

A Gaussian Process Assisted Offline Estimation of Multivariate Gaussian Distribution Algorithm

Xin-Xin Ma

School of Computers Science &
Engineering
South China University of Technology
GuangZhou, China
maxinxinscut@163.com

Wei-Neng Chen

School of Computers Science &
Engineering
South China University of Technology
GuangZhou, China
cwnraul634@aliyun.com

Qiang Yang

School of Artificial Intelligence
Nanjing University of Information
Science and Technology
Nanjing, China

Abstract—Surrogated assisted evolutionary algorithms are commonly used to solve real-world expensive optimization problems. However, in some situations, no online data is available during the evolution process. In this situation, we have to build surrogate models based on offline historical data, which is known as offline data-driven optimization. Since no new data can be used to improve the surrogate models, offline data-driven optimization remains a challenging problem. In this paper, we propose a Gaussian process assisted offline estimation of multivariate Gaussian distribution algorithm to address the offline data-driven optimization problem. Instead of using surrogate models to predict the fitness values of individuals, we utilize a surrogate model to predict the rankings of individuals based on the frequently used lower confidence bound. In this way, the robustness of the proposed algorithm could be enhanced. Experiments are conducted on five commonly used benchmark problems. The experimental results demonstrate that the proposed offline surrogate model and the multivariate Gaussian estimation of distribution algorithm are able to achieve competitive performance.

Keywords—offline data-driven optimization, surrogate models, Gaussian Regression Process, multivariate Gaussian distribution, estimation of distribution algorithms

I. INTRODUCTION

Evolutionary algorithms (EAs) have become one of the most important optimization techniques and have been successfully applied to solve many complex optimization problems in real world[1-5]. As EAs involve a population of individuals evolving generation by generation to approximate the global optimum of an optimization problem, it usually requires a lot of objective function evaluations. However, in many real-world optimization problems, the evaluation of objective function is complicated and time-consuming, or even requires physical simulations. Such problems are called computational expensive optimization problems. In these

problems, it becomes impossible for EAs to perform objective function evaluations for all individuals during all generations. To address these problems more efficiently, surrogate assisted evolutionary algorithms (SAEAs) have been proposed [6-8]. Different from traditional EAs, SAEAs use historical data to build surrogate models and use these models to estimate the fitness of individuals instead of using the computational expensive objective function, thus the search efficiency of SAEAs can be significantly improved [6].

Generally speaking, SAEAs consist of two parts, namely an optimizer and a surrogate model. In the literature, a variety of EAs have been used as the optimizer, such as genetic algorithms (GA) [9], differential evolution (DE) [10], particle swarm optimization (PSO) [11], etc.

Recently, estimation of distribution algorithms (EDAs) have emerged as a promising EA[12, 13]. Different from traditional EAs, EDAs work by building probability distributions based on the spatial distribution of some good individuals in the search space, and sampling a new generation of individuals based on these probability distributions. With this characteristic, EDAs are also classified as a type of model-based optimization algorithms. Traditionally, Gaussian distributions, Cauchy distributions, histogram, etc. are used as the probability model in EDAs. The covariance matrix adaptation-evolution strategy (CMA-ES) is also a kind of state-of-the-art EDAs [14]. Particularly, CMA-ES is a ranking-based search algorithm.

As a type of optimizer, EDAs can also be utilized in SAEAs. Compared with other EAs, EDAs may have the following additional advantages in SAEAs. First, since the objective function evaluation in SAEAs is based on surrogate models instead of the real objective function, it is usually not very accurate and includes some uncertainty. Meanwhile, probability distribution is an important tool for handling uncertainty. Hence, EDAs may have advantages to cope with the problems without accurate objective function evaluations. Second, while the surrogate models in SAEAs are used to model the landscape of the fitness function of the problem, the probability distribution models in EDAs are used to model the distribution of promising individuals in the search space. It is possible to investigate the relationship of these two types of models to further improve performance of EAs. Third, the probability distributions built in an EDA is actually a kind of

This work was supported in part by the Key Project of Science and Technology Innovation 2030 supported by the Ministry of Science and Technology of China (Grant No. 2018AAA0101300), in part by the National Natural Science Foundation of China under Grants 61976093 and 61772142, in part by the Science and Technology Plan Project of Guangdong Province 2018B050502006, and in part by Guangdong Natural Science Foundation Research Team 2018B030312003. Corresponding author: Wei-Neng Chen (cwnraul634@aliyun.com)

search memory of the individuals. In some specific problems, it is also possible to incorporate prior knowledge with principled techniques to modify the distribution. In this way, EDAs have shown promising performance on some complex optimization problems [12, 13].

Though EDAs may be promising to be incorporated as the optimizer in SAEAs, only a few studies considered the use of EDA in SAEAs. In the literature, CMA-ES has been used as the optimizer in SAEAs, combining with either a global or some local surrogate models [15]. For example, Kern et al. proposed a local meta-model CMAES (Imm-CMA) [16], which uses a set of local quadratic models. Hansen [17] designed a relatively simple global linear-quadratic surrogate approach for CMA-ES.

Most researches on SAEAs treat the optimizer and the surrogate model as two independent parts. In fact, the model in EDAs can provide information for the surrogate model in principle. Besides, exploring deep connections between the optimizer and the surrogate model can balance these two parts naturally and save computational cost by sharing some parameters of models. To this end, in this paper we intend to design a Gaussian process assisted offline estimation of multivariate Gaussian distribution algorithm. Specifically, we adopt the multivariate Gaussian distribution algorithm as the optimizer and use the Gaussian process (GP) modeling [18, 19] as the surrogate to solve offline data-driven optimization problems. In particular, instead of using the surrogate model to predict the fitness value of individuals, we utilize the surrogate model to predict the ranking of individuals based on the commonly-used lower confidence bound. In this way, the robustness of the EDA-based SAEAs can be improved.

The remainder of this paper is organized as follows. In Section 2, we provide the background of offline data driven optimization. In Section 3, we introduce our proposed method in detail. Section 4 conducts extensive experiments to validate the proposed algorithm. Finally, we draw a conclusion in Section 5.

II. BACKGROUND

A. Offline Data-Driven Optimization

In traditional EAs, the evaluation of objective function is called for every individual in every generation. However, in many real-world applications, the evaluation of objective function is usually very computationally expensive. Even worse, in some situations, there is no exact mathematical model to characterize the optimization problems and only data can be used to build a model to evaluate the fitness values of individuals.

Generally, the methods that build surrogate models based on data to characterize the optimization problems are named data-driven evolutionary algorithms [20]. According to whether to use new data in the evolution process to improve surrogate models, data-driven evolutionary algorithms can be divided into 1) online data-driven algorithms and 2) offline data-driven algorithms. The former usually utilize the new generated data in the evolution process to improve the quality of surrogate models, while the latter have no access to use new data to update the agent model in the iterative process of

evolutionary algorithms. Since offline data-driven algorithms are unable to improve the surrogate models, they are generally more challenging.

Since many practical problems only have historical data and are unable to update data in real time, offline data-driven optimization is more common in real-world applications. The main challenge of offline data-driven optimization is how to effectively make use of limited data to build high-quality surrogate models. A recent work [21] by Wang et al. designed an offline data-driven evolutionary algorithm using the ensemble learning method in machine learning area. The ensemble learning method aims to improve the accuracy of the surrogate model to assist the evolutionary algorithm to evolve efficiently.

B. Gaussian Process Modeling

The Gaussian regression process, also known as Kriging interpolation, is a stochastic approximation method [12]. A universal Kriging model assumes that the fitness function can be approximated as:

$$Y(\mathbf{x}) = \mu(\mathbf{x}) + \epsilon(\mathbf{x}) \quad (1)$$

where $\mu(\mathbf{x})$ is the trend function and $\epsilon(\mathbf{x})$ is the error term with mean $\mathbf{0}$ and covariance matrix $\sigma^2 \phi$.

The covariance is computed as follows:

$$\text{cov}(Y(\mathbf{x} + \mathbf{h}), Y(\mathbf{x})) = \sigma^2 \phi(\mathbf{h}) \quad (2)$$

where $\text{cov}(Y(\mathbf{x} + \mathbf{h}), Y(\mathbf{x}))$ is the correlation function.

In general, the parameters above can be determined by maximizing the likelihood function for input datapoints.

$$\frac{1}{(2\pi\sigma^2)^{K/2} \sqrt{\det(\mathbf{C})}} \exp \left[-\frac{(\mathbf{y} - \mu)^T \mathbf{C}^{-1} (\mathbf{y} - \mu)}{2\sigma^2} \right] \quad (3)$$

where $\mathbf{C} = \sigma^2 \phi$ is the covariance matrix.

1) *Ordinary Kriging*: In reality, the trend function $\mu(\mathbf{x})$ is very difficult to capture. Therefore, the following special case is widely used and known as the ordinary kriging:

$$Y(\mathbf{x}) = \mu_0 + \epsilon(\mathbf{x}) \quad (4)$$

2) *Blind Kriging*: The blind kriging is a modified variant of the universal kriging. As often, the trend function is not prior in the blind kriging. Therefore, it usually tries to select variables to construct the trend function through forward Bayesian process [11].

$$Y(\mathbf{x}) = \mathbf{v}(\mathbf{x})' \mu_m + \epsilon(\mathbf{x}) \quad (5)$$

where $(\mathbf{x})' = (\mathbf{1}, \mathbf{v}_1, \dots, \mathbf{v}_m)$, $\mu_m = (\mu_0, \mu_1, \dots, \mu_m)$ and m is the number of selected variables.

III. THE PROPOSED METHOD

A. Multivariate Gaussian Distribution Estimation

To accommodate with the Gaussian regression model, we adopt the multivariate Gaussian distribution algorithm as the search engine in our proposed SAEA. In particular, the Gaussian distribution is widely observed in nature and the central limit theorem provides the theoretical basis for the applicability of the multivariate Gaussian distribution. Inspired by CMA-ES which is also based on the Gaussian distribution, we utilize the estimated multivariate Gaussian distribution model to generate new candidate solutions for EDA.

Algorithm 1 Pseudo-Code of Multivariate Gaussian Distribution Estimation algorithm for generating offspring

Input: D : database; λ : the number of offspring; n : the number of all data points in the dataset
1: Sort the dataset items according to their fitness;
2: $\bar{x}_D = \frac{\sum_{j=1}^n x_j}{n}$;
3: $\omega_i = \frac{\ln(\mu+1) - \ln i}{\mu \ln(\mu+1) - \sum_{j=1}^{\mu} \ln i}$; //the weight function
4: **For** $j = 1$ to μ **do**
5: $y_j = x_j - \bar{x}_D$;
6: $C = y_{1:\mu} \times \text{diagonal}(\omega_{1:\mu}) \times y_{1:\mu}^T$;
7: **End For**
8: **For** $k=1$ to λ **do**
9: $S_k = \bar{x}_\mu + N(0, C)$;
10: **End For**
Output: S : the offspring set

Specifically, a multivariate Gaussian distribution model can be estimated by a mean vector \mathbf{m} and a covariance matrix \mathbf{C} as Eq. (6).

$$\mathcal{N}(\mathbf{m}, \mathbf{C}) \quad (6)$$

where \mathbf{m} is the modal value of the distribution, and \mathbf{C} is the covariance matrix that determines the orientation of the distribution.

In order to generate more promising offspring, the modal value of the constructed Gaussian distribution is determined by the first ranked μ points, and the covariance matrix is computed based on the difference vectors between the μ points and the mean vector \bar{x}_D of the input dataset. The constructed multivariate Gaussian distribution provides a heuristic search for the promising area. The details of the multivariate Gaussian distribution estimation algorithm are presented in Algorithm 1. For the input database, it is first sorted according to the fitness values of items. The mean value \bar{x}_D of the whole database is calculated and the weight vector ω is set. Then, the first ranked μ data points are combined via the weight vector ω . The covariance matrix is then calculated as shown in lines 4-7. After the estimation of the mean vector and the covariance matrix, λ offspring are generated by the constructed multivariate Gaussian distribution model.

B. The Proposed Framework

For the accuracy of the surrogate model, only the data with real fitness is used to build the surrogate model. To initialize the database, we sample enough offline data points via the multivariate Gaussian distribution. For example, for the D-dimension Griewank function with a search space $[-600, 600]$, we sample points by the normal distribution with mean vector $[10, \dots, 10]$, standard deviation 600 and the identity matrix as the covariance matrix. Particularly, $11 \times D$ points are sampled and evaluated by the real objective function to initialize the database. Considering that the minimum function value of the Griewank function is 0, we use $[10, \dots, 10]$ as the mean vector to reduce bias. In this work, we do not use the popular Latin hypercube sampling (LHS) method because 1) the estimation of multivariate Gaussian distribution works better on a prior normal distribution; and 2) the offline data in the real-world is hardly evenly distributed.

The specific algorithm flow is shown in Fig. 1 and the main steps are listed as follows:

- Step 1:** Use the normal distribution sampling to get $11 \times D$ points and use the real objective function to evaluate these points. These points constitute the initial database.
- Step 2:** Establish a global GP surrogate model based on the initial database.
- Step 3:** Add all the points in the initial database to the hybrid database.
- Step 4:** Estimate the multivariate Gaussian distribution of the hybrid data set and generate λ offspring.
- Step 5:** Evaluate the offspring using the surrogate model and add the most promising solutions into the hybrid database.
- Step 6:** Repeat Steps 4 and 5 until the convergence condition is satisfied.

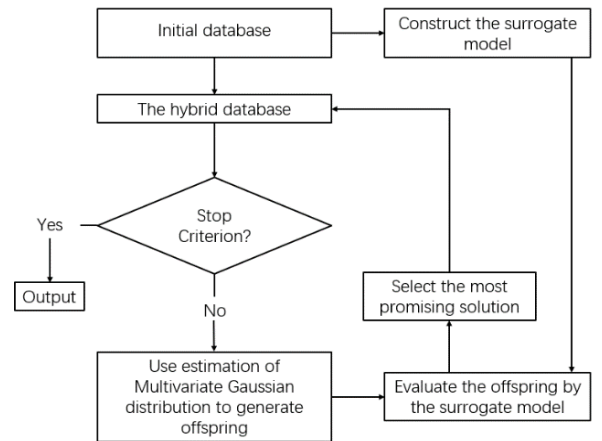


Fig. 1. The flow diagram of the proposed framework

There are two remarks for the proposed framework: 1) the prediction of the Gaussian process regression model gives the estimated distribution of prediction value at the evaluated

point. Specifically, it is a normal distribution with mean value $\hat{f}(x)$ and variance $s(x)$. The commonly used lower confidence bound (LCB) presented below is employed to evaluate the quality of the solutions. The more promising region has a low predicted mean value while the less explored region has a high variance. Indexed by LCB, the algorithm can realize the trade-off between the exploitation and exploration in the evolution process.

$$f_{lcb}(x) = \hat{f}(x) - \omega s(x) \quad (7)$$

where ω is set as 2 in this paper; 2) the evaluation of the surrogate model is only used for sorting the solutions. Therefore, this increases the robustness of the algorithm.

IV. EXPERIMENTS

In this section, the performance of the developed offline algorithm is assessed via experiments conducted on five commonly used benchmark problems. Particularly, all the benchmark problems in the experiments contain 10 dimensions. The number (λ) of offspring generated by the estimation of multivariate Gaussian distribution is set as 50 and the number (μ) of elite solutions used to estimate multivariate Gaussian distribution is set as 10. As for the surrogate model implementation, we utilize the well-documented blind dace toolbox to build surrogate model [4]. In the following, we will first introduce the used five benchmark problems and then we will investigate the performance of the proposed offline algorithm in four aspects: 1) convergence curves of the blind kriging assisted offline algorithm, 2) comparison between different surrogates (the ordinary kriging and the blind kriging) on convergence curves, 3) comparison between different surrogates on the performance of the five benchmark problems, and 4) comparison between two version GPEME on the Five Benchmark Problems

A. Benchmark Problems

In this paper, we utilize the following five widely used benchmark problems to validate the proposed offline algorithm:

1) Ellipsoid Problem

$$\min f(x) = \sum_{i=1}^n ix_i^2$$

$$x \in [-5.12, 5.12], n = 10$$

$$\text{minimum: } f(x^*) = 0$$

2) Rosenbrock Problem

$$\min f(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (a - x_i)^2]$$

$$x \in [-2.048, 2.048], n = 10$$

$$\text{minimum: } f(x^*) = 0$$

3) Ackley Problem

$$\min f(x) = -a \times \exp \left(-b \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(cx_i) \right) + a + \exp(1)$$

$$x \in [-32.768, 32.768], n = 10, a = 20, b = 0.2$$

$$\text{minimum: } f(x^*) = 0$$

4) Griewank Problem

$$\min f(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right)$$

$$x \in [-600, 600], n = 10$$

$$\text{minimum: } f(x^*) = 0$$

5) Rastrigin Problem

$$\min f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$$

$$x \in [-5, 5], n = 10$$

$$\text{minimum: } f(x^*) = 0$$

B. Convergence Curve of the Blind Kriging

Firstly, experiments are carried out to verify whether the offline algorithm can converge to a reasonable solution. We conduct experiments on the 10-dimension Ellipsoid and Griewank problems. The convergence curves of the proposed algorithm are shown in Figs. 2-5. The first 110 black points in the figures (Figs. 2 and 4) are those in the initial database, while the red points are the solutions that are selected into the hybrid database during the iterations of the proposed algorithm.

From these figures, we can see that the proposed offline algorithm converges in less than 120 iterations and the final converged solution has a reasonable real objective value for the two tested problems.

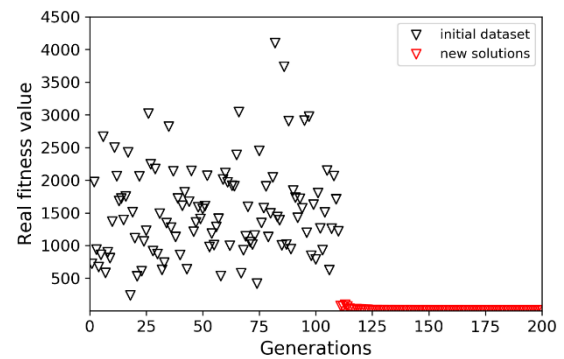


Fig. 2. Convergence curve of the blind kriging assisted offline algorithm on the Ellipsoid problem with points in the initial dataset included.

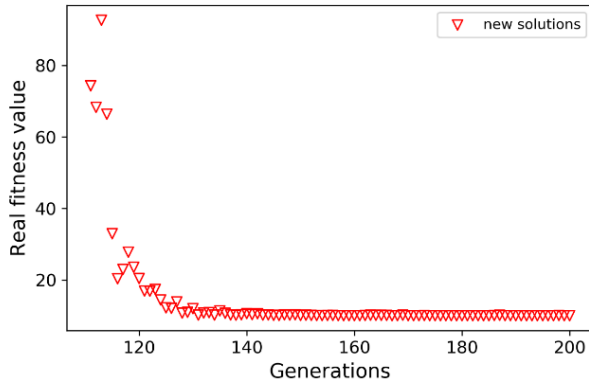


Fig. 3. Convergence curve of the blind kriging assisted offline algorithm on the Ellipsoid problem with points in the initial dataset excluded.

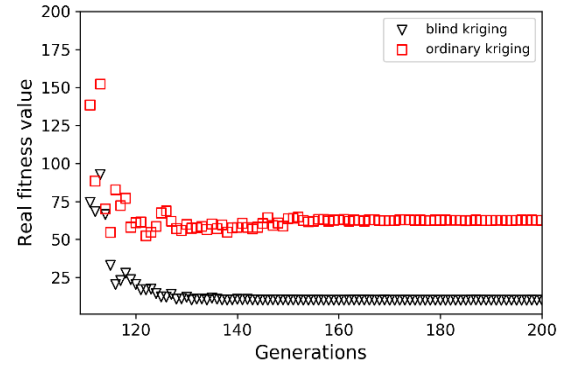


Fig. 6. Convergence curve comparison between the blind kriging and the ordinary kriging assisted offline algorithms on the Ellipsoid problem.

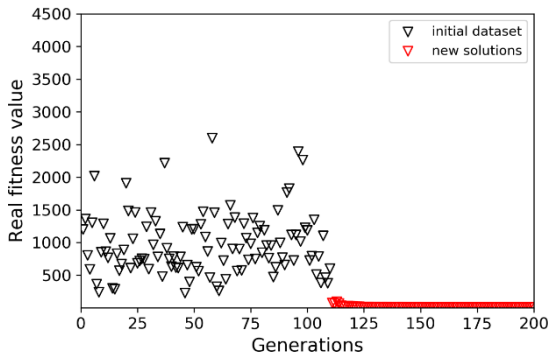


Fig. 4. Convergence curve of the blind kriging assisted offline algorithm on the Griewank problem with points in the initial dataset included.

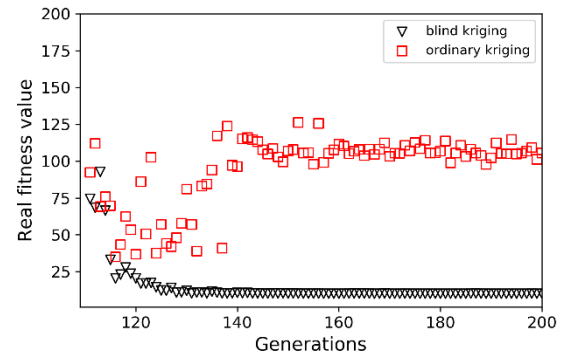


Fig. 7. Convergence curve comparison between the blind kriging and the ordinary kriging assisted offline algorithms on the Griewank problem

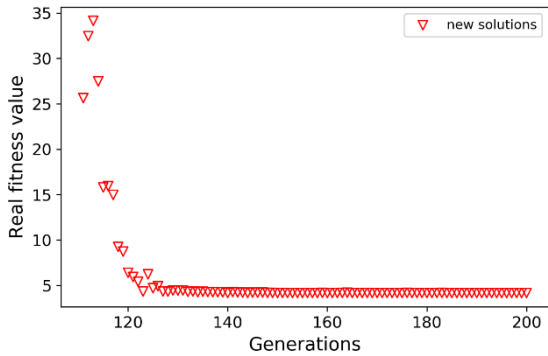


Fig. 5. Convergence curve of the blind kriging assisted offline algorithm on the Griewank problem with points in the initial dataset excluded.

C. Comparison Between Different Surrogates on Convergence Curves

In this section, we observe the performance on the proposed offline algorithm with different surrogate models. Particularly, we utilize the ordinary kriging model and the blind kriging model to assist the proposed offline algorithm to optimize the Ellipsoid problem and the Griewank problem respectively.

The convergence curves of the offline algorithm with the above two kriging models are presented in Figs. 6-7. From these two figures, we can see that the final converged solution of the blind kriging assisted offline algorithm is much better than that of the ordinary kriging assisted one because the final converged solution of the blind kriging assisted offline algorithm is much closer to the optimal solution of the problems.

The above comparison results indicate that the surrogate model plays a vital role in the offline data-driven optimization algorithms. Specifically, a suitable surrogate model can optimize the search process while a poor surrogate model may mislead the search process.

TABLE I. OPTIMIZATION RESULTS OF THE ORDINARY KRIGING ASSISTED OFFLINE ALGORITHM ON THE 5 BENCHMARK PROBLEMS.

Problem	Ackley	Rosenbrock	Griewank	Rastrigin	Ellipsoid
Average	20.2151	2161.2103	39.1689	207.9319	66.6535
Std	0.6642	2546.1635	20.9740	32.0897	24.9303
Best	18.5103	150.5413	10.3185	158.7313	33.2107
Worst	20.8897	10576.656	91.9052	289.2027	110.3896

TABLE II. OPTIMIZATION RESULTS OF THE BLIND KRIGING ASSISTED OFFLINE ALGORITHM ON THE 5 BENCHMARK PROBLEMS

Problem	Ackley	Rosenbrock	Griewank	Rastrigin	Ellipsoid
Average	20.6856	1192.8667	4.2368	fail	12.0913
Std	1.0509	888.8773	2.5377	fail	10.5130
Best	17.3134	151.9561	1.5700	fail	0.3420
Worst	21.9584	3298.6181	9.2648	fail	35.9545

TABLE III. COMPARISON WITH TWO VERSION GPME ON THE FIVE BENCHMARK PROBLEMS

Problem	Ackley	Rosenbrock	Griewank	Rastrigin	Ellipsoid
Our method (Blind Kriging)	20.7 \pm 1.1	1192.9 \pm 888.8	4.2\pm2.5	fail	12.1\pm10.5
GPME online	13.8 \pm 2.5	186.0 \pm 66.9	27.2 \pm 11.3	71.5 \pm 12.7	37.8 \pm 15.3
GPME offline	19.2 \pm 0.5	625.6 \pm 196.7	103.6 \pm 31.4	102.5 \pm 16.6	129.8.0 \pm 34.3

D. Comparison Between Different Surrogates on the Five Benchmark Problems

Finally, we execute the offline algorithm with different surrogates on all the five problems to make a comprehensive comparison. It should be noted that all algorithms are executed independently for 20 times.

The statistical results are summarized in Table I and Table II. It should be noted that in the evolution process of the offline algorithm, no new data is available and used to improve the surrogates. When the algorithm stops, the final solution is evaluated by the real objective function to assess the performance of the offline algorithm.

Comparing Table I with Table II, we can see that 1) on the Griewank, Ellipsoid and Rosenbrock problems, the quality of the final solution of the offline algorithm assisted by the blind kriging is much better than that by the ordinary kriging; 2) on the Rosenbrock problem, the solution quality of the offline algorithms assisted by the two surrogates is not high, because the Rosenbrock problem has a very narrow valley, leading to that the Gaussian regression process is unable to capture this feature well; 3) although the performance of the offline algorithm using the ordinary kriging on most functions is not satisfied, the algorithm is more robust; 4) on the Rastrigin problem, the blind kriging assisted offline algorithm is unable to converge to a good solution. In fact, the real objective value of the obtained solution during iterations is getting worse and worse. But the ordinary kriging assisted one can still converge to a reasonable solution. This comparison reflects that the accuracy of the surrogate model is problem

dependent. There is hardly a general suitable single global surrogate model. Therefore, we need to select an appropriate surrogate model for specific problems.

E. Comparison with two version GPME on the Five Benchmark Problems

Most offline data driven algorithms use ensemble-based model. For example, DDEA-SE proposed by Wang et al. is a state-of-art ensemble-based offline data-driven EAs whose ensemble consists of 2000 RBF models. Since our method is single surrogate-assisted, comparisons to the existed ensemble-based offline data-driven EAs may not be fair. Here we choose an online data-driven EA: GPME, which use the same Gaussian process surrogate model as ours. Online data-driven EAs can be transferred into offline versions by initializing the surrogate using all allowed computational budget and stop once the first real fitness evaluation is required. The reference paper [14] has provided the optimization result of GPME online version and GPME offline version on the five benchmark problems. Here we use the provided data for comparison presented in Table III. From Table III, we can conclude that our method achieve much better results on Griewank and Ellipsoid problem. These verifies the Multivariate Gaussian Distribution Estimation algorithm can capture the characteristic of Griewank and Ellipsoid problem thus achieve a better performance. Also, our method does not perform well on Rosenbrock and Rastrigin problems, this enlighten us to design some strategies to improve the robustness of our method.

V. CONCLUSIONS

In this paper, we have proposed a blind kriging assisted offline estimation of multivariate Gaussian distribution algorithm to tackle expensive optimization problems. Instead of using surrogate models to predict the fitness values of individuals in existing studies, we employ the blind kriging model to predict the rankings of individuals based on the commonly used lower confidence bound. In this way, the robustness of the proposed algorithm could be enhanced. Extensive experiments conducted on five commonly used benchmark problems demonstrate that the proposed offline algorithm assisted by the blind kriging model could achieve competitive or even better performance than another commonly used offline surrogate model (namely the ordinary kriging model) assisted offline algorithm.

In this paper, we have taken the relation between the Multivariate Gaussian Distribution algorithm and the blind kriging model into consideration to optimize problems. However, the considered relation is just a shallow combination of the two models. In the future, we can explore the internal relationship by sharing some hyper parameters of two models. In addition, advanced machine learning methods like transfer learning, are also under consideration to boost the performance of offline data-driven methods.

In future study, it would be interesting to consider EDAs for large-scale data-driven optimization [22]-[24] and complex optimization problems under uncertainty [25].

REFERENCES

- [1] T.-F. Zhao *et al.*, "Evolutionary Divide-and-Conquer Algorithm for Virus Spreading Control Over Networks," *IEEE Transactions on Cybernetics*, in press, 2020.
- [2] Z.-X. Zhang, W.-N. Chen, H. Jin, and J. Zhang, "A Preference Biobjective Evolutionary Algorithm for the Payment Scheduling Negotiation Problem," *IEEE Transactions on Cybernetics*, in press, 2020.
- [3] A. Song, W.-N. Chen, Y.-J. Gong, X. Luo, and J. Zhang, "A divide-and-conquer evolutionary algorithm for large-scale virtual network embedding," *IEEE Transactions on Evolutionary Computation*, in press, 2019.
- [4] W.-N. Chen, Y.-H. Jia, F. Zhao, X.-N. Luo, X.-D. Jia, and J. Zhang, "A cooperative co-evolutionary approach to large-scale multisource water distribution network optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 842-857, 2019.
- [5] Y.-H. Jia *et al.*, "Distributed cooperative co-evolution with adaptive computing resource allocation for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 188-202, 2018.
- [6] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61-70, 2011.
- [7] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft computing*, vol. 9, no. 1, pp. 3-12, 2005.
- [8] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 329-355, 2009.
- [9] T. Back, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.
- [10] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341-359, 1997.
- [11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, 1995, vol. 4: IEEE, pp. 1942-1948.
- [12] Q. Yang *et al.*, "Adaptive multimodal continuous ant colony optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 191-205, 2016.
- [13] Q. Yang, W.-N. Chen, Y. Li, C. P. Chen, X.-M. Xu, and J. Zhang, "Multimodal estimation of distribution algorithms," *IEEE transactions on cybernetics*, vol. 47, no. 3, pp. 636-650, 2016.
- [14] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary computation*, vol. 11, no. 1, pp. 1-18, 2003.
- [15] L. Bajer, Z. Pitra, J. Repický, and M. Holeňa, "Gaussian process surrogate models for the CMA evolution strategy," *Evolutionary computation*, vol. 27, no. 4, pp. 665-697, 2019.
- [16] S. Kern, N. Hansen, and P. Koumoutsakos, "Local meta-models for optimization using evolution strategies," in *Parallel Problem Solving from Nature-PPSN IX*: Springer, 2006, pp. 939-948.
- [17] N. Hansen, "A global surrogate assisted CMA-ES," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2019, pp. 664-672.
- [18] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq, "A surrogate modeling and adaptive sampling toolbox for computer based design," *Journal of Machine Learning Research*, vol. 11, no. Jul, pp. 2051-2055, 2010.
- [19] B. Liu, Q. Zhang, and G. G. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180-192, 2013.
- [20] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: an overview and case studies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 442-458, 2018.
- [21] H. Wang, Y. Jin, C. Sun, and J. Doherty, "Offline data-driven evolutionary optimization using selective surrogate ensembles," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 203-216, 2018.
- [22] Q. Yang, W.-N. Chen, J. Deng, Y. Li, T. Gu and J. Zhang, "A Level-based Learning Swarm Optimizer for Large Scale Optimization", *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 578-594, 2018.
- [23] F.-F. Wei, *et al.*, "A Classifier-Assisted Level-Based Learning Swarm Optimizer for Expensive Optimization", *IEEE Transactions on Evolutionary Computation*, in press, 2020
- [24] Q. Yang, *et al.*, "A Distributed Swarm Optimizer with Adaptive Communication for Large Scale Optimization", *IEEE Transactions on Cybernetics*, in press, 2019.
- [25] W. Shi, *et al.*, "Handling Uncertainty in Financial Decision Making: a Clustering Estimation of Distribution Algorithm with Simplified Simulation", *IEEE Transactions on Emerging Topics in Computational Intelligence*, in press, 2020.