Solving Expensive Optimization Problems in Dynamic Environments with Meta-learning ——Supplementary Material

Huan Zhang, Jinliang Ding, *Senior Member, IEEE,* Liang Feng, *Senior Member, IEEE,* Kay Chen Tan, *Fellow, IEEE,* Ke Li, *Senior Member, IEEE,*

I. Appendix A

In this section, we sequentially introduce the mathematical definitions of the benchmark problem, the working mechanisms of the peer algorithms, and the fundamental ideas of the statistical measurements.

A. *Mathematical Definitions of Benchmark Problem*

In our experiments, we choose the moving peaks benchmark (MPB) [1] as the benchmark test suite. The function of MPB has the following form:

$$f(\mathbf{x}, t) = \max_{i \in \{1, \dots m\}} \{H_i(t) - W_i(t) \parallel \mathbf{x} - X_i(t) \parallel\}, \tag{1}$$

where m is the number of peaks, $\|\cdot\|$ represents the Euclidean norm, $H_i(t)$, $W_i(t)$, and $X_i(t)$ denote the height, width, and location of the i-th peak at the t-th time step (or environment), respectively. To be specific, the height, width, and location of peaks change can be described as follows:

$$\begin{aligned} H_i(t) &= H_i(t-1) + h_{sev} \cdot \sigma, \\ W_i(t) &= W_i(t-1) + w_{sev} \cdot \sigma, \\ X_i(t) &= X_i(t-1) + \boldsymbol{v}_i(t), \\ \boldsymbol{v}_i(t) &= \frac{x_{sev}}{\boldsymbol{v}_i(t-1) + \boldsymbol{r}}[(1-\lambda)\boldsymbol{r} + \lambda \boldsymbol{v}_i(t-1)]. \end{aligned} \tag{2}$$

where h_{sev}, w_{sev}, and x_{sev} indicate height severity, width severity, and shift severity, respectively, and σ is a random number obeyed a Gaussian distribution with variance 1 and mean 0. $\boldsymbol{v}_i(t)$ and $\boldsymbol{v}_i(t-1)$ represent the current and previous shift vector of the i-th peak, respectively. r is a random vector and λ is the correlation coefficient. $\lambda = 0$ denotes that the peak relocations are in a random direction, whereas they depend on the previous direction for $\lambda > 0$.

TABLE I
Parameters for MPB

Parameter	Value	Parameter	Value
Peak shape	Cone	Correlation coefficient, λ	0.5
Number of peaks	5	Range of space for each dimension	[0, 100]
Shift severity, x_{sev}	5, 10	Range of height for each peak, H_i	[1, 12]
Height severity, h_{sev}	7, 10	Range of width for each peak, W_i	[30, 70]
Width severity, w_{sev}	1		

B. *Peer Algorithms*

A concise description of peer algorithms is provided as follows. More details can be referred to the corresponding original papers.

1) Discounted information through noise sampling strategy (`DIN`) [2]: The strategy represents the best variant for tracking the changing global optima in dynamic environments described in [2]. It discounts older samples by introducing artificial measurement noise, which is a function of the time step of the samples and is added to the covariance function.

H. Zhang and J. Ding are with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, 110819, China (e-mail: huanzhang0320@163.com; jlding@mail.neu.edu.cn).
L. Feng is with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: liangf@cqu.edu.cn).
K. C. Tan is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong SAR (e-mail: kctan@polyu.edu.hk).
K. Li is with University of Electronic Science and Technology of China, Chengdu, 611731, China, and the Department of Computer Science, University of Exeter, Exeter, EX4 4QF, UK (e-mail: ke.li@ieee.org).

2) Model-based optimization (MBO) [3]: This algorithm picks out the observations collected from the previous time steps within a sliding time window. In addition, the time is augmented as a covariate, allowing the surrogate to model the effect of the time directly.

3) Transfer Bayesian optimization (TBO) [4]: This algorithm develops a transfer learning strategy to empower BO for handling expensive DOPs. The basic idea is to employ an augmented covariance function to realize knowledge transfer among GPR models from different time steps. To save the computational cost for GPR modeling, a decay mechanism is proposed to remove less irrelevant samples.

C. Statistical Tests

In this paper, three statistical measures are adopted in our empirical study. Their basic ideas are introduced as follows.

- Wilcoxon signed-rank test [5]: It is a non-parametric statistical hypothesis test. Particularly, the significance level is set to 0.05 in our experiments.
- Scott-Knott test [6]: This measure is used to rank the performance of peer algorithms over 20 runs on test problems. In a nutshell, it applies a statistical test and effect size to split the performance of all peer algorithms into several clusters. In the same cluster, the performance of peer algorithms is statistically equivalent. Note that the clustering procedure terminates until no split can be made. Finally, each cluster is assigned a rank according to the mean E_{BBC} values obtained by all peer algorithms within the cluster. Since a smaller E_{BBC} value is preferred, a smaller rank suggests a better performance of the algorithm achieves.
- A_{12} effect size [7]: To make sure that the difference is not caused by a trivial effect, A_{12} is used as the effect size measure to assess the probability that one algorithm outperforms another. To be specific, given a pair of peer algorithms, $A_{12} = 0.5$ denotes they are equivalent, whereas $A_{12} > 0.5$ indicates that one obtains better results for more than 50% of the times. $0.56 \leq A_{12} < 0.64$, $0.64 \leq A_{12} < 0.71$, and $A_{12} \geq 0.71$ represent a small, a medium, and a large effect size, respectively.

II. Appendix B

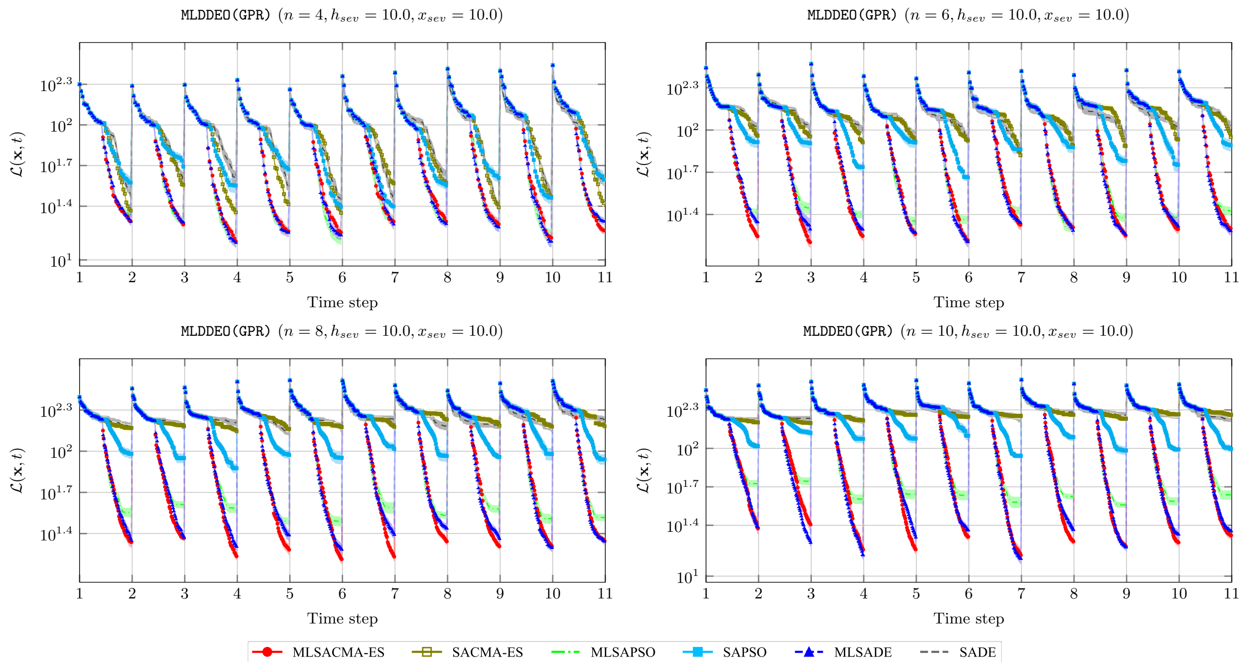
In this section, we first analyze the remaining concerns: 4) the robustness of the proposed algorithm under varying degrees of environmental changes, and 5) the investigation of the proposed framework applying different optimization solvers and surrogate models. Then, the experimental results (figures and tables) relevant to the main text are also presented.

A. Robustness of the proposed algorithm

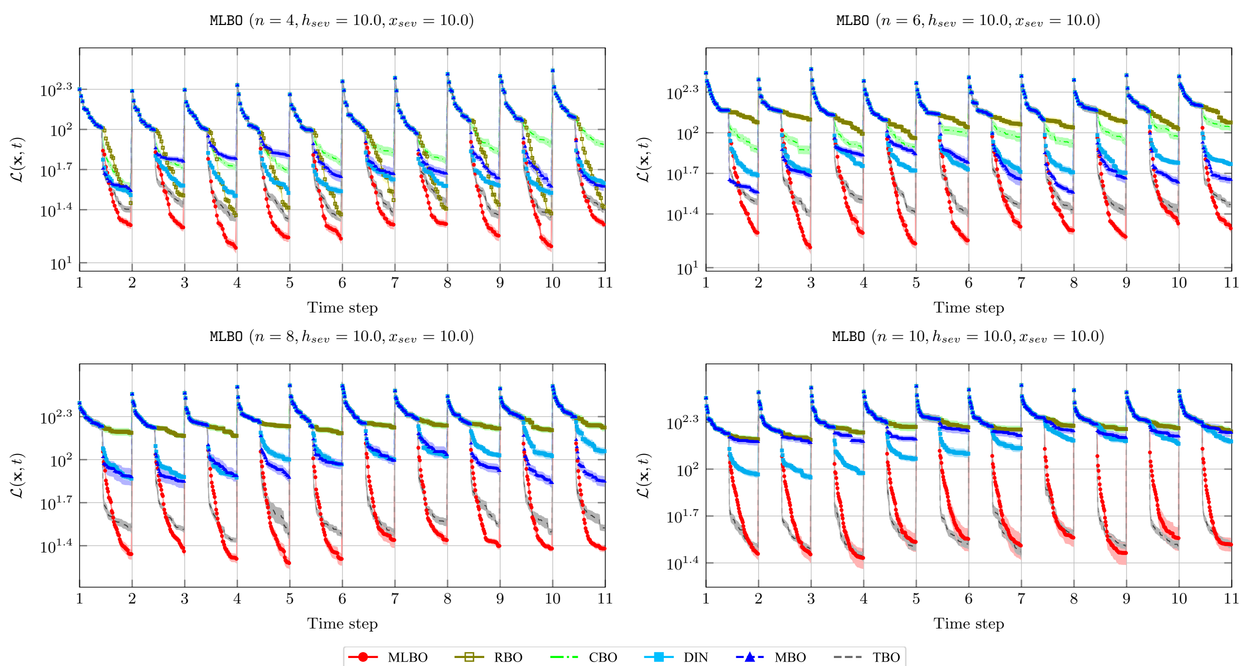
To investigate the robustness of the proposed algorithm, experiments are carried out on different settings of h_{sev} and x_{sev}. In particular, $(h_{sev}, x_{sev}) = (7.0, 5.0)$ denotes a slight change of the fitness landscape while $(h_{sev}, x_{sev}) = (10.0, 10.0)$ indicates a severe change.

Again, we plot the loss function curves between different algorithms under the setting of $(h_{sev}, x_{sev}) = (10.0, 10.0)$ during the evolutionary process as shown in Figs. 1 and Figs. 2. Based on the results shown in Table II and III, as well as Figs. 1 and Figs. 2, we observe that, even under a severe environmental change, the proposed algorithm instances can still obtain promising results with respective peer algorithms. To be specific, the proposed algorithm instances obtain the smallest E_{BBC} metric value as shown in Table II and III. In addition, the proposed algorithm instances can approximate the current global optimum more accurately than peer algorithms within the same function evaluations, as shown in Figs. 1 and Figs. 2. It is interesting to note that the metric values of our proposed algorithm instances are close under different environmental changes, indicating that the proposed instances are resilient and robust to different levels of environmental changes. SACMA-ES, SAPSO, SADE, and RBO perform similarly poorly under different environments due to random search after each change. In contrast, MBO and CBO are more sensitive to the levels of environmental changes. It can be seen from the improvement of the metric values when increasing the value of h_{sev} and x_{sev}. The reason may be that the knowledge transfer approaches adopted in MBO and CBO might bring negative impacts when solving DOPs with severe changes.

We also apply the Scott-Knott test and the A_{12} effect size to evaluate the performance of our algorithm instances against the state-of-the-art peer algorithms on benchmark problems with (h_{sev}, x_{sev}) = (10.0, 10.0). The distribution and median of the results are shown as box plots in Figs. 3 and the bar charts are shown in Figs. 4, respectively. From these results, it is obvious that our proposed algorithm instances consistently outperform other peer algorithms in the corresponding comparisons, although a severe change poses difficulties to the problem. Specifically, our proposed algorithm instances are consistently ranked in the smaller values, as the box plots shown in Figs. 3. Moreover, in all comparisons obtained from MLDDEO and MLBO, the large effect size is always 100%, as shown in Figs. 4.

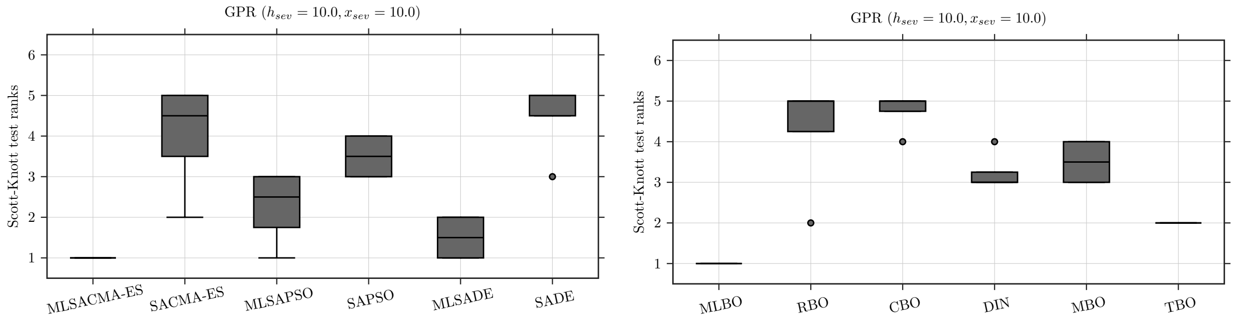


Figs. 1. Loss function of the mean errors between the true objective functions and the best objective values along with a confidence level over time at different dimensions with $h_{sev} = 10.0$ and $x_{sev} = 10.0$ when comparing `MLDDEO` by using GPR with other peer algorithms.

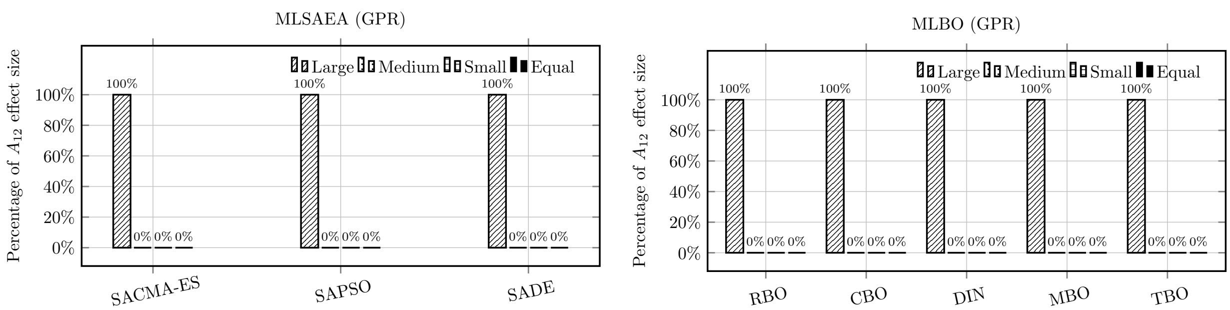


Figs. 2. Loss function of the mean errors between the true objective functions and the best objective values along with a confidence level over time at different dimensions with $h_{sev} = 10.0$ and $x_{sev} = 10.0$ when comparing `MLBO` with other peer algorithms.

Response to concern 4: Increasing the levels of environmental changes usually makes the problem-solving process more difficult. However, when facing the different magnitude of environmental changes considered in this paper, the proposed algorithm instances can achieve promising results compared to respective peer algorithms under two optimization mechanisms. In short, the proposed framework has more robust performance to different levels of environmental changes.



Figs. 3. Box plots of Scott-Knott test ranks of E_{BBC} achieved by the proposed algorithm instances compared against the corresponding state-of-the-art peer algorithms with $h_{sev} = 10.0$ and $x_{sev} = 10.0$ (the smaller rank is, the better performance achieved).



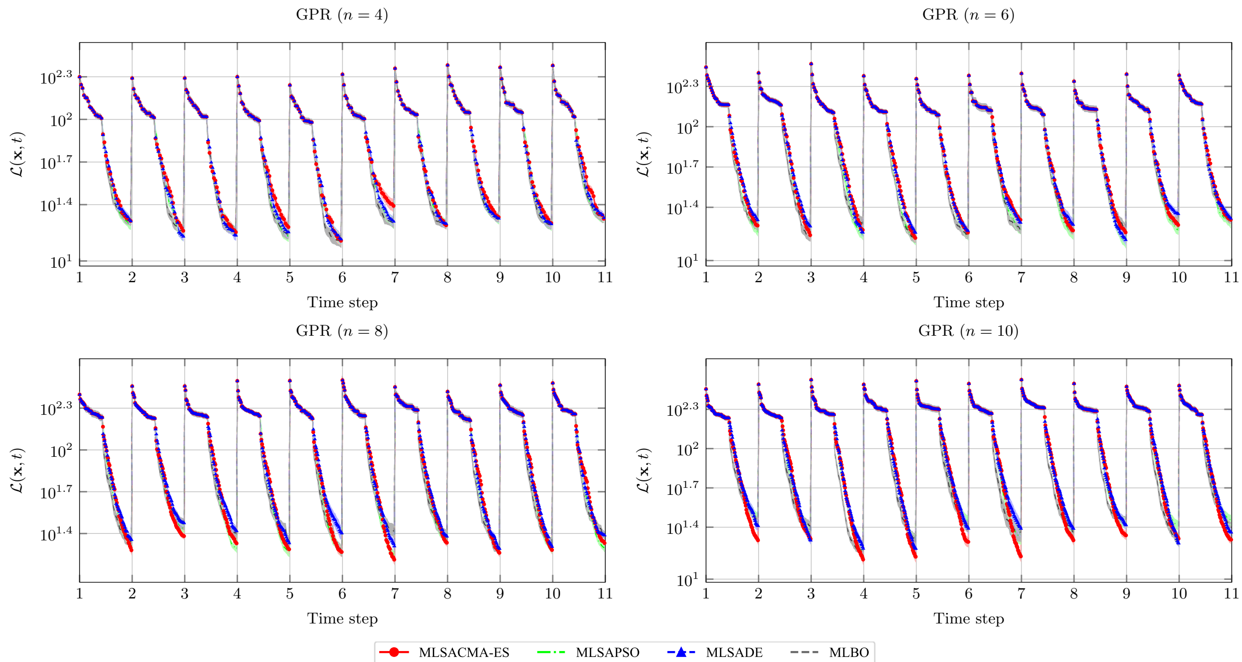
Figs. 4. Percentage of A_{12} effect size of E_{BBC} with $h_{sev} = 10.0$ and $x_{sev} = 10.0$ when comparing MLDDEO or MLBO with corresponding state-of-the-art peer algorithms.

B. Further studies of the proposed framework

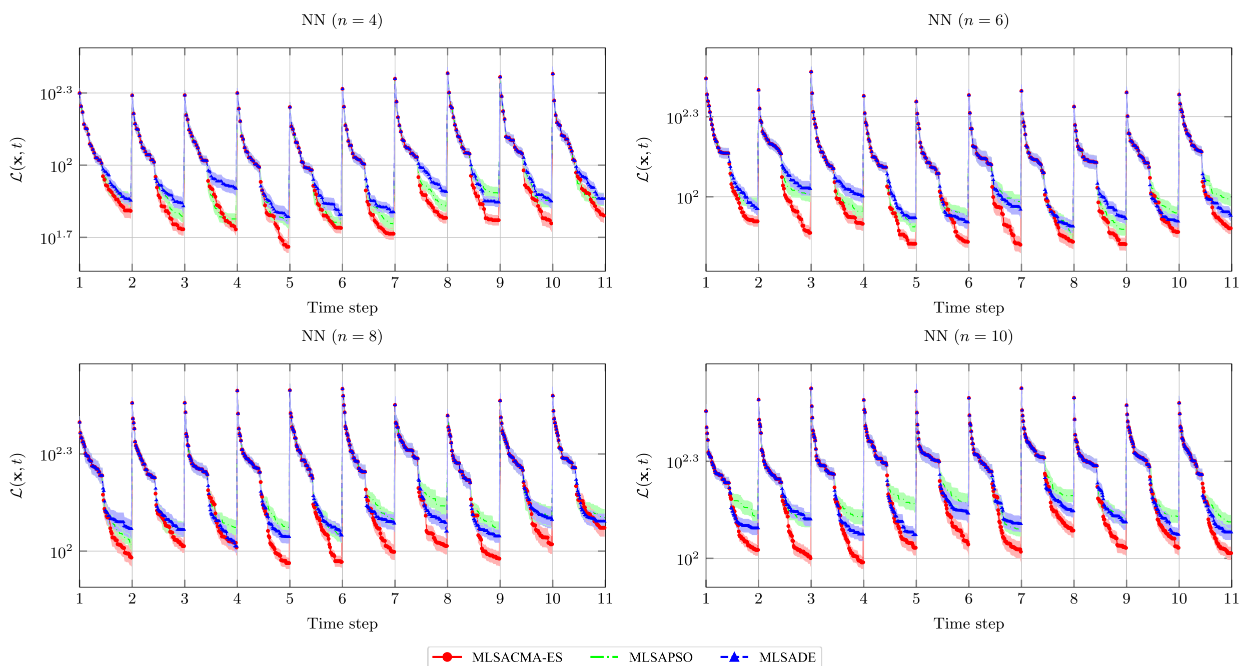
To investigate concern 5, we proceed to address the following two research questions. 1) Given a surrogate model, how does the performance differ in the proposed algorithm instances? 2) How is the performance using different surrogate models in the proposed method? Since we consider two different optimization mechanisms (i.e., `MLDDEO` and `MLBO` as introduced in Section III-A) and two surrogate models (i.e., GPR and NN) in the algorithm framework, this subsection aims to study the performance comparison of seven different algorithm instances under these two optimization mechanisms and two surrogate models separately. The statistical comparison results of E_{BBC} based on the Wilcoxon signed-rank test are shown in Table II and III. In addition, the loss function curves during the evolutionary process among all algorithm instances can be found in Figs. 5 and Figs. 6. From these results, we can see that `MLDDEO` is more effective in handling higher dimensional problems compared to `MLBO`. In particular, the metric and loss values obtained by all algorithm instances are close in the lower dimensions (i.e., $n = 4, 6$). However, in higher dimensions, `MLDDEO` has shown better E_{BBC} values and smaller loss values than `MLBO`. One plausible explanation is that SAEAs utilize a population-based search strategy to explore the search space. On the one hand, this strategy maintains a diverse set of solutions. On the other hand, it guides the surrogate model to focus on the potential region during the evolutionary process where the parent population and offspring population reside. One advantage of finding diverse and better solutions is that the quality of the meta-training set for `meta-learning` phase is improved, thus better facilitating to optimize the new environment. In contrast, BO relies on random sampling to evaluate a large number of points in the search space to find good solutions, which becomes infeasible as the dimensionality increases. It is interesting to note that `MLSACMA-ES` achieves better performance than `MLSAPSO` and `MLSADE` in 10-th dimension. One possible reason for this is that `MLSACMA-ES` uses a covariance matrix adaptation strategy to efficiently search for the optimal solution. This makes it less susceptible to getting stuck in local optima compared to `MLSADE` and `MLSAPSO`.

To facilitate an explicit analysis of ranking among these algorithm instances, the Scott-Knott test is used to classify them into different groups based on their performance on the MPB problems. All the Scott-Knott test results on different dimensions are pulled together, and the distribution and median are presented as box plots in Figs. 7. From this result, it is evident that `MLSACMA-ES` is the superior algorithm in our proposed framework, as it consistently ranks in the first place regardless of GPR or NN is used as the surrogate model.

Furthermore, this subsection also investigates the performance using different surrogate models in the proposed method. Based on the results obtained in the previous subsection, we evaluate the A_{12} effect size of E_{BBC} between `MLDDEO` with regard to GPR and NN, respectively. We calculate the percentage of different effect sizes obtained by a pair of dueling algorithms. From the results depicted in Figs. 8, it is evident that using GPR as the surrogate model consistently outperforms that of NN. In particular, all comparison results for three SAEAs have a large effect size when employing GPR as the surrogate



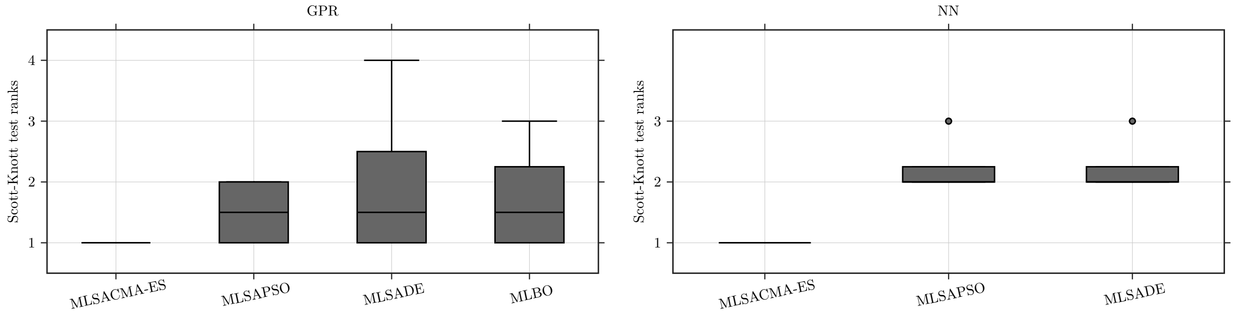
Figs. 5. Loss function of the mean errors between the true objective functions and the best objective values obtained by the proposed algorithm instances using GPR as the surrogate model along with a confidence level over time at different dimensions with $h_{sev} = 7.0$ and $x_{sev} = 5.0$.



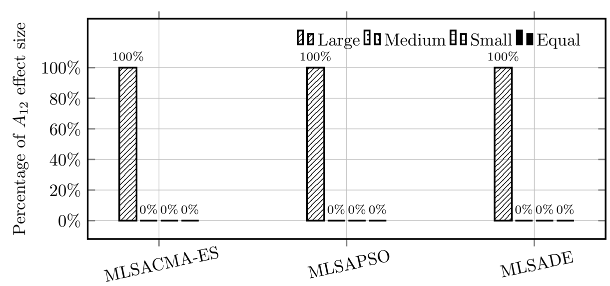
Figs. 6. Loss function of the mean errors between the true objective functions and the best objective values obtained by the proposed algorithm instances using NN as the surrogate model along with a confidence level over time at different dimensions with $h_{sev} = 7.0$ and $x_{sev} = 5.0$.

model in contrast to using NN. Overall, these comparison results suggest that GPR is more capable of modeling objective functions than NN. Nevertheless, as introduced in Section III-D, the computational cost for training a GPR model is higher than that of NN.

Response to concern 5: We have the following takeaways from the empirical study. 1) All proposed algorithm instances show competitive results in both optimization mechanisms for solving relatively low-dimensional problems. When addressing higher-dimensional problems, `MLDDEO` outperforms `MLBO`, and `MLSACMA-ES` is the best algorithm instance



Figs. 7. Box plots of Scott-Knott test ranks of E_{BBC} achieved by each algorithm instance of the proposed framework by using GPR and NN with $h_{sev} = 7.0$ and $x_{sev} = 5.0$, respectively (the smaller rank is, the better performance achieved).



Figs. 8. Percentage of A_{12} effect size of E_{BBC} when comparing `MLDDEO` with GPR as the surrogate model against NN.

in our proposed framework. 2) The population-based search strategy of SAEAs enables them to produce a higher quality meta-training set for the `meta-learning` phase, which enhances the optimization of the new environment, particularly in high-dimensional ones. 3) `MLSACMA-ES` employs a covariance matrix adaptation strategy for effective optimal solution search, making it less prone to trapping in local optima compared to `MLSADE` and `MLSAPSO`. This observation is consistent when using both GPR and NN as the surrogate model. 4) Using GPR as the surrogate model is more capable than NN in our proposed `MLDDEO` framework. The primary drawback of GPR is its cubic worst-case time complexity for model building.

References

[1] J. Branke, "Memory enhanced evolutionary algorithms for changing optimization problems," in *CEC'99: Proc. of the 1999 Congress on Evolutionary Computation*, vol. 3, 1999, pp. 1875–1882.

[2] S. Morales-Enciso and J. Branke, "Tracking global optima in dynamic environments with efficient global optimization," *Eur. J. Oper. Res.*, vol. 242, no. 3, pp. 744–755, 2015.

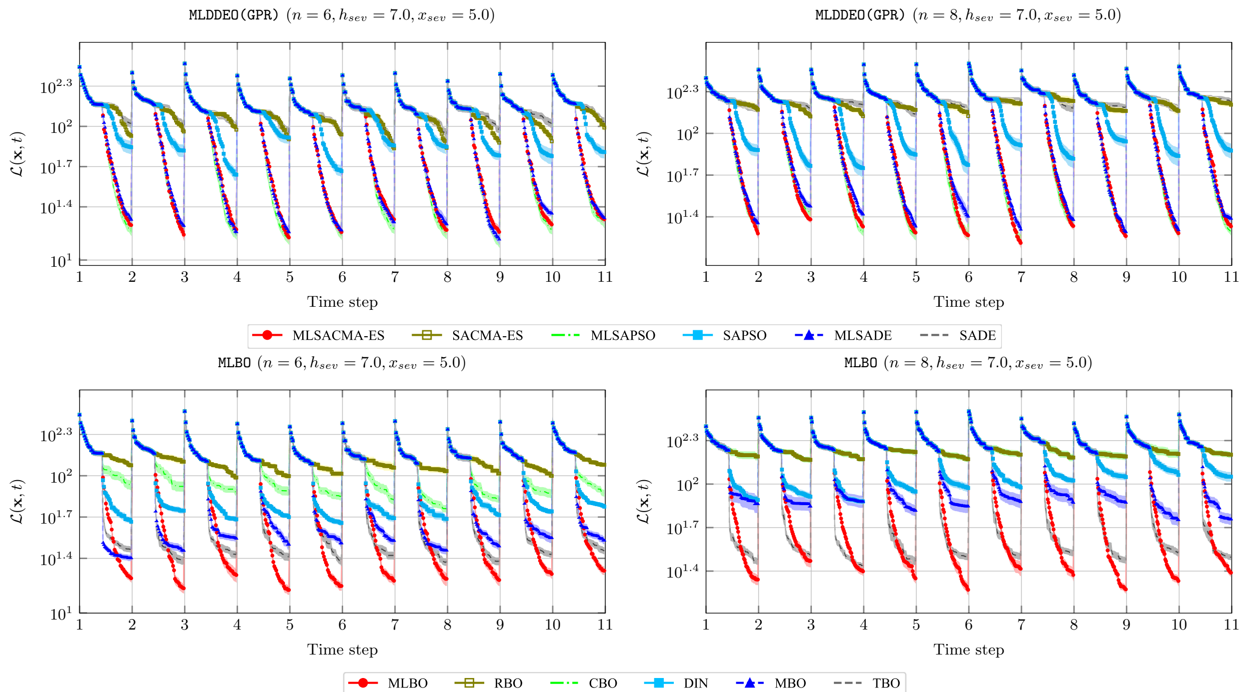
[3] J. Richter, J. Shi, J. Chen, J. Rahnenführer, and M. Lang, "Model-based optimization with concept drifts," in *GECCO'20: Proc. of the 2020 Genetic and Evolutionary Computation Conference*. ACM, 2020, pp. 877–885.

[4] R. Chen and K. Li, "Transfer bayesian optimization for expensive black-box optimization in dynamic environment," in *2021 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2021, Melbourne, Australia, October 17-20, 2021*. IEEE, 2021, pp. 1374–1379.

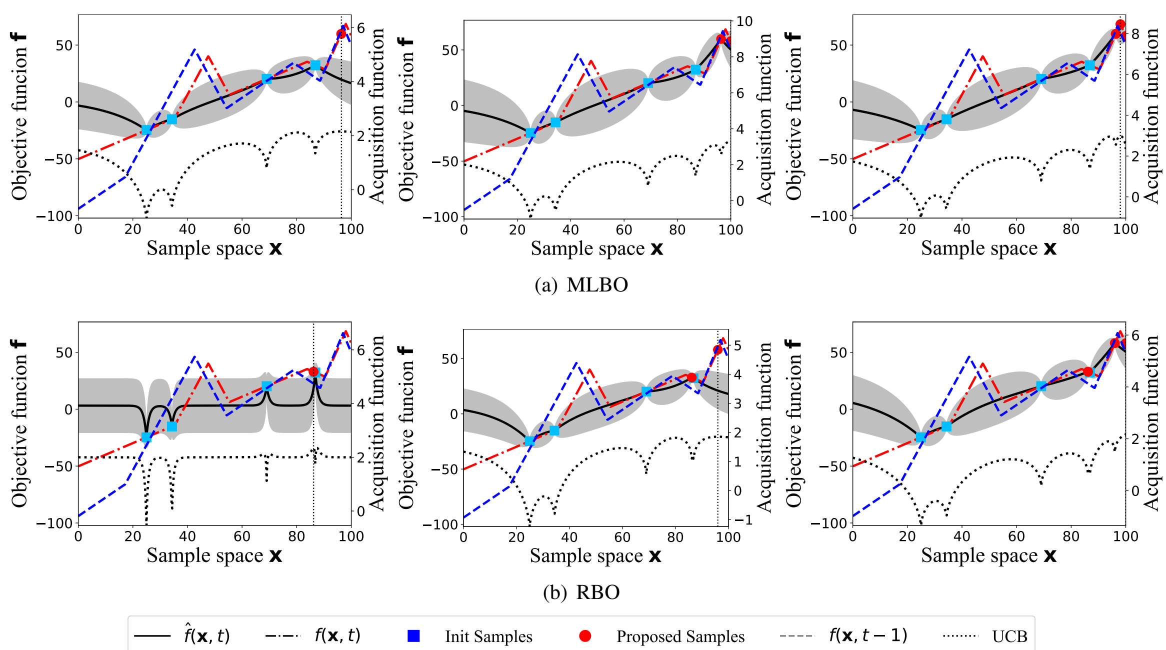
[5] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull*, vol. 1, no. 6, pp. 80–83, 1945.

[6] N. Mittas and L. Angelis, "Ranking and clustering software cost estimation models through a multiple comparisons algorithm," *IEEE Trans. Software Eng.*, vol. 39, no. 4, pp. 537–551, 2013.

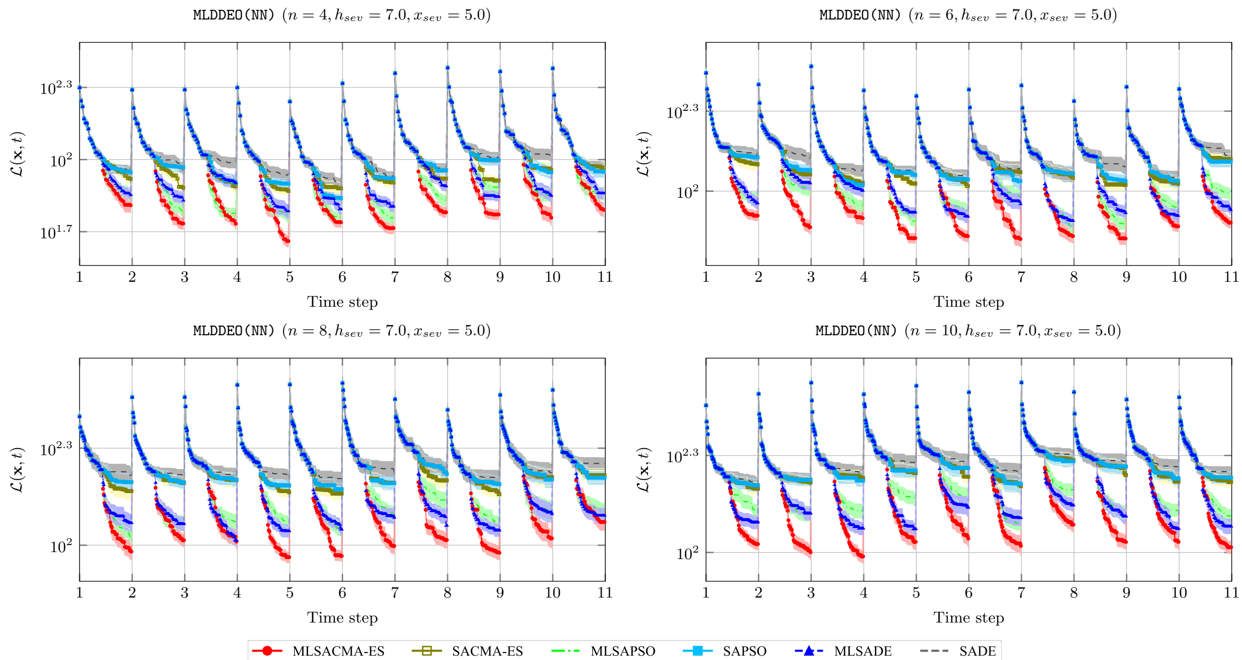
[7] A. Vargha and H. D. Delaney, "A critique and improvement of the cl common language effect size statistics of mcgraw and wong," *Journal of Educational and Behavioral Statistics*, vol. 25, pp. 101–132, 2000.



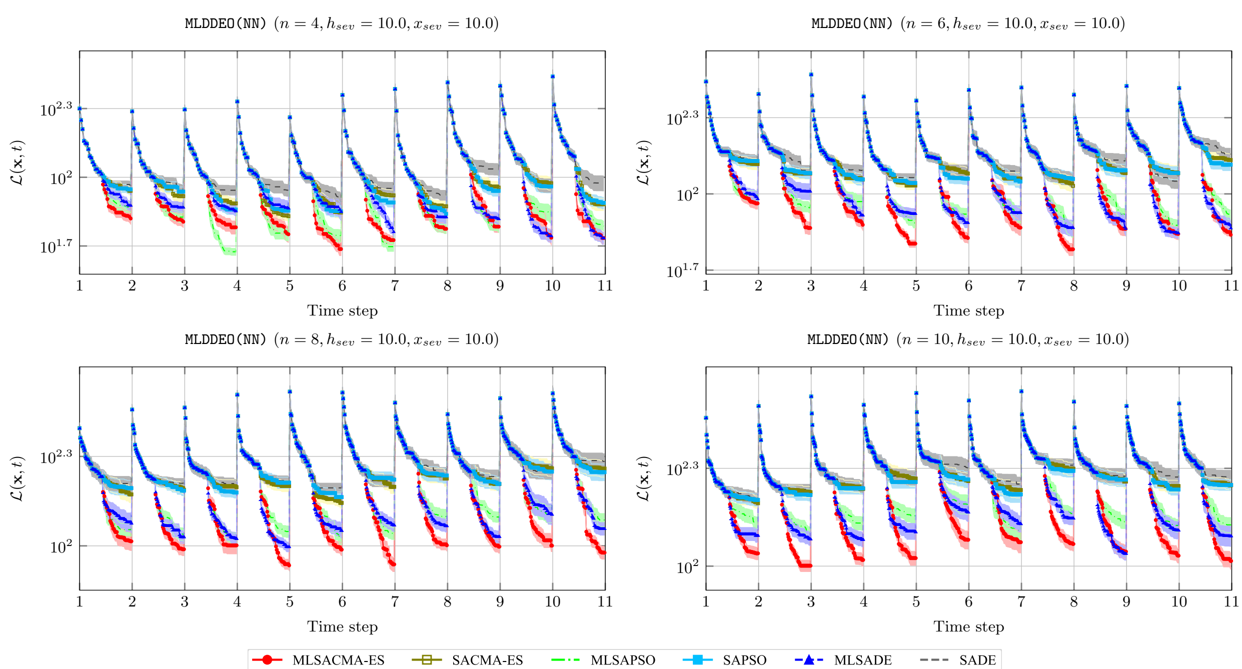
Figs. 9. Loss function of the mean errors between the true objective functions and the best objective values along with a confidence level over time at different dimensions with $h_{sev} = 7.0$ and $x_{sev} = 5.0$ when comparing MLDDEO by using GPR and MLBO with other peer algorithms, respectively.



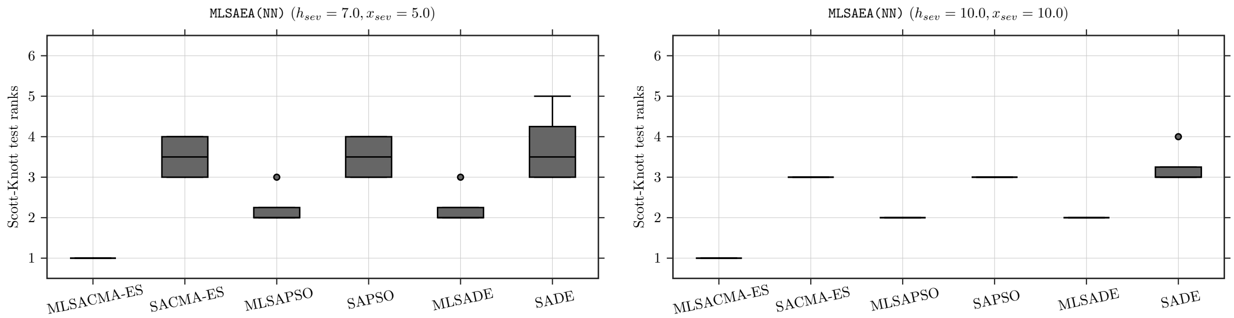
Figs. 10. Illustrative example of the search dynamics of MLBO and RBO across three new samples after environmental changes in the first scenario.



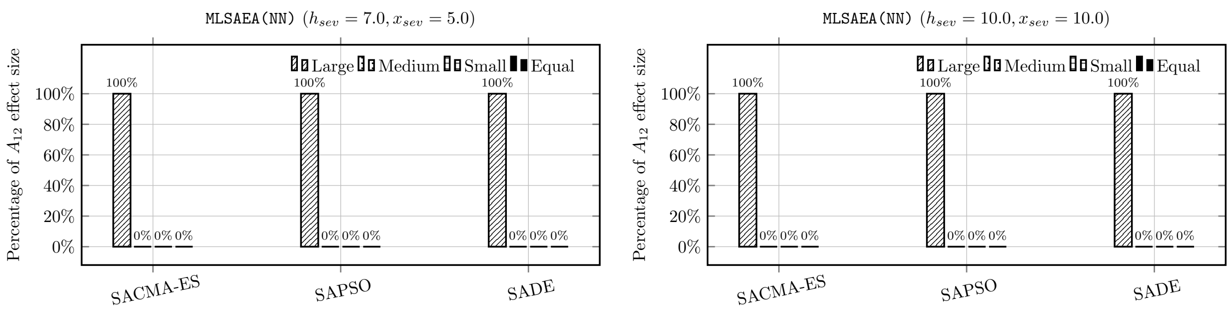
Figs. 11. Loss function of the mean errors between the true objective functions and the best objective values along with a confidence level over time at different dimensions with $h_{sev} = 7.0$ and $x_{sev} = 5.0$ when comparing MLDDEO by using NN with other peer algorithms.



Figs. 12. Loss function of the mean errors between the true objective functions and the best objective values along with a confidence level over time at different dimensions with $h_{sev} = 10.0$ and $x_{sev} = 10.0$ when comparing MLDDEO by using NN with other peer algorithms.



Figs. 13. Box plots of Scott-Knott test ranks of E_{BBC} achieved by MLDDEO using NN as the surrogate model compared against the corresponding SAEAs (the smaller rank is, the better performance achieved).



Figs. 14. Percentage of A_{12} effect size of E_{BBC} when comparing MLDDEO with corresponding SAEAs by using NN.

TABLE II

Performance Comparison Results of E_{BBC} Obtained by the Proposed Algorithm Instances with regard to Other Peers on the MPB Problems. In Particular, Gaussian Process Regression (GPR) Is Used as the Surrogate Model.

E_{BBC}	n=4		n=6		n=8		n=10	
	(7, 5)	(10, 10)	(7, 5)	(10, 10)	(7, 5)	(10, 10)	(7, 5)	(10, 10)
MLSACMA-ES	1.84E+1(8.14E+0)†	1.73E+1(6.96E+0)†	1.77E+1(8.23E+0)†	1.83E+1(7.38E+0)†	**2.01E+1(6.43E+0)**	**2.04E+1(6.97E+0)**	**1.90E+1(5.99E+0)**	**2.08E+1(1.01E+1)**
SACMA-ES	2.77E+1(9.40E+0)‡	2.77E+1(1.11E+1)‡	7.94E+1(2.96E+1)‡	7.85E+1(3.11E+1)‡	1.46E+2(5.55E+1)‡	1.46E+2(4.91E+1)‡	1.67E+2(5.32E+1)‡	1.69E+2(5.31E+1)‡
MLSAPSO	**1.72E+1(7.71E+0)**	1.79E+1(7.40E+0)†	**1.64E+1(7.98E+0)**	2.60E+1(9.19E+0)‡	2.12E+1(7.11E+0)†	3.76E+1(1.26E+1)‡	2.51E+1(8.66E+0)‡	4.52E+1(1.47E+1)‡
SAPSO	3.70E+1(1.47E+1)‡	3.56E+1(1.52E+1)‡	6.24E+1(2.40E+1)‡	6.82E+1(2.53E+1)‡	7.01E+1(3.29E+1)‡	9.21E+1(3.97E+1)‡	7.79E+1(2.48E+1)‡	1.11E+2(3.41E+1)‡
MLSADE	1.76E+1(7.86E+0)†	1.71E+1(7.00E+0)†	1.86E+1(7.84E+0)†	1.98E+1(7.97E+0)†	2.48E+1(9.03E+0)†	2.41E+1(8.34E+0)†	2.38E+1(8.08E+0)‡	2.17E+1(7.71E+0)†
SADE	3.26E+1(1.50E+1)‡	3.26E+1(1.50E+1)‡	8.90E+1(3.80E+1)‡	8.90E+1(3.80E+1)‡	1.47E+2(5.28E+1)‡	1.47E+2(5.28E+1)‡	1.72E+2(5.54E+1)‡	1.72E+2(5.54E+1)‡
MLBO	1.76E+1(7.68E+0)†	**1.70E+1(7.03E+0)**	1.73E+1(8.68E+0)†	**1.77E+1(7.30E+0)**	2.35E+1(5.94E+0)‡	2.36E+1(6.83E+0)†	2.56E+1(7.11E+0)‡	3.36E+1(3.99E+1)‡
RBO	2.83E+1(1.19E+1)‡	2.63E+1(1.06E+1)‡	1.04E+2(4.06E+1)‡	1.06E+2(4.23E+1)‡	1.56E+2(5.34E+1)‡	1.60E+2(5.22E+1)‡	1.70E+2(5.72E+1)‡	1.75E+2(5.81E+1)‡
CBO	4.78E+1(1.41E+1)‡	5.60E+1(2.12E+1)‡	7.17E+1(3.39E+1)‡	8.34E+1(3.61E+1)‡	1.56E+2(5.35E+1)‡	1.60E+2(5.23E+1)‡	1.70E+2(5.72E+1)‡	1.75E+2(5.81E+1)‡
DIN	3.06E+1(1.12E+1)‡	3.44E+1(9.00E+0)‡	4.78E+1(1.57E+1)‡	5.00E+1(1.76E+1)‡	9.00E+1(3.93E+1)‡	8.93E+1(3.21E+1)‡	1.12E+2(3.46E+1)‡	1.18E+2(3.64E+1)‡
MBO	3.06E+1(9.71E+0)‡	4.45E+1(1.93E+1)‡	3.03E+1(1.16E+1)‡	4.70E+1(1.36E+1)‡	6.83E+1(6.31E+1)‡	7.87E+1(5.16E+1)‡	1.57E+2(5.64E+1)‡	1.59E+2(6.14E+1)‡
TBO	2.42E+1(1.01E+1)‡	2.42E+1(9.00E+0)‡	2.60E+1(1.03E+1)‡	2.76E+1(9.52E+0)‡	3.14E+1(9.13E+0)‡	3.22E+1(9.55E+0)‡	3.06E+1(9.02E+0)‡	3.20E+1(8.03E+0)‡

TABLE III

Performance Comparison Results of E_{BBC} Obtained by the Proposed Algorithm Instances with regard to Other Peers on the MPB Problems. In Particular, a Neural Network (NN) Is Used as the Surrogate Model.

E_{BBC}	n=4		n=6		n=8		n=10	
	(7, 5)	(10, 10)	(7, 5)	(10, 10)	(7, 5)	(10, 10)	(7, 5)	(10, 10)
MLSACMA-ES	**5.74E+1(2.19E+1)**	**5.89E+1(2.42E+1)**	**7.52E+1(2.65E+1)**	**7.62E+1(2.89E+1)**	**1.05E+2(3.66E+1)**	**1.00E+2(3.11E+1)**	**1.10E+2(3.41E+1)**	**1.12E+2(3.69E+1)**
SACMA-ES	7.98E+1(2.73E+1)‡	7.94E+1(2.94E+1)‡	1.14E+2(4.42E+1)‡	1.18E+2(4.90E+1)‡	1.53E+2(5.44E+1)‡	1.60E+2(5.51E+1)‡	1.67E+2(5.00E+1)‡	1.74E+2(5.36E+1)‡
MLSAPSO	6.60E+1(2.49E+1)†	6.14E+1(2.31E+1)†	8.99E+1(3.27E+1)†	8.61E+1(3.16E+1)†	1.23E+2(4.35E+1)‡	1.19E+2(3.67E+1)‡	1.39E+2(4.24E+1)‡	1.39E+2(4.43E+1)‡
SAPSO	8.53E+1(2.91E+1)‡	7.91E+1(2.76E+1)‡	1.16E+2(4.50E+1)‡	1.19E+2(4.74E+1)‡	1.57E+2(5.27E+1)‡	1.62E+2(5.39E+1)‡	1.72E+2(5.34E+1)‡	1.74E+2(5.16E+1)‡
MLSADE	7.08E+1(2.40E+1)†	6.78E+1(2.55E+1)†	9.08E+1(3.60E+1)†	8.51E+1(3.43E+1)†	1.19E+2(3.93E+1)†	1.16E+2(3.97E+1)†	1.29E+2(3.74E+1)†	1.29E+2(4.27E+1)†
SADE	9.12E+1(3.08E+1)‡	8.94E+1(3.50E+1)‡	1.23E+2(4.97E+1)‡	1.24E+2(4.82E+1)‡	1.65E+2(5.83E+1)‡	1.67E+2(5.78E+1)‡	1.79E+2(5.60E+1)‡	1.81E+2(5.71E+1)‡

TABLE IV
PERFORMANCE COMPARISON RESULTS OF E_{BBC} OBTAINED BY MLBO WITH DIFFERENT K AT $h_{sev} = 7$ AND $x_{sev} = 5$.

n	$K = 5$	$K = 1$	$K = 15$	$K = 30$	$K = 50$
4	1.76E+1(7.68E+0)	**1.69E+1(7.58E+0)**†	1.76E+1(7.85E+0)†	1.73E+1(7.97E+0)†	1.75E+1(7.86E+0))†
6	1.73E+1(8.68E+0)	2.13E+1(7.97E+0)†	1.71E+1(8.39E+0)†	1.72E+1(8.33E+0)†	**1.69E+1(8.26E+0))**†
8	**2.35E+1(5.94E+0)**	1.39E+2(4.92E+1)‡	2.51E+1(7.61E+0)†	2.57E+1(7.90E+0)†	2.67E+1(9.29E+0))†
10	**2.56E+1(7.11E+0)**	1.57E+2(5.05E+1)‡	2.87E+1(9.48E+0)†	3.21E+1(1.02E+1)‡	3.49E+1(1.22E+1))‡

‡ and † indicate $K = 5$ performs significantly better than and equivalently to the corresponding algorithm, respectively.

TABLE V
PERFORMANCE COMPARISON RESULTS OF E_{BBC} OBTAINED BY MLSACMA-ES (NN) WITH DIFFERENT ξ AT $h_{sev} = 7$ AND $x_{sev} = 5$.

n	$\xi = 5$	$\xi = 1$	$\xi = 10$
4	6.13E+1(2.22E+1)	7.30E+1(2.49E+1)†	**5.74E+1(2.19E+1)**†
6	7.65E+1(2.73E+1)	9.51E+1(3.99E+1)†	**7.52E+1(2.65E+1)**†
8	**1.04E+2(3.39E+1)**	1.27E+2(4.67E+1)‡	1.05E+2(3.66E+1)†
10	1.16E+2(4.01E+1)	1.40E+2(4.43E+1)†	**1.10E+2(3.41E+1)**†