2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing (HPSC), and IEEE Intl Conference on Intelligent Data and Security (IDS)

Asynchronous Parallel Surrogate Optimization Algorithm based on Ensemble Surrogating Model and Stochastic Response Surface Method

Application to quantitative strategy parameter tuning

Sun Yongze
University of Chinese Academy of Sciences
Computer Network Information Center, Chinese Academy of Sciences
Beijing, China
sunyongze@outlook.com

Jue Wang
Computer Network Information Center
Chinese Academy of Sciences
Beijing, China
wangjue@sccas.cn

Zhonghua Lu
Computer Network Information Center
Chinese Academy of Sciences
Beijing, China
zhlu@cnic.cn

***Abstract*—Surrogate model-based optimization algorithm remains as an important solution to expensive black-box function optimization. The introduction of ensemble model enables the algorithm to automatically choose a proper model integration mode and adapt to various parameter spaces when dealing with different problems. However, this also significantly increases the computational burden of the algorithm. On the other hand, utilizing parallel computing resources and improving efficiency of black-box function optimization also require combination with surrogate optimization algorithm in order to design and realize an efficient parallel parameter space sampling mechanism. This paper makes use of parallel computing technology to speed up the weight updating related computation for the ensemble model based on Dempster-Shafer theory, and combines it with stochastic response surface method to develop a novel parallel sampling mechanism for asynchronous parameter optimization. Furthermore, it designs and implements corresponding parallel computing framework and applies the developed algorithm to quantitative trading strategy tuning in financial market. It is verified that the algorithm is both feasible and effective in actual application. The experiment demonstrates that with guarantee of optimizing performance, the parallel optimization algorithm can achieve excellent accelerating effect.**

Surrogate Model; Ensemble Model; Parallel Computing; Quantitative Trading; (key words)

I. INTRODUCTION

A lot of practical problems concerning engineering design and management should be resolved through optimization. The primary goal is to design and choose adequate parameters in order to maximize or minimize certain performance measure. For example, in deep learning model design, we should design in advance proper hyper-parameters such as number of network layers, learning rate and regularization ratio so that the model can achieve better generalization performance; and in auto circuit design, it is necessary to choose proper parameters such as line width and number of pcb layers to generate circuits with outstanding performance and power consumption. However, in a lot of real cases, we can't establish or acquire a explicit expression of relationship between those parameters that should be designed and performance evaluation. The performance indicators of certain parameters can only be determined through experiment or simulation. These problems can be viewed as a black-box optimization problem. More specifically, they may be treated as an expensive black-box global function optimization problem when the experiment or simulation operations of parameter performance indicators demand massive computation cost and time. For instance, in deep learning network design, in order to assess the performance of network design hyper-parameters, it is necessary to adopt hyper-parameter search method base on the dataset to acquire the performance indicators of hyper-parameters. For complex network and large datasets, the performance indicators for each group of hyper-parameters required high computing cost of both time and power.

It is of great significance to study how to solve the expensive black-box function optimization problem on algorithm level for all related problems in various fields. Some conventional black-box function optimization algorithms, such as those based on population theory like genetic algorithm and particle swarm optimization, usually demand too massive function evaluation times and can't be applied to such issues due to overly high expenditure of single-step computation. For those expensive black-box function optimization problems, surrogating model-based algorithms get widely applied. Based on resolved set of parameter space points, they usually employ different models to fit the whole parameter space and then utilize surrogating model to substitute the function evaluation and computing process with expensive emulation and simulation, to predict the distribution of optimal or better solution in the parameter space, improving the efficiency of optimization algorithms, and reduce the necessary function evaluation times.

Various models in practice can act as surrogating models, including Gaussian process, support vector machine and different interpolation models. A significant fact is the more accurately a surrogating model simulates the parameter space, the closer the prediction about unknown areas in the parameter space is to actual situation. Therefore, the optimization algorithm based on corresponding surrogating model can achieve high optimization efficiency. When it comes to different problems and scenarios, due to lack of priori

* To whom correspondence should be addressed. This work has been supported by the National Natural Science Foundation of China (Grant No. 61873254) and the project (SGGR0000JSJS1800569) from State Grid Corporation of China.

978-1-7281-0006-7/19/$31.00 ©2019 IEEE
DOI 10.1109/BigDataSecurity-HPSC-IDS.2019.00024

knowledge about the parameter space, it is impossible to directly choose the most suitable surrogating model for parameter space. When lacking experience and knowledge about a specific practical problem or model, a possible solution is to fit the parameter space with multiple surrogating models at the same time, and then output one or several proper ones for integration according to their performance on the parameter space modeling and increase the ratio of the advantageous ones. In this way, it can be made sure that the optimization algorithm adopts most suitable parameter space surrogating model for different problems and the adaptability of the surrogating model to those problems can be improved as well.

But the introduce of ensemble models makes it necessary to maintain and update multiple surrogating models, and perform computation concerning model selection and weight list constructing. The additional computing load introduced will significantly increase the complexity and running time of the algorithm. Therefore, to fully consider the computing load of ensemble surrogating model and utilize computing resources to speed up of expensive function evaluation, this paper adopts parallel computing technology to accelerate the ensemble surrogating model related computation and designs a asynchronous parallel sampling mechanism for parameter space that build an highly efficient asynchronous parallel solution to the expensive black-box function problems.

In quantitative trading of financial market, the design of strategies usually involves several artificially preset parameters. A common practice is to traverse the whole parameter space through grid search on the basis of historical data back-test and finally choose a group of parameters that maximize the profit acquisition or other profit indicators in the backtest. However, affected by several factors such as higher historical data scale and higher strategic complexity, backtest of single group of parameters takes a lot of time so that the grid search method becomes extremely inefficient. A possible result is that it fails to acquire a group of ideal parameters even after long-time search. This paper views it as an expensive black-box function problem and adopts the algorithm developed this article to solve the parameter optimization problems in quantitative trading strategy.

This paper designs and implements a high-performance asynchronous parallel algorithm based on ensemble surrogating model and stochastic response surface method. Its major contributions are as follows:

1. It designs and implements a parallel algorithm framework based on ensemble surrogating model and utilizes Dempster-Shafer theory to compute and update the weights of various models. The aforesaid framework can be applied to the surrogate optimization algorithms based on various surrogating models and asynchronous parallel sampling strategy, achieves acceleration of ensemble model-related computation and asynchronous parallel sampling of parameter space, improves the performance of algorithm of solving expensive black-box function optimization problem. Whist reducing the time consumption of solving these problems, it is suitable for all sorts of expensive black-box function optimization problems.

2. In this paper, a novel asynchronous parallel sampling strategy based on stochastic response surface algorithm is proposed. It is verified that on the premise of algorithm convergence performance, the strategy can make full use of the parallel computational resources to speed up the solving process of expensive function optimization problems.

3. This paper suggests applying the surrogating model-based black-box function optimization algorithm to quantitative trading strategy tuning in financial market and proves the algorithm is effective in such problems.

II. RELATED WORKS

Surrogating model based optimization algorithms have been widely used in all sorts of expensive black-box function optimization problems. For example, reference [1] used neural network as a surrogating model to improve the efficiency of genetic algorithm in solving aerodynamical design; reference [2] improved efficiency of CMA-ES algorithm with SVM as surrogating model; reference [3] applied deep network to multi-object optimization NSGA-ii. As to the deep learning hyper-parameter optimization, the Bayes optimization algorithm adopting Gaussian process as its surrogating model which has attracted wide attention[4, 5]. It is generally believed that a surrogating model acts to approximate the function value to some extent instead of directly figuring out the real function value through expensive evaluation like experiment and simulation so that the parameter can be assessed through the model without demanding direct solution. This can minimize the function evaluation times, acquire the better solution, and improve the efficiency of the optimization algorithm. However, what kind of surrogating mode should be chosen for a specific problem remains inconclusive.

Ensemble surrogating model sets to fit the parameter space with multiple surrogating models and determines the weight of each model through its predicting performance so that the model with better predicting performance on the parameter space can have higher weight. Therefore, the ensemble model can automatically adapt to different problems. In reference [6], three methods were proposed to determine the weight of surrogating model by the performance measure, but only one was chosen from them and it was impossible to measure the performance of one model in several aspects. Reference [7] further put forward the idea of blending Dempster-Shafer theory with multiple model performance measures to figure out the model weight, in which model performance measure resulted from Leave-One-Out-Validation of set of known points in each step.

The stochastic response surface method is to generate a lot of candidate points in the parameter space through stochastic sampling under certain mechanism and scored those candidate points with the surrogating model and distance measure. The points with highest score will be selected as for next function evaluation point for optimization algorithm[8]. This method is applied to such problems as deep learning hyper-parameter tuning[9] and groundwater modeling[10], in which harvests satisfactory effect. Reference [11] designed and implemented SO-M-s algorithm by adopting the Dempster-Shafer theory-based ensemble surrogate model mechanism in combination with stochastic response surface method in the hope to prove

the stochastic response surface method-based method is no less efficient than the original algorithm in sampling the candidate points.

On the basis of stochastic response surface method, reference [12] selected several candidate points and evaluate those points to carry out synchronous parallel sampling of the parameter space and speed up the solving process of the algorithm. Reference [13] realized a kind of synchronous sampling mechanism based on multi-objective scale and adopted multi-objective optimization method to choose several candidate points on Pareto front for synchronous evaluation by two measures: distance and score. No report about asynchronous parallel sampling strategy based on stochastic response surface method has been made yet. In kriging model-based surrogate model optimization algorithm, reference [14] employed constant liar mechanism by treating the parameter points under evaluation and their current predicted values as real values and adding them into the set of solved points in surrogate model so as to update the model and generate sampling points for next worker. It did achieve asynchronous parallel sampling of the parameter space. This method is also applied to implement the asynchronous parallel sampling mechanisms[5, 15, 16] of those surrogate optimization algorithms like Bayes optimization algorithm and SMAC algorithm and realize the effect of speeding up solving process of the algorithms.

In quantitative trading strategy research field, most of the studies focus on the investigation and implementation of different quantitative trading models, but few of them involve the parameter tuning in specific trading strategies. Reference [17] applied Bayesian optimization algorithm to quantitative trading, optimized such parameters as position-taking time in Markov investment portfolio strategy in order to find the better parameter set that stood out in both risk and revenue.

III. SURROGATE OPTIMIZATION ALGORITHM BASED ON ENSEMBLE MODEL AND STOCHASTIC RESPONSE SURFACE METHOD

In this section, we are going to give an introduction to general steps in ensemble surrogate model and stochastic response surface method:

A. Ensemble model

As depicted in references [7, 11], this paper utilizes multiple surrogate models to establish an ensemble expression of the parameter space and provide the ensemble output of M various models (see the formula below).

$$S_{mix}(x) = \sum_{r\in M} w_r s_r(x),\ \sum_{r\in M} w_r = 1, w_r \geq 0\ \forall r \in M \qquad (1)$$

In the formula, output of M ensemble models means the weighted average of the independent outputs by each of the models, and w_r means the weight of No. r model. Weight w_r assesses the performance of each model in current problem through the set of known parameter points and higher weight is endowed to the model with better performance. In reference [7], all the performance measures of several models were normalized to be expressed as possibility, and Dempster-Shafer theory was used to blend multiple performance measures to determine the final weight. Detailed steps for weight computation are demonstrated in Algorithm 1.

Algorithm 1 Dempster-Shafer theory-based ensemble model weight updating algorithm

Input: M surrogate models $S_i, i \in 1, \dots M$, set of known points $X_{n\times d}$, Y_n;

Output: weight of M surrogate models $w_i, i \in 1, \dots, M$;

1: For n known points (x_i, y_i), apply Leave-one-out-validation method by treating every point as a test set while the rest n-1 sets as the training sets, and training M surrogate models to acquire the predicated values of all those surrogate models about the samples in the test set;

2: Figure out the predicted values of all the models about n known points after being trained with the rest points through Leave-one-out-validation and real values of n points; compute separately three performance measures: the correlation coefficient between the predicted value of each model and the real value $Corr_i$, root-mean-square error $RMSE_i$, and maximum absolute value error MAE_i where $i \in 1, \dots, M$;

3: Compute the normalized scores of all models under three performance measures, in which $m_i^{Corr} = \frac{Corr_i}{\sum_{j\in M} Corr_i}$, $m_{RMSE}^i = \frac{\frac{1}{RMSE_i}}{\sum_{j\in M}\frac{1}{RMSE_i}}$, $m_i^{MAE} = \frac{\frac{1}{MAE_i}}{\sum_{j\in M}\frac{1}{MAE_i}}$

4: Treat the normalized scores of models' three performance measures as the possibility distribution of same event in three experiments, fuse three normalized scores as per Dempster-Shafer theory, and figure out the weight w_i of each model (M models in total);

In accordance with Algorithm 1, when there are n data points, to figure out the weight of each surrogate model, the algorithm should train each model for n times. As the optimization proceeds, the number of known points n keeps growing, so does the computational load of Algorithm 1. If there are many models and model training requires complicated computation, the weight computation will become the primary challenge for implementation of ensemble surrogate model.

To reduce the computational load for updating ensemble model weight with Algorithm 1, reference [11] substituted the Leave-one-out-validation in Algorithm 1 with k-fold cross validation so that the training times required by model became related to K but not to the sample quantity in weight updating and re-acquisition of each model's performance indicators. In order to guarantee the accuracy of calculated weight, K should grow with increase in number of known points.

B. Stochastic response surface method

In those surrogate model-based algorithms such as Bayesian optimization, next point to be evaluated is usually sampled through an auxiliary function established through surrogate model and corresponding uncertain measures in the parameter space. The auxiliary function represents balance between exploitation of unknown parameters space and

exploration of highly uncertain areas on the basis of surrogate model, which searches the optimal solution to the auxiliary function in parameter space through another optimization algorithms based on gradient or other mechanisms and sets it as the next sampling point

By comparison, the stochastic response surface method need not additional optimization steps for the auxiliary function, because it randomly generates a number of candidate points through sampling in the parameter space, utilizes the surrogate model to predict values of those candidate points, and sets the predicted values as exploitation measures. The minimal distance from candidate points to all the points in the set of known points is viewed as the uncertainty measure. Their normalized weighted minimal point are selected as the next sampling point. The steps of stochastic response surface method are listed in Algorithm 2.

Algorithm 2 Stochastic response surface method

1: Conduct sampling in the parameter space and generate N initial sampling points;

2: Utilize the surrogate model to figure out the predicted function value S_i of each candidate point;

3: Compute the distance measure of each candidate point $D_i \leftarrow \min(||x_i - x_{evaled}||)$; /* the distance measure means the minimum Euclidean distance from candidate point to all the points in the set of known ones*/

4: $V_S \leftarrow (S_{max} - S_i)/(S_{max} - S_{min})$;/*compute the relative score of each candidate point's surrogate model predicted function value*/

5: $V_D \leftarrow (D_i - D_{min})/(D_{max} - D_{min})$;/*compute the relative score of each candidate point's distance measure*/

6: $V \leftarrow w * V_s + (1-w) * V_D, w \in [0,1]$; /*compute the weighted average of two scores and set it as the final score of each candidate point; the higher w is, the more inclined it is to choose the better point deemed by the surrogate model; and the lower w is, the more inclined it is to explore the unknown area with higher uncertainty/*

7: Choose the candidate point x^* with highest score as the solving point of next function value, acquire the updated set of known points after $f(x^*)$, and repeat the steps above until the terminal condition is met;

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

There are two common mechanisms behind generation of candidate points: one is to conduct homogenous sampling within the upper and lower parameter bounds while another is to add stochastic disturbance at the place of known optimal point. It is usually believed that the former one can better explore entire unknown space while the latter can better search the neighborhoods of the current optimal point. In reference[18], only a subset of coordinates are added by random perturbation to generate candidate points when the parameter space is of high dimensional. For each coordinate axis, the disturbance possibility is:

$$P = \begin{cases} \max(0.1, 8/d) & if\ d > 8 \\ 1 & otherwise \end{cases} \quad (2)$$

A surrogate optimization algorithm SO-M-c (Surrogate Model Combination Algorithm with Candidate Point strategy)[11] is derived from the combination of Dempster-Shafer theory-based ensemble surrogate model and stochastic response surface algorithm:

Algorithm 3 SO-M-c algorithm

Input: dimension d of the parameter space, upper bound of parameter space highBound and lower bound lowBound, maximum function evaluation number N_{max}, M various surrogate models, and number of candidate points $n_{candidates}$;

Output: optimal parameter x_{best};

1: Utilize symmetric Latin hypercube sampling strategy to sample 2(d+1) initial points in the parameter space, solve the function value of those initial points, and add the set of known points;

2: Apply Algorithm 1 to the set of known points to figure out the weight list of each surrogate model; when number of known points $n \leqslant 50$, adopt Leave-One-Out-Validation; otherwise use K-fold cross validation, $k = 10, 50 < n \leq 100$; $k = 20, 100 < n \leq 150$; $k = 30, 150 < n \leq 200$; ... ...

3: for $i \leftarrow 1$ to $N_{initial}$

4: Randomly sample and generate two groups of candidate points in upper and lower bounds of the parameter space with those points in first group being generated through homogenous sampling and those in second group being equal to the coordinates of current optimal point plus displacement disturbance of $g\delta\Upsilon$ where $\delta = upBound - lowBound$, $g \in \{0.1, 0.01, 0.001\}$ randomly selected in each generation of candidate points. $\Upsilon \sim N(0,1)$ results from stochastic sampling in standard normal distribution and every coordinate of x_{best} is equal to the probability P in Formula (3) plus displacement of $g\delta\Upsilon$

5: Figure out the score of every candidate point as described in Algorithm 2 with score weight w being circulated periodically in the list $[0., 0.1, 0.2, \ldots, 0.9, 1]$, and choose the point with lowest score as the solving point for next function value;

6: Compute and solve the function value of chosen point, add it to the set of known pints, and increase the function evaluation numbers by 1;

7: When function evaluation number reaches N_{max}, terminate the algorithm and return to the current optimal point x_{best}; or go back to Step 2;

Reference [11] indicated Algorithm 3 has equivalent optimization efficiency as the methods adopting genetic algorithm optimization based auxiliary function.

C. *Asynchronous Parallel algorithm based on ensemble model and stochastic response method*

Revealed by Algorithm 3, as for computational load introduced by the adoption of ensemble model, when weights of all the models get updated in each step, as cross validation is used to compute the performance measures of all the models, it is necessary to divide the set of known points and train all the models for multiple times. Even if K-fold cross validation is adopted to cut down the times of training, the times of training will still grow as algorithm proceeds and K rises. It is especially noteworthy that when the models adopted are rather complicated, training and computing expensive and duration remain high, the additional time consumed by the ensemble model will be significantly higher. This becomes a primary factor affecting the performance of the algorithm.

On the other hand, besides the computational load of optimization algorithm itself, as the evaluation of expensive black-box function takes longer time, the time required for the algorithm to obtain the better solution can be significantly reduced when parallel sampling of function space, utilization of computational resources. The parallel sampling mechanism realized in references [12, 13] is a synchronous one in which multiple sampling points are generated in each batch to evaluate the function value before model updating and generation of next batch of sampling points. In a great number of practical problems, adoption of different parameters may cause function evaluation time vary significantly. For instance, in quantitative trading strategy parameter tuning, the difference in parameters incurs varying trading frequencies. In such case, the computational load also varies significantly so that the setting of some parameters may cause the back-test simulation to be terminated in advance. For deep learning parameter tuning task, as the parameter of single-batch sample quantity is set to be different, this can affect both the model performance and time consumed by the training. In similar tasks, if synchronous parallel sampling strategy is adopted, the computational resources will be left in waiting or being idle and the optimization efficiency of the algorithm will be affected.

The parallel algorithm designed in this paper starts to solve two problems. The parallel structure based on Master-Slave architecture is as shown below:

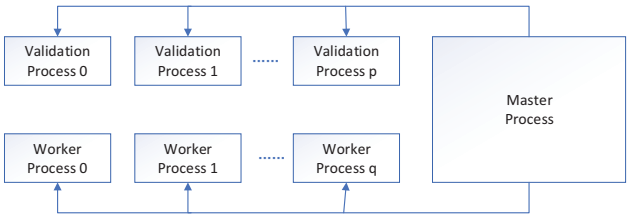

Fig. 1 Master-Slave Structure of Parallel Ensemble Surrogate Optimization Algorithm

Those processes other than the master one are divided into two parts: the Validation process part is used in the parallel cross validation of all the models in computation of ensemble model's weight while the Worker process part in the asynchronous parallel function value evaluation of the parameter space. The master process is responsible to coordinate the whole computational process. More specifically, the computation completed by master process includes:

1 Generate initial sampling points and send them to all the worker processes for evaluating the function value;

2 Receive the function value solving results from all the Worker Processes, preserve and distribute the results to all the Validation Processes, and ensure both Validation Processes and Master Process have preserved a duplicate of the same set of known points;

3 In case of a change in the set of known points, schedule the task of set validation, compute the k-group computational task required by weight updating, distribute the task to all the Validation processed, collect the computing results, and figure out the weight of each ensemble model as per Algorithm 1;

4 When any Worker process becomes idle, generate new to-be-evaluate points of a parameter space according to asynchronous sampling mechanism for the idle Worker process to solve.

Because both Master Process and Validation Process have the same duplicate of the set of known points, when validation task is issued, the Master Process need not communicate with the Validation Process about the data, and it should send only the indexes of the training set and test set corresponding to the validation task of specific Validation Process. This can reduce the communication load therein.

With the parallel framework above, several validation processes can be utilized to speed up the weight updating part of the ensemble model. To further make full use of multiple Worker processes and achieve asynchronous parallel sampling and evaluation of the parameter space, we refer to the common constant liar mechanism in the parallel surrogate optimization algorithms and combines it with the stochastic response surface method to realize following parallel sampling algorithm:

Algorithm 4 Stochastic response surface method-based asynchronous sampling algorithm of parameter space

1: Set the number of evaluated points n=0, utilize symmetric Latin super-cube sampling to generate 2(d+1) initial sampling points, and add all those points into the evaluated task set X_2

2: When any Worker process becomes idle, appeal for problem-solving points:

if n<2(d+1), the n-th initial sampling point will be sent to the idle Worker process for problem solving, $n \leftarrow n + 1$;

else, use steps 4 and 5 in Algorithm 3 to generate next sampling point x. The difference lies in the fact that in the distance measure computation, the distance score of all the candidate points is not the

minimal distance from the set of solved points but that from the set of problem-solving task X_2 and addition of x into this set.

3: When Worker completes the function evaluation task for the parameter point, the result will be sent to the Master Process and added into the set of evaluated points and the surrogate model will be updated accordingly.

The key point in Algorithm 4 is that the distance score computation in stochastic response surface method is no longer based on the set of solved points but the parallel set of both evaluated and under-evaluation points so that new sampling point will not be too closer to the sampling point under evaluating and the algorithm's optimization efficiency won't be reduced. As a matter of fact, Algorithm 4 is the asynchronous version of the algorithm in reference [12], because both of them adopt similar mechanism to take into consideration the sampled but not evaluated parameter point in distance measure computation so as to guarantee a reasonable distribution of different candidate points. Algorithm 4 can also be viewed as of constant liar mechanism under which the point under evaluating process has no effect on the surrogate model at all but can affect the computation of uncertain measure.

To sum up, the parallel structure designed in this section and the asynchronous sampling mechanism designed for stochastic response surface method can utilize the resources and speed up Algorithm 3 from two perspectives, namely ensemble model weight updating and parallel sampling of the parameter space. The parallel framework designed and implemented in this section can be applied to accelerate all sorts of ensemble models and coexist with both stochastic response surface method and other surrogate optimization algorithms that implement asynchronous sampling mechanism on the parameter space.

IV. QUANTITATIVE TRADING STRATEGY TUNING BASED ON SURROGATE OPTIMIZATION ALGORITHM IN FINANCIAL MARKET

In financial market's programming trading strategy, a lot of designed and implemented strategies contain several preset parameters. Whether those parameters are correctly set significantly affects the performance of a strategy in real market. Generally speaking, such parameters can be chosen by the quantitative traders on the basis of their experience. However, in reality, due to such reasons as market fluctuation and strategic complexity, artificially set parameters can hardly reach the expected revenue performance. It is usually necessary to back-test the parameters according to actual historical data and simulate the performance of the strategy against historical data and choose the parameters with best or better performance indicators in the back-test. In this way, the quantitative trading strategy parameter tuning can be viewed as a black-box function optimization problem. The historical data scale maybe large. For example, the data volume required by time-varying high-density data strategies such as deal-by-deal transaction and order entry is far higher than the strategy built on the basis of daily linear data. This paper applies the parallel surrogate optimization algorithm implemented on the basis of ensemble model and stochastic response surface method to the MACD-based commodity futures strategy tuning.

MACD strategy is built on the basis of exponential moving average convergence divergence indicator to generate such decisions as buy-in and sell-out. In actual application, it is necessary to preset first seven parameters: loss limit, profit limit, holding duration, factor computing frequency, short-term ema scale, long-term ema scale, and difference smoothing scale. We express the parameter in vector form $\mathbf{x} = [x_1, x_2, \dots, x_7]^{\mathrm{T}}$, and make hypothesis about the historical trading data about specific commodity futures, certain functional relationship exists between the final Sharpe ratio S and parameters of the strategy:

$$S = f(\mathbf{x}) \qquad (3)$$

Where function value $f(\mathbf{x})$ doesn't have any explicit analytical form, so it has to be determined by execution of historical data simulation strategy. Our goal is to acquire a group of parameters and maximize S or minimize $-f(\mathbf{x})$. Therefore, the strategy parameter tuning problem can be solved through corresponding expensive black-box function optimization algorithm.

V. EXPERIMENT AND RESULTS

A. Algorithm implementation and experimental settings

We use MPI to realize inter-process communication and related parallel mechanism and implement the algorithm designed hereby. For selection of ensemble model, following three models are used in the experiment to form the ensemble model:

Model 1: Gaussian regression model (GPRegression) adopting radial kernel function;

Model 2: radial basis interpolation model (RBFInterpolation, CubicKernel, LinearTail) adopting cube kernel function and linear tail function;

Model 3: radial basis difference model (RBFInterpolation, TPSKernel, LinearTail) adopting splint kernel function and linear tail function.

In subsequent experiment, the effectiveness of cross validation accelerating mechanism in algorithm is tested first. It means when single Worker process is involved, standard test functions including Square, Ackley and Rosenbrock are adopted to acquire the algorithm running time against different numbers of Validation processed in order to prove the effectiveness of the accelerating mechanism in performance improvement and compare it with non-ensemble model in term of optimization effect. The non-ensemble model uses model 2 alone as the surrogate model.

Then, we test the effectiveness of asynchronous sampling mechanism in expensive function scenes. When number of Validation processes is limited, the running time of such standard test functions as Square, Ackley and Rosenbrock is artificially prolonged to simulate the expensive function and test the running time and convergence effect against varying Worker processes.

Finally, the algorithm is tested in MACD quantitative trading strategy parameter tuning in terms of its feasibility and effectively. It is verified through grid search that the optimal Sharpe ratio of the strategy framework used in this paper sharp_best≈1.58.

B. Experimental results

For three standard test functions, namely Ackley, Square and Rosenbrock, the dimension d is set to be 3, 8 and 15, respectively, while the max function evaluation times to be $N_{max} = 400$ to test the results and algorithm running time against single Worker process but varying Validation processes. The experiment is repeated for five times to collect related statistical data. A comparison of the experimental results about the algorithm proposed by this paper with the non-ensemble model serial algorithm is demonstrated in following table:

Table 1 Test results of standard objective functions against varying number of Validation processes

	Ensemble Models (Validation Process=1)			Ensemble Models (Validation Process=2)			Ensemble Models (Validation Process=4)			Ensemble Models (Validation Process=8)			Non-Ensemble Model		
	best	worst	mean	best	worst	mean	best	worst	mean	best	worst	mean	best	worst	mean
Ackly (dim=3)	0.000396813	0.003576105	**0.00203042**	0.001932806	0.003120829	0.002513297	0.001947388	**0.00349858**	0.002594711	**0.00146415**	0.003083021	0.00248983	0.018759	0.048786148	0.033394982
Ackly (dim=8)	0.005393037	**0.00714309**	0.006395396	0.004376771	0.007996055	0.005962623	0.004406529	0.007527639	0.005737363	**0.0045701**	0.007582007	**0.00552047**	0.221735128	0.398005932	0.302410523
Ackly (dim=15)	0.019670077	0.467666229	0.117808599	0.053206185	**0.07483353**	0.043692395	0.01352325	0.636230168	0.207681516	**0.01322798**	0.114061167	**0.03712729**	0.522173335	0.942884683	0.696476617
Rosenbrock (dim=3)	0.151066348	0.302257496	0.230687411	0.05000232	0.345990926	**0.18611214**	**0.04689642**	0.34783851	0.211088639	0.161125449	**0.27333538**	0.227408861	0.260761701	0.420730418	0.381558182
Rosenbrock (dim=8)	5.136543958	**5.72934054**	5.397214124	**4.14799505**	5.892993594	**5.15197516**	4.466308475	5.741720956	5.233662716	5.216129155	5.873131827	5.492545159	7.043555348	8.618122264	7.619810858
Rosenbrock (dim=15)	13.06961022	15.96450457	14.07507253	12.91829005	14.63120345	13.73762255	**11.721997**	**13.6912723**	13.80475336	13.25584204	14.64385401	**13.7306379**	16.80045739	19.91239019	18.48635534
Rastrigin (dim=3)	0.994996633	1.038199024	1.006337611	**0.00104136**	**0.99765112**	**0.79741667**	0.003905949	0.999011894	0.79939181	0.995620376	4.182390879	1.005391469	2.005341856	3.033446518	2.216280264
Rastrigin (dim=8)	8.968866718	14.95618508	11.67685724	**8.00647159**	**13.0390986**	**10.7984018**	8.116163876	15.93614403	12.19782417	8.084440589	17.96670652	14.53549949	25.41018187	33.0752891	28.98879401
Rastrigin (dim=15)	25.14702761	39.8839636	32.35855779	21.94296901	44.35387426	34.98920044	**21.9279217**	45.64963457	**30.6666108**	26.99507341	**42.8747086**	36.16726685	59.62609989	76.95379296	66.86168514

It can be inferred from the table that introduction of multiple Validation processes doesn't affect the optimization effect of the ensemble model-based stochastic response surface algorithm. In addition, a comparison of ensemble model with non-ensemble model in three standard problems reveals ensemble model yields significantly better results than the single model in low-dimension and high-temperature cases. The introduction of ensemble model does improve the adaptability and optimization effect of the algorithm in dealing with different problems.

For setting of standard problems, the running time is the average value from five times. The worker process number is fixed to be 1, and the running time for different validation processes is as shown in following table:

Table 2 Algorithm running time for standard problems against varying number of Validation processes (unit: second)

		validation process=1	validation process=2	validation process=4	validation process=8
Ackley	dim=3	10.62101552	5.57266376	2.81009285	1.36524076
	dim=8	11.55520437	5.94408 02	2.95222421	1.46998612
	dim=15	12.43860449	6.22200 22	3.24231898	1.57922055
Rosenbrock	dim=3	10.76011749	5.62372 15	2.72940539	1.36783767
	dim=8	11.48873052	5.80821 01	2.98639904	1.48714376
	dim=15	11.64408854	6.29077 46	3.21600102	1.49903483
Rastrigin	dim=3	11.31164237	5.68382569	2.75891783	1.41791645
	dim=8	11.83517771	5.94664701	3.01036209	1.49678189
	dim=15	12.11670847	6.09170651	3.22332071	1.52162727

The speed-up ratios of algorithm under different dimensions are demonstrated in following figure:

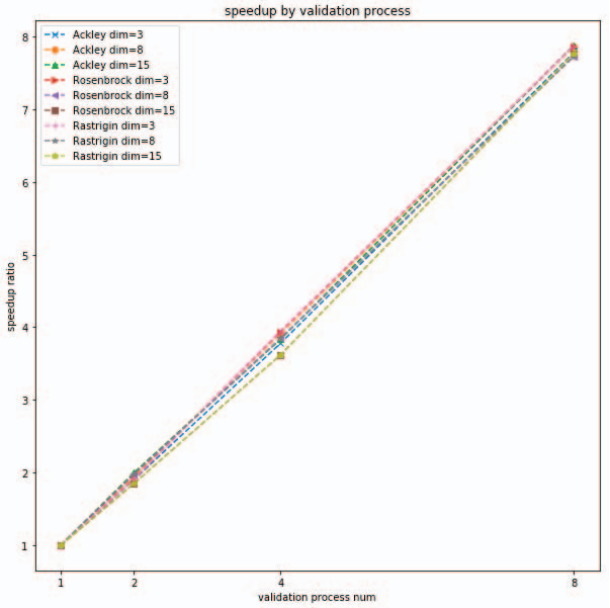

Fig. 2 Speed-up ratios for different test problems against varying number of Validation processes

The function evaluation process for 3 standard test functions is artificially set to last 3 seconds in order to simulate the expensive function evaluation case. When the number of Validation processes is set to be 2, the running time and solving effect of the algorithm against different Worker processes are tested. Statistical data are collected from five repeated experiments. The experimental results are listed in table 3:

Table 3 Test results of standard objective functions against varying number of Worker processes

	Ensemble Models (worker Process=1)			Ensemble Models (worker Process=2)			Ensemble Models (worker Process=4)			Ensemble Models (worker Process=8)		
	best	worst	mean	best	worst	mean	best	worst	mean	best	worst	mean
Ackly (dim=3)	0.00282 1	0.004[5][5] 23	0.00343 79	0.00223 22	0.00439 2[5]	**0.0029[5] [5]**	0.00202 43	**0.0042**	0.00304 1[5]	**0.001[5] [5]**	0.00[5]80 24	0.00340 [5]4
Ackly (dim=8)	0.00498 24	0.00931 32	0.00[5]78 [5][5]	**0.0032[5] 9**	**0.00[5]9[5] 3**	**0.00[5]32 3**	0.004[5]8 94	0.0088[5] 2	0.00[5]43 11	0.00430 42	0.009[5][5] 03	0.00[5]8[5] 8[5]
Ackly (dim=1[5])	0.01449 4[5]	0.1[5]774 28	**0.04714 7**	0.01981 83	0.197[5]8 74	0.0[5]143 32	0.01[5]38 4	0.1[5][5]8 19	0.0[5]801 94	**0.01438 1**	**0.10814 2**	0.04921 93
Rosenbrock (dim=3)	0.02349 83	0.43834 44	0.17370 [5]7	0.0[5]174 33	0.41712 4[5]	0.22[5][5]2 42	**0.01[5]0[5] 3**	0.30[5]94 4[5]	0.20897 83	0.0[5]3[5]9 48	**0.2193[5] 1**	**0.1[5]412 [5]**
Rosenbrock (dim=8)	4.19711 29	[5].8[5]497 [5][5]	[5].03[5]33 17	4.[5]0703 2	[5].01[5][5]4 37	[5].48744 0[5]	**4.0[5]911 [5]**	[5].174[5]2 83	4.934[5]3 48	4.[5][5]4[5][5] 1	**[5].1[5][5]3 [5]**	**4.81924 [5]**
Rosenbrock (dim=1[5])	12.[5][5]40 1[5]	14.7[5]71 8	13.[5][5][5]4 01	**12.0[5][5][5] 4**	14.[5][5]43 [5]1	**13.[5]909 4**	12.8992 13	**14.[5]3[5][5] 2**	13.[5]794 [5][5]	12.3240 [5]2	1[5].4938 4[5]	14.0297 [5][5]
Rastrigin (dim=3)	0.99[5]98 32	1.00018 7	0.99790 34	0.99[5]18 [5]1	1.00483 7[5]	0.99921 81	**0.99[5]01 [5]**	1.0110[5] 99	1.0012[5] 2[5]	0.99[5]29 01	**1.00103**	**0.99729 4**
Rastrigin (dim=8)	**8.00[5][5][5] [5]**	1[5].93[5]9 [5]	**11.77[5]2 1**	8.02[5]8[5] 2[5]	**1[5].9719 9**	12.9[5]98 2[5]	8.977[5]4 78	1[5].928[5] 0[5]	12.[5]909 [5]4	8.9[5][5][5]2 7[5]	1[5].9317 97	12.[5][5]4[5] 8[5]
Rastrigin (dim=1[5])	29.91[5]4 81	[5]0.8729 08	3[5].10[5]9 03	23.9303 [5]1	[5]1.822[5] 47	34.93[5]0 73	23.9178 87	**4[5].1322**	33.9[5]0[5] 02	**23.7147**	[5]0.8294 11	**33.71[5]9**

As revealed by the experimental results, introduction of multiple Worker processes doesn't affect the problem-solving effect of the algorithm at all. Several asynchronous Worker processes can obtain equivalent problem-solving effect as serial sampling.

In case of expensive function evaluation simulation, when number of worker processes is changed, the running time of problem-solving is as shown in following table:

Table 4 Problem-solving time for simulated expensive function algorithm against varying number of Worker processes (unit: second)

		worker process =1	worker process =2	worker process =4	worker process =8
Ackley	dim =3	1226.7 863	651.53 3888	325.79 3981	162.55 4877
	dim =8	1262.8 9921	668.86 1799	332.96 8976	168.07 8262
	dim =15	1295.7 0697	694.32 9754	343.80 119	170.56 0825
Rosenbrock	dim =3	1226.4 5626	656.04 9216	323.46 3287	163.46 9867
	dim =8	1262.6 9157	667.01 6668	335.36 2022	165.85 3553
	dim =15	1302.1 9017	692.87 0646	342.56 3761	170.64 0197
Rastrigin	dim =3	1228.4 9256	657.75 6842	323.99 5184	161.70 3741
	dim =8	1264.0 4268	668.06 6718	332.75 449	168.94 2454
	dim =15	1296.9 7765	696.28 1752	345.41 037	170.28 9751

A comparison of the speed-up ratios introduced by worker process is made in following figure:

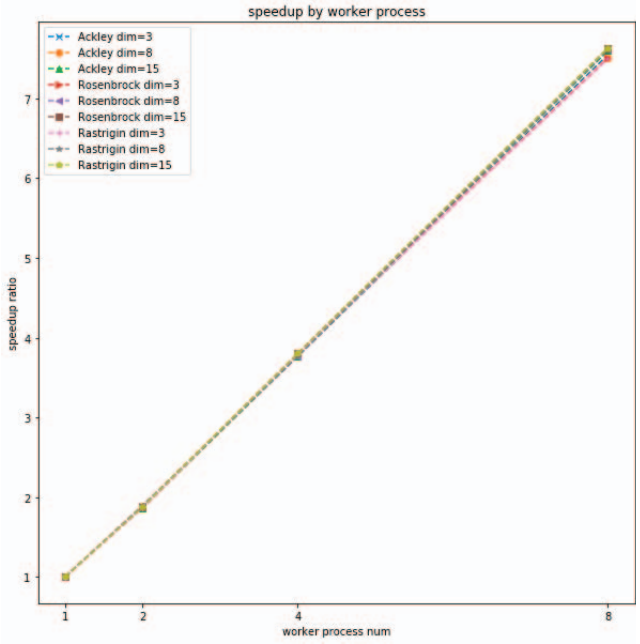

Fig. 3 Speed-up ratios against varying number of Worker processes and different test problems

Finally, the algorithm proposed by this paper is applied to the MACD-based commodity futures strategy with number of Validation processed being fixed to be 4 and function evaluation times $N_{max} = 400$. The resulting optimal Sharpe ratios and algorithm's problem-solving duration against varying number of Worker processes are as shown below (both expressed as average from the statistical data of five experiments):

Table 5 Results and duration of MACD-based strategy parameter tuning solving against varying number of Worker processes

	worker process=1	worker process=2	worker process=4	worker process=8
Optimal Sharpe ratio	1.4015315	1.3980154	**1.414465**	1.4014548
duration of the algorithm (s)	1341.3524	717.30074	357.69397	171.96825

Fig. 4 compares the convergence efficiency against varying number of worker processes.

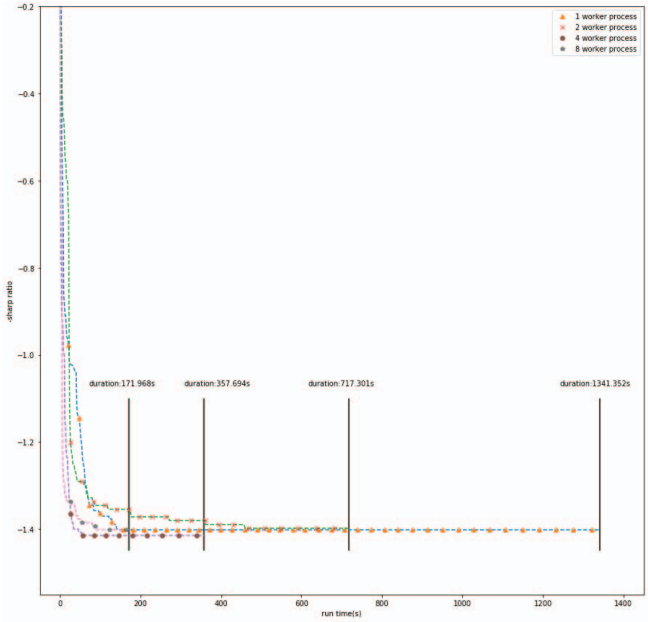

Fig. 4 Convergence efficiency of the algorithm proposed in this paper when applied to MACD strategy tuning problem against varying number of worker processes

When traditional grid search is adopted in MACD strategy parameter tuning, suppose every parameter takes n values, the times of problem-solving will be n^7. It means when n=4, the parameter value should be computed for 16,384 times and historical back-test of equal times should be completed also. By contrast, when the algorithm proposed in this paper is applied, a combination of quasi-optimal parameters can be acquired at extremely low computational cost (about 400 times' function solving). This proves the significantly higher efficiency of the algorithm in optimizing the parameters and its critical significance for solving practical problems.

VI. Conclusion

In view of ensemble surrogate model and performance challenge concerning expensive black-box function optimization, a parallel framework is designed and implemented to speed up the ensemble model-related computation mechanism. The paper further improves and implements a kind of asynchronous parallel sampling mechanism on the basis of stochastic response surface method and proposes for the first time applying the distributed surrogate optimization algorithm to quantitative trading strategy parameter tuning (or the algorithm to the MACD strategy tuning problem). The experiment demonstrates the parallel framework and sampling mechanism proposed by this paper effectively speed up the updating of ensemble model, improve the efficiency of algorithm in solving expensive black-box function optimization, and harvest good application effect in quantitative trading strategy tuning problem. It is expected to further explore the application of the proposed ensemble model framework and algorithm in other various problems in the future study.

REFERENCES

[1] MENGISTU T, GHALY W. Aerodynamic optimization of turbomachinery blades using evolutionary methods and ANN-based surrogate models [J]. Optimization and Engineering, 2008, 9(3): 239-55.

[2] POLOCZEK J, KRAMER O. Local SVM Constraint Surrogate Models for Self-adaptive Evolution Strategies, Berlin, Heidelberg, F, 2013 [C]. Springer Berlin Heidelberg.

[3] PETER T. Using Deep Learning as a surrogate model in Multi-objective Evolutionary Algorithms [J].

[4] KLEIN A, FALKNER S, BARTELS S, et al. Fast Bayesian optimization of machine learning hyperparameters on large datasets [J]. arXiv preprint arXiv:160507079, 2016,

[5] SNOEK J, LAROCHELLE H, ADAMS R P. Practical bayesian optimization of machine learning algorithms; proceedings of the Advances in neural information processing systems, F, 2012 [C].

[6] GOEL T, HAFTKA R T, SHYY W, et al. Ensemble of surrogates [J]. Structural and Multidisciplinary Optimization, 2007, 33(3): 199-216.

[7] MüLLER J, PICHé R. Mixture surrogate models based on Dempster-Shafer theory for global optimization problems [J]. Journal of Global Optimization, 2011, 51(1): 79-104.

[8] REGIS R G, SHOEMAKER C A. A stochastic radial basis function method for the global optimization of expensive functions [J]. INFORMS Journal on Computing, 2007, 19(4): 497-509.

[9] ILIEVSKI I, AKHTAR T, FENG J, et al. Efficient Hyperparameter Optimization for Deep Learning Algorithms Using Deterministic RBF Surrogates, F, 2017 [C].

[10] ASHER M J, CROKE B F, JAKEMAN A J, et al. A review of surrogate models and their application to groundwater modeling [J]. Water Resources Research, 2015, 51(8): 5957-73.

[11] MüLLER J, SHOEMAKER C A. Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems [J]. Journal of Global Optimization, 2014, 60(2): 123-44.

[12] REGIS R G, SHOEMAKER C A. Parallel stochastic global optimization using radial basis functions [J]. INFORMS Journal on Computing, 2009, 21(3): 411-26.

[13] KRITYAKIERNE T, AKHTAR T, SHOEMAKER C A. SOP: parallel surrogate global optimization with Pareto center selection for computationally expensive single objective problems [J]. Journal of Global Optimization, 2016, 66(3): 417-37.

[14] GINSBOURGER D, LE RICHE R, CARRARO L. Kriging is well-suited to parallelize optimization [M]. Computational intelligence in expensive optimization problems. Springer. 2010: 131-62.

[15] GINSBOURGER D, JANUSEVSKIS J, LE RICHE R. Dealing with asynchronicity in parallel Gaussian process based global optimization; proceedings of the 4th International Conference of the ERCIM WG on computing & statistics (ERCIM'11), F, 2011 [C].

[16] HUTTER F, HOOS H H, LEYTON-BROWN K. Parallel algorithm configuration [M]. Learning and Intelligent Optimization. Springer. 2012: 55-70.

[17] BRECQUE C. Bayesian Optimization for Quantitative Trading [M]. 2018.