far less time for the model fitting due to the use of local modeling. This shows that the use of local modeling is a crucial component of MORBO that limits the computational overhead from the model fitting. The model fitting for MORBO on the optical design problem is less than 25 seconds while methods such as DGEMO and TSEMO that rely on global modeling require far more time for model fitting after only collecting $1{,}200$ points. Additionally, while MORBO needs to fit as many as $56 \times 5 = 280$ GP models on the Mazda problem due to the 54 black-box constraints and the use of 5 trust regions, the total time for model fitting still is less than 3 minutes while this problem is completely out of reach for the other BO methods that rely on global modeling.

[3] When η is close to the number of trust regions, the "local" models will fit to nearly all observations, and hence, the models will essentially be global models. The value of η at the start of the optimization depends on the initial trust region edge length and the dimension of the search space.

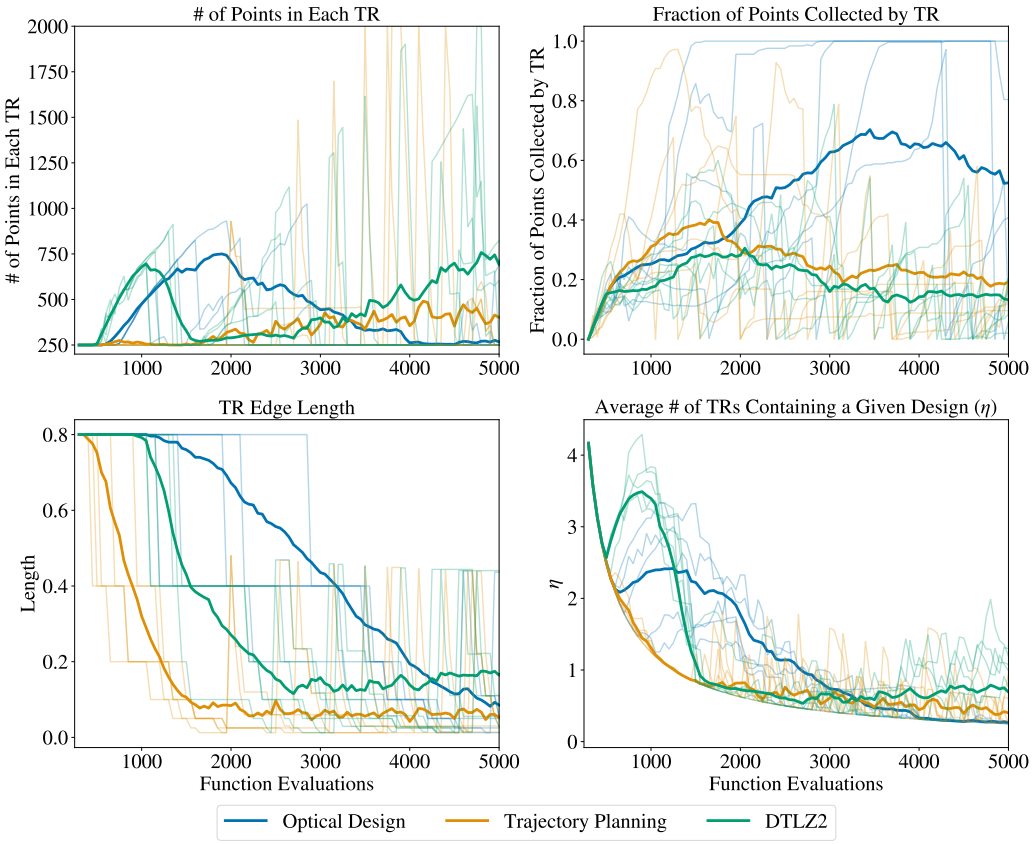

Figure 3: For the optical design, trajectory planning, and DTLZ2 problems. We show the average across replications as a solid line and traces from the first replication as transparent lines. (Upper Left) The number of points in each trust region. Trust regions often usually have a few hundred points on average, which results in computationally efficient local modeling. (Upper Right) The number of points in a trust region that was collected by that trust region. This shows that a large fraction of data within a trust region was actually collected by another trust region. (Lower Left) The trust region length. As the optimization proceeds, the trust regions shrink to focus on specific parts of the search space. (Lower Right) The average number of TRs that contain a given design, $\eta \in [0, N_{\text{TR}}]$. This shows that as the optimization progresses and the TRs shrink, on average less than 1 TR contains a given design. This is empirical validation of the claim in Appendix E that η typically becomes small as the optimization progresses and therefore, the complexity improvements are substantial.

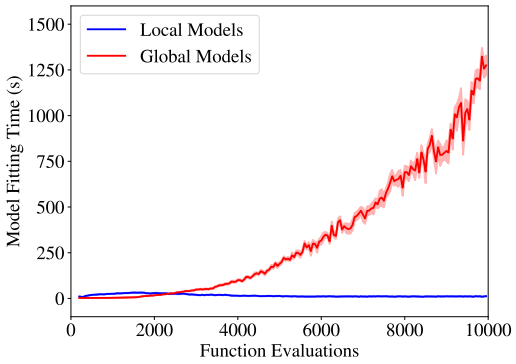

Figure 4: Model fitting time for MORBO with local modeling compared to MORBO with one global model on the 146-dimensional optical design problem. Fitting a global model takes almost 20 minutes towards the end of the optimization run compared to 10 seconds for MORBO.

PROBLEM	DTLZ3 ($M = 2$)	DTLZ5 ($M = 2$)	DTLZ7 ($M = 2$)	DTLZ3 ($M = 4$)	DTLZ5 ($M = 4$)	DTLZ7 ($M = 4$)
MORBO	11.0 (0.6)	9.7 (0.4)	10.6 (0.4)	11.5 (0.9)	10.5 (0.5)	10.6 (0.4)
NSGA-II	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
qPAREGO	139.5 (24.6)	49.1 (2.2)	26.0 (2.5)	137.2 (15.4)	113.2 (6.6)	49.0 (3.5)
TS-TCH	64.5 (3.4)	93.9 (5.8)	89.6 (3.5)	143.3 (5.9)	167.6 (8.8)	141.3 (6.1)
qNEHVI	133.2 (23.9)	48.9 (4.9)	20.8 (1.7)	25.9 (2.3)	19.8 (1.7)	6.8 (0.4)
DGEMO	5425.1 (142.0)	1438.0 (29.0)	180.0 (35.3)	N/A	N/A	N/A
TSEMO	4246.3 (91.8)	2481.5 (48.5)	958.4 (49.1)	3767.4 (91.0)	1892.3 (801.5)	402.0 (31.7)
MOEAD-EGO	3474.6 (108.6)	1824.0 (40.1)	1130.3 (16.0)	4206.1 (120.5)	2526.3 (77.5)	1048.0 (37.8)

Table 2: Model fitting wall time in seconds. The mean and two standard errors of the mean are reported. All models were fit on 2x Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz. For $M = 4$, qNEHVI exceeded GPU memory during acquisition optimization and therefore has shorter average model fitting times.

PROBLEM	WELDED BEAM	VEHICLE SAFETY	ROVER	OPTICAL DESIGN	MAZDA
MORBO	7.81 (0.02)	12.58 (0.26)	9.3 (0.19)	23.57 (0.36)	172.53 (1.89)
qPAREGO	0.5 (0.1)	0.1 (0.0)	51.6 (16.4)	46.7 (10.7)	N/A
TS-TCH	0.5 (0.0)	0.2 (0.0)	45.9 (1.8)	40.5 (4.9)	N/A
qNEHVI	0.5 (0.0)	0.1 (0.0)	97.8 (16.3)	46.4 (3.2)	N/A
DGEMO	N/A	N/A	809.7 (127.6)	1109.3 (178.7)	N/A
TSEMO	N/A	1.0 (0.1)	305.3 (38.2)	859.4 (131.4)	N/A
MOEA/D-EGO	N/A	0.9 (0.0)	373.2 (51.7)	736.4 (110.4)	N/A

Table 3: Model fitting wall time in seconds. The mean and two standard errors of the mean are reported. All models were fit on 2x Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz. For DGEMO, TSEMO and MOEA/D-EGO only $1{,}450$ evaluations were performed on Rover (Trajectory Planning) and only $1{,}250$ evaluations were performed on Optical Design, so the fitting times are shorter than if the full $2{,}000$ evaluations had been performed.

F ADDITIONAL RESULTS

F.1 LOW-DIMENSIONAL PROBLEMS

We consider two low-dimensional problems to allow for a comparison with existing BO baselines. The first problem we consider is a vehicle safety design problem ($d = 5$) in which we tune thicknesses of various components of an automobile frame to optimize proxy metrics for maximizing fuel efficiency, minimizing passenger trauma in a full-frontal collision, and maximizing vehicle durability. The second problem is a welded beam design problem ($d = 4$), where the goal is to minimize the cost of the beam and the deflection of the beam under the applied load [Deb and Sundar, 2006]. The design variables are the thickness and length of the welds and the height and width of the beam. In addition, there are 4 black-box constraints that must be satisfied.

Figure 5 presents results for both problems. While MORBO is not designed for such simple, low-dimensional problems, it is still competitive with other baselines such as TS-TCH and qParEGO on the vehicle design problem, though it cannot quite match the performance of qNEHVI and TSEMO.[4] The results on the welded beam problem illustrate the efficient constraint handling of MORBO.[5] On both problems, we observe that NSGA-II struggles to keep up, performing barely better (vehicle safety) or even worse (welded beam) than quasi-random Sobol exploration.

[4]DGEMO is not included on this problem as it consistently crashed due to an error deep in the low-level code for the graph-cutting algorithm.

[5]DGEMO, TSEMO, MOEA/D-EGO, and TS-TCH are excluded as they do not consider black-box constraints.

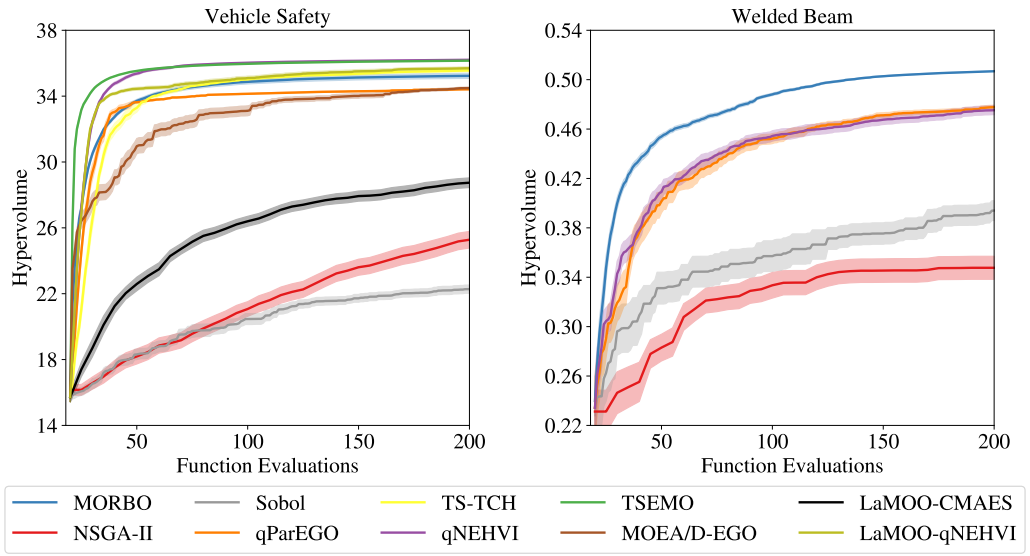

Figure 5: (Left) qNEHVI performs the best on the vehicle design problem ($d = 5$) with 3 objectives. (Right) MORBO outperforms the other methods on welded beam problem ($d = 4$) with 4 constraints.

F.2 CANDIDATE GENERATION WALL TIME

Problem Batch Size	Welded Beam ($q = 1$)	Vehicle Safety ($q = 1$)	Rover ($q = 50$)	Optical Design ($q = 50$)	Mazda ($q = 50$)
MORBO	1.3 (0.0)	9.6 (0.7)	23.4 (0.4)	9.8 (0.1)	188.16 (1.72)
qParEGO	14.5 (0.3)	1.3 (0.0)	213.4 (11.2)	241.9 (14.9)	N/A
TS-TCH	N/A	0.6 (0.0)	31.3 (1.1)	48.1 (1.2)	N/A
qNEHVI	30.4 (0.4)	9.1 (0.1)	997.5 (62.8)	211.27 (6.66)	N/A
NSGA-II	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
DGEMO	N/A	N/A	697.1 (52.5)	2278.7 (199.8)	N/A
TSEMO	N/A	3.4 (0.1)	3.3 (0.0)	4.6 (0.1)	N/A
MOEA/D-EGO	N/A	44.3 (0.3)	71.1 (4.3)	97.5 (6.7)	N/A
LaMOO-CMAES	N/A	0.6 (0.0)	2.6 (0.0)	51.9 (0.3)	N/A
LAMOO-qNEHVI	N/A	24.0 (2.3)	292.4 (25.2)	258.8 (1.9)	N/A

Table 4: Batch selection wall time (excluding model fitting) in seconds. The mean and two standard errors of the mean are reported. MORBO, qParEGO, TS-TCH, and qNEHVI were run on a Tesla V100 SXM2 GPU (16GB RAM), while DGEMO, TSEMO, MOEA/D-EGO and NSGA-II were run on 2x Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz. For Welded Beam and Vehicle Safety, we ran NSGA-II with $q = 5$ in order to avoid a singleton population. For DGEMO, TSEMO and MOEA/D-EGO only 1,450 evaluations were performed on Rover (Trajectory Planning) and only 1,250 evaluations were performed on Optical Design, so the generation times are shorter than if the full 2,000 evaluations had been performed.

While candidate generation time is often a secondary concern in classic BO applications, where evaluating the black box function often takes orders of magnitude longer, existing methods using a single global model and standard acquisition function optimization approaches can become the bottleneck in high-throughput asynchronous evaluation settings that are common with high-dimensional problems. Tables 4 and 5 provides a comparison of the wall time for generating a batch of candidates for the different methods on the different benchmark problems. We observe that the candidate generation for MORBO is two orders of magnitudes faster than for other methods such as qParEGO and qNEHVI on the trajectory planning problem where all methods ran for the full 2,000 evaluations.

Problem Batch Size	DTLZ3 ($M = 2$) ($q = 50$)	DTLZ5 ($M = 2$) ($q = 50$)	DTLZ7 ($M = 2$) ($q = 50$)	DTLZ3 ($M = 4$) ($q = 50$)	DTLZ5 ($M = 4$) ($q = 50$)	DTLZ7 ($M = 4$) ($q = 50$)
MORBO	26.0 (1.3)	25.1 (0.9)	293.0 (21.9)	976.9 (89.8)	973.0 (91.8)	293.0 (21.9)
qParEGO	315.8 (20.2)	299.0 (27.2)	233.0 (21.5)	372.9 (46.6)	373.1 (34.6)	232.4 (22.2)
TS-TCH	43.6 (1.4)	49.6 (2.0)	39.5 (1.9)	56.5 (1.8)	69.2 (7.5)	51.4 (3.4)
qNEHVI	2877.7 (321.3)	1879.6 (285.4)	816.9 (49.1)	4412.9 (600.7)	3778.2 (266.5)	57.6 (4.4)
NSGA-II	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.1 (0.0)	0.0 (0.0)	0.0 (0.0)
DGEMO	N/A	N/A	N/A	N/A	N/A	N/A
TSEMO	6.3 (0.1)	7.2 (0.1)	6.8 (0.1)	2878.1 (162.0)	952.0 (298.1)	22.2 (3.7)
MOEAD-EGO	277.8 (1.2)	224.9 (3.2)	245.3 (2.9)	308.7 (2.9)	303.7 (3.1)	292.2 (3.5)

Table 5: Batch selection wall time (excluding model fitting) in seconds on DTLZ problems with 2 and 4 objectives with $d = 100$. The mean and two standard errors of the mean are reported.

F.3 PARETO FRONTIERS

We show the Pareto frontiers for the welded beam, trajectory planning, optical design, and Mazda problems in Figure 6. In each column we show the Pareto frontiers corresponding to the worst, median, and best replications according to the final hypervolume. We exclude the vehicle design problem as it has three objectives which makes the final Pareto frontiers challenging to visualize.

Figure 6 shows that even on the low-dimensional 4D welded beam problem, MORBO is able to achieve much better coverage than the baseline methods. MORBO also explores the trade-offs better than other methods on the trajectory planning problem, where the best run by MORBO found trajectories with high reward that ended up being close to the final target location. In particular, other methods struggle to identify trajectories with large rewards while MORBO consistently find trajectories with rewards close to 5, which is the maximum possible reward. On both the optical design and Mazda problems, the Pareto frontiers found by MORBO better explore the trade-offs between the objectives compared to NSGA-II and Sobol. We note that MORBO generally achieves good coverage of the Pareto frontier for both problems. For the optical design problem, we exclude the partial results found by running the other baselines for 1k-2k evaluations and only show the methods the ran for the full 10k evaluations. For the Mazda problem we show the Pareto frontiers of the true objectives and not the normalized objectives that are described in Section 5.1. MORBO is able to significantly decrease the vehicle mass at the cost of using a fewer number of common parts, a trade-off that NSGA-II fails to explore. It is worth noting that the number of common parts objective is integer-valued and that exploiting this additional information may unlock even better optimization performance of MORBO.

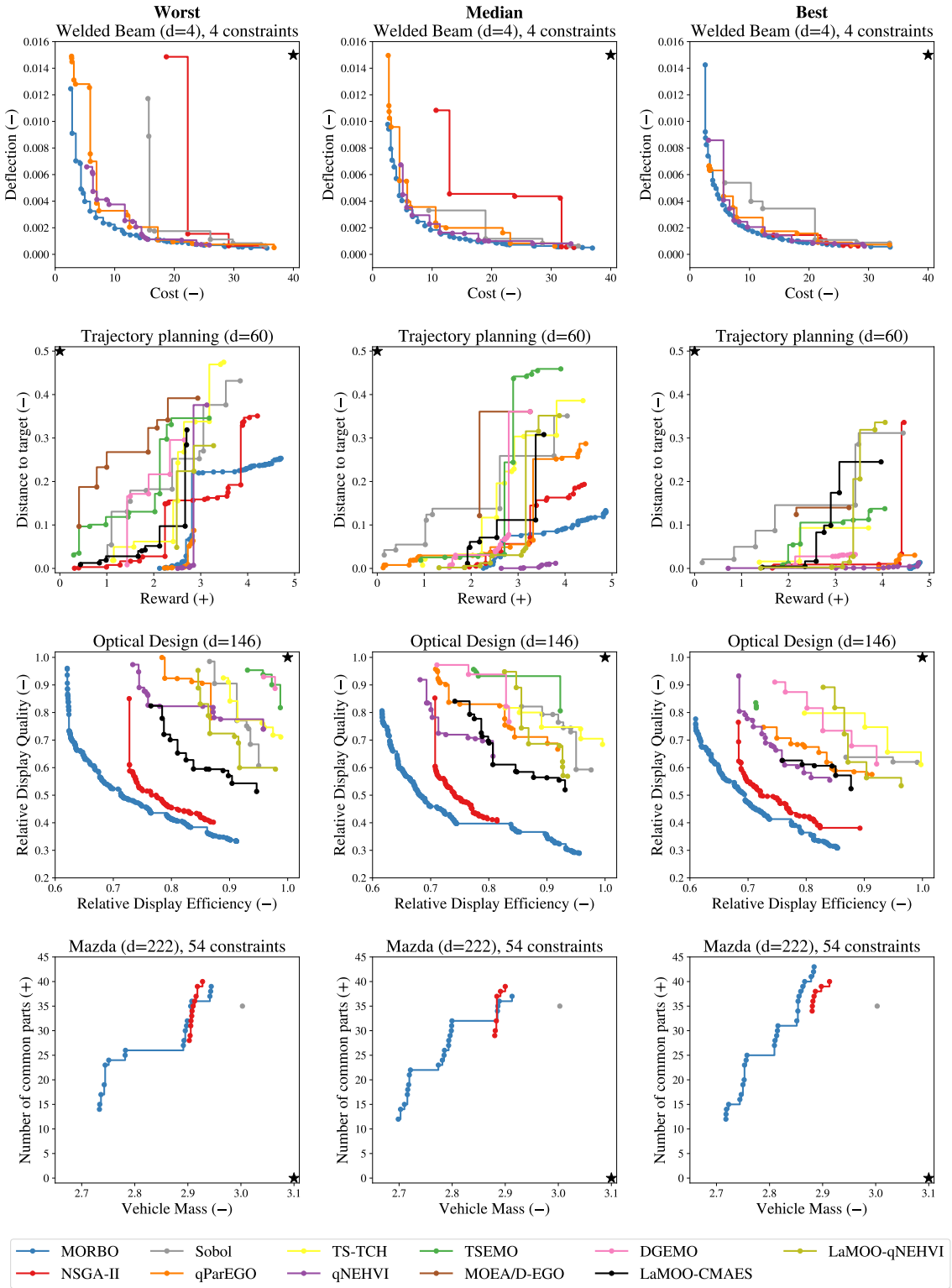

Figure 6: In each column we show the Pareto frontiers for the worst, median, and best replications according to the final hypervolume. We indicate whether an objective is minimized/maximized by $-/+$, respectively. The reference point is illustrated as a black star. The use of multiple trust regions allows MORBO to consistently achieve good coverage of the Pareto frontier, in addition to large hypervolumes.

F.4 ADDITIONAL BENCHMARK PROBLEMS

To study the performance of MORBO on a broader range of problems, we evaluate MORBO on two-objective and four-objective versions of DTLZ3, DTLZ5, and DTLZ7 problems with $d = 100$. As shown in Figure 8, MORBO performs best on the four-objective DTLZ7 and achieve the best final hypervolume on the four-objective DTLZ3 problem. On the two-objective problems, MORBO always ranks in the top 4 methods as shown in Figure 8. To compare the performance in general across the DTLZ3, DTLZ5, and DTLZ7 problems with a given number of objectives, we rank the methods by the average final hypervolume across replications and compute the average rank across the three problems. As shown in Table 6, MORBO achieves the lowest rank across all methods (which is best) on both M=2 and M=4 problems. DGEMO is not evaluated on the 4-objective problems because the open-source implementation (`https://github.com/yunshengtian/DGEMO/tree/master`) does not support more than two objectives. Although DGEMO, MOEA/D-EGO and qNEHVI all perform competitively in the two objective setting, all methods are significantly slower than MORBO.

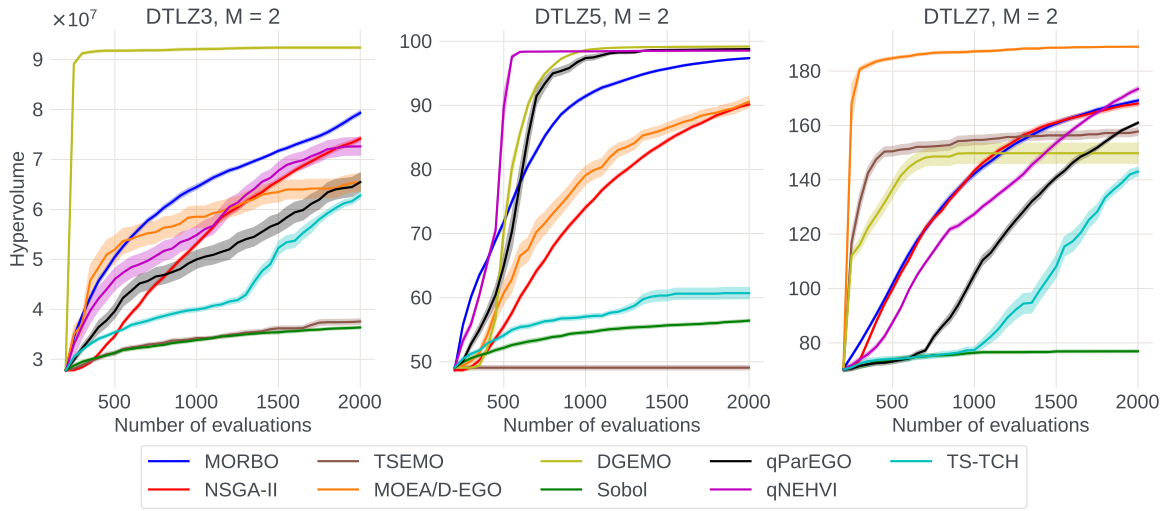

Figure 7: Optimization performance on two-objective DTLZ3, DTLZ5, and DTLZ7 problems with $d = 100$ and $q = 50$.

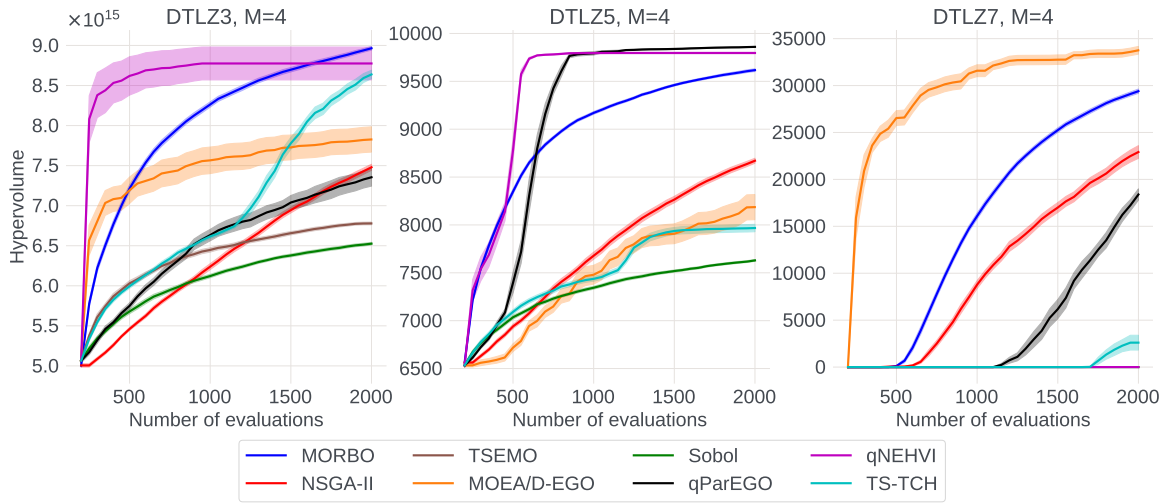

Figure 8: Optimization performance on four-objective DTLZ3, DTLZ5, and DTLZ7 problems with $d = 100$ and $q = 50$.

	AVG. RANK FOR M=2	AVG. RANK FOR M=4
MORBO	3.0	1.67
qPAREGO	4.0	3.3
qNEHVI	3.0	3.16
TS-TCH	7.3	4.3
NSGA-II	4.3	3.7
DGEMO	3.0	8.2
TSEMO	7.7	8.3
MOEA/D-EGO	4.0	5.5
SOBOL	8.7	6.8

Table 6: Mean rank across DTLZ3, DTLZ5, and DTLZ7 problems based on final mean hypervolume with $d = 100$ and $q = 50$. A lower rank means the method achieves better final performance on average across the DTLZ3, DTLZ5, and DTLZ7 problems with M objectives.

References

E. Bradford, A. M. Schweidtmann, and A. Lapkin. Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. *J. of Global Optimization*, 2018.

S. Daulton, M. Balandat, and E. Bakshy. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In *Advances in Neural Information Processing Systems 34*, 2021.

K. Deb and J. Sundar. Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006.

K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. volume 1, 2002.

D. Eriksson and M. Poloczek. Scalable constrained Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021.

D. Eriksson, M. Pearce, J. R. Gardner, R. Turner, and M. Poloczek. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems 32*, 2019.

T. Kohira, H. Kemmotsu, O. Akira, and T. Tatsukawa. Proposal of benchmark problem based on real-world car structure design optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2018.

X. Li and C.-P. Chen. Inequalities for the gamma function. *Journal of Inequalities in Pure and Applied Mathematics*, 8, 2007.

Z. Ma and Y. Wang. Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons. *IEEE Transactions on Evolutionary Computation*, 23(6), 2019.

C. Oh, E. Gavves, and M. Welling. BOCK: Bayesian optimization with cylindrical kernels. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, 2018.

B. Paria, K. Kandasamy, and B. Póczos. A flexible framework for multi-objective Bayesian optimization using random scalarizations. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115, 2020.

G. Pleiss, M. Jankowiak, D. Eriksson, A. Damle, and J. Gardner. Fast matrix square roots with applications to Gaussian processes and Bayesian optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, 2007.

T. Ray and K. Liew. A swarm metaphor for multiobjective design optimization. *Engineering Optimization*, 34(2), 2002.

R. G. Regis and C. A. Shoemaker. Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. *Engineering Optimization*, 45(5), 2013.

R. Tanabe and H. Ishibuchi. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89, 2020.

Z. Wang, C. Gehring, P. Kohli, and S. Jegelka. Batched large-scale Bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, volume 84, 2018.

R. Zhang and D. Golovin. Random hypervolume scalarizations for provable multi-objective black box optimization. In *International Conference on Machine Learning*, 2020.

Y. Zhao, L. Wang, K. Yang, T. Zhang, T. Guo, and Y. Tian. Multi-objective optimization by learning space partitions, 2021.