Supplementary Materials

1. DEMO: Differential Evolution for Multi-Objective Optimization

The DEMO (Differential Evolution for Multi-Objective, Robič & Filipič (2005)) algorithm is used for the multi-objective optimization of acquisition functions, we briefly introduce the algorithm in this section. It should be noted that other multi-objective optimization algorithms can also be used for the proposed MACE algorithm.

The DEMO algorithm follows the basic procedure of evolutionary algorithms. Firstly, a set of points are randomly sampled to form the initial population, at each iteration, random variations are added to parent population via mutation and crossover to generate the children population, the parent population and children population are compared to create the parent population for the next generation. During the evolution, the Pareto front is recorded. The DEMO algorithm is summarized in Algorithm 1.

Algorithm 1 DEMO

1: Randomly sample the N points in the design space to form the initial population P^1
2: **for** t = 1, 2, ... G **do**
3: M^t = `mutation`(P^t)
4: C^t = `crossover`(P^t, M^t)
5: P^{t+1} = `selection`(P^t, C^t)
6: **end for**
7: Return the recorded Pareto front

At the i-th iteration, we denote the population as $P^i \in R^{N \times D}$, where N is the number of population, and D is the dimension of input variables. The mutated population $M^i \in R^{N \times D}$ is generated by

$$M_j^i = P_{r1}^i + F \times (P_{r2}^i - P_{r3}^i),\ j \in \{1 \dots NP\} \quad (1)$$

where M_j^i means the j-th row of M^i, while P_{r1}^i, P_{r2}^i and P_{r3}^i are the $r1$-th, $r2$-th and $r3$-th rows of P^i, the $r1$, $r2$ and $r3$ are three integers randomly chosen from $[1, N]$. The scaling factor $F \in (0, 1)$ is an algorithm parameter to control the mutation.

After the mutation, crossover operations are performed to generated the children population $C^i \in R^{N \times D}$. For each element of C^i, two random numbers r_r and r_i are generated, r_r is a real-valued number uniformly sampled from $(0, 1)$, r_i is an integer number uniformly sampled from $[1, D]$, the element of C^i is calculated by

$$C_{jk}^i = \begin{cases} P_{jk}^i & r_r < CR \text{ and } r_i \neq k \\ C_{jk}^i & \text{otherwise.} \end{cases} \quad (2)$$

where the crossover rate CR is an algorithm parameter to control the crossover.

Now that we have the parent population P^i and the children population C^i, we perform selection to generate the new population P^{i+1}. Firstly, an empty archive is created, we perform pair-wise compairsion between parents and children, if one parent solution dominates its child solution, the parent solution is added into the archive; if the child solution dominates its parent, the child is added into the archive; if the parent and its child don't dominate each other, both the parent and the child are added into the archive. After the pair-wise compairsion, non-dominated sorting(Deb et al., 2002) is performed to select G solutions as the parent population of the next generation. The non-dominated sorting method defines a complete order between a group of solutions, details of the non-dominated sorting can be seen in (Deb et al., 2002).

As has been mentioned, four algorithm parameters are to be set for the DEMO algorithm: the population size N, the number of generations G, the scaling factor F and the crossover rate CR. We set $N = 100$, $G = 250$, $F = 0.5$ and $CR = 0.3$ for all the experiments performed in the paper.

2. Additional Experiments with Varied Batch Sizes

We performed additional experiments with varied batch sizes $B = 2$, $B = 3$ and $B = 5$, the results for the analytical benchmark functions are shown in Table 1, Table 2 and Table 3. The optimization results of the operational amplifier are listed in Table 4, the optimization results of the class-E power amplifier are given in Table 5. With varied batch sizes, the proposed MACE method remain competitive compared with the state-of-the-art batch Bayesian optimization methods.

Table 1. Statistics of the regrets of the benchmark functions with batch size $B = 2$

Algorithm	MACE	BLCB	EI-LP	QKG	QEI
Ackley	1.15 ± 0.646	1.7 ± 0.85	**0.507 ± 0.408**	4.31 ± 1.81	3.07 ± 0.786
Alpine1	**1.38 ± 0.768**	3.23 ± 1.03	1.55 ± 0.689	3.17 ± 0.749	2.4 ± 0.904
Branin	**5.6e-6 ± 1.03e-5**	1.86e-4 ± 2.65e-4	0.0257 ± 0.0395	0.21 ± 0.159	8.27e-4 ± 1.56e-3
Eggholder	116 ± 65.4	132 ± 66.2	**82.7 ± 51.6**	115 ± 78.3	104 ± 78.4
Hartmann6	**0.0479 ± 0.0584**	0.0719 ± 0.0587	0.161 ± 0.301	0.257 ± 0.0823	0.178 ± 0.128
Rosenbrock	**1.05e-3 ± 0.0011**	5.56e-3 ± 9.28e-3	8.37 ± 5.63	9.41 ± 10.7	10.3 ± 8.52
Ackley10D	**2.75 ± 0.497**	3.13 ± 0.723	18.5 ± 1.02	18.4 ± 0.943	18.8 ± 0.608
Rosenbrock10D	**223 ± 104**	552 ± 223	1.1e+03 ± 496	957 ± 439	757 ± 405

Table 2. Statistics of the regrets of the benchmark functions with batch size $B = 3$

Algorithm	MACE	BLCB	EI-LP	QKG	QEI
Ackley	1.37 ± 1.39	1.71 ± 1.02	**0.216 ± 0.148**	5.49 ± 1.94	2.34 ± 0.788
Alpine1	**1.03 ± 0.746**	2.63 ± 1.2	1.1 ± 0.376	3.18 ± 0.225	2.25 ± 0.42
Branin	**2.85e-5 ± 3.18e-5**	8.14e-5 ± 1.27e-4	0.0344 ± 0.0183	0.247 ± 0.188	5.21e-5 ± 1.35e-4
Eggholder	**65.3 ± 62.9**	82.6 ± 32.2	65.9 ± 43.3	117 ± 79.2	81.7 ± 63.1
Hartmann6	**0.012 ± 0.0359**	0.0477 ± 0.0584	0.0489 ± 0.0531	0.335 ± 0.188	0.189 ± 0.108
Rosenbrock	**9.46e-4 ± 7.75e-4**	0.00148 ± 0.00212	3.78 ± 3.4	4.28 ± 5.5	5.44 ± 4.21
Ackley10D	3.05 ± 0.682	**3.05 ± 0.431**	17.6 ± 3.53	18.5 ± 0.731	18.6 ± 0.438
Rosenbrock10D	**208 ± 92.5**	389 ± 187	653 ± 473	695 ± 307	953 ± 410

Table 3. Statistics of the regrets of the benchmark functions with batch size $B = 5$

Algorithm	MACE	BLCB	EI-LP	QKG	QEI
Ackley	1.7 ± 1.02	1.38 ± 0.836	**0.105 ± 0.0978**	5.27 ± 1.38	2.16 ± 1.11
Alpine1	**0.654 ± 0.317**	1.68 ± 1.26	0.766 ± 0.441	3.21 ± 0.497	2.05 ± 0.341
Branin	**1.26e-5 ± 1.81e-5**	2.99e-5 ± 3.42e-5	0.0144 ± 0.0154	0.163 ± 0.163	2.02e-5 ± 5.21e-5
Eggholder	74.1 ± 74.3	61.1 ± 33.5	63.5 ± 94.3	71 ± 29.4	**49.1 ± 25.8**
Hartmann6	0.0477 ± 0.0584	**0.0358 ± 0.0546**	0.0552 ± 0.0546	0.47 ± 0.221	0.198 ± 0.105
Rosenbrock	**5.48e-4 ± 8.12e-4**	9.39e-4 ± 6.83e-4	2.72 ± 1.97	3.42 ± 4.8	6.69 ± 5.34
Ackley10D	**2.63 ± 0.486**	3.05 ± 0.319	15.7 ± 5.69	18.1 ± 0.476	18.1 ± 0.653
Rosenbrock10D	**81.9 ± 22.9**	348 ± 83.7	645 ± 470	893 ± 393	705 ± 314

Table 4. Optimization Results of the Operational Amplifier with $B = 2$, $B = 3$ annd $B = 5$

Algorithm	MACE	BLCB	EI-LP
B=2	**-689 ± 4.37**	-649 ± 28.8	-627 ± 48.5
B=3	**-690 ± 0.518**	-672 ± 20.4	-621 ± 45.2
B=5	**-690 ± 0.0251**	-684 ± 6.86	-626 ± 49

Table 5. Optimization Results of the class-E Power Amplifier with $B = 2$, $B = 3$ annd $B = 5$

Algorithm	MACE	BLCB	EI-LP
B=2	**-4.13 ± 0.207**	-4.01 ± 0.208	-3.65 ± 0.312
B=3	**-4.45 ± 0.326**	-4.17 ± 0.163	-3.87 ± 0.306
B=5	**-4.26 ± 0.18**	-4.17 ± 0.111	-4.18 ± 0.222

References

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2): 182–197, 2002.

Robič, T. and Filipič, B. Demo: Differential evolution for multiobjective optimization. In Coello Coello, C. A., Hernández Aguirre, A., and Zitzler, E. (eds.), *Evolutionary Multi-Criterion Optimization*, pp. 520–533, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31880-4.