

Particle Swarm Optimization assisted by Gaussian Processes for Multimodal Function Optimization

Yan Zhang, Yi Zhang

Guangxi Key Laboratory of Geomechanics and Geotechnical Engineering,
Guilin, China
E-mail: yanzi22858@126.com

Abstract-In many real-world optimization problems, the location of multiple optima is often required in a search space. In order to evaluate the solution, thousands of fitness function evaluations are involved that is a time consuming or expensive processes. Therefore, standard Particle Swarm Optimization (PSO) meets a special challenge for a very large number of problem function evaluations. Applying models as a surrogate of the real fitness function is proved effective way to address this challenge. This study proposes a model assisted PSO, which uses a Gaussian Process (GP) approximation model. In this algorithm, the training datasets for establishing a GP are generated by the first two generations particle information. Once the GP model is obtained, the function value is evaluated using trained GP model instead of real function evaluation, so that the total number of function evaluation is clearly reduced. In order to improve the predictive capacity of GP, the training datasets are dynamically renewed through sorting datasets and replacing the worst dataset during iterative process. Numerical results from simulations on several 20 dimensional complex multimodal functions are presented. Furthermore, a comparison of the new algorithm with the standard PSO is also made. Results show that the new algorithm is much more efficient than standard PSO.

Keywords-particle swarm optimization; surrogate; gaussian process; multimodal function optimization

I. INTRODUCTION

Multimodal optimization is a concerned research area in many practical applications such as physics, engineering design and other scientific applications. However, many cases in this domain involve objective functions of multimodal, highly non-linear, with steep and flat regions and irregularities, this could be a very challenging task [1]. Also, for such complex problems, it is impossible to solve using traditional mathematical methods such as quasi-Newton methods and nonlinear conjugate gradient methods because they will get stuck at a stationary point or a local minimum [2]. It is proved that stochastic search methods can solve multimodal optimization problem. Such as genetic algorithms (GA) [3,4], differential evolution (DE)[5,6] particle swarm optimization (PSO) [7,8] and ant colony optimization (ACO) [9,10]. These methods evaluate the objective function in a random sample of points from the search space and subsequently manipulate the sample. Thus, they require thousands of fitness evaluations in order to determine the best sample. In many real world problems,

this computational burden cannot be afforded. Examples of such problems include large scale finite element method analysis or computational fluid dynamics simulations. In such problems, the cost of a single function evaluation is in the order of hours of supercomputer time. This problem limited the applications of stochastic search methods in real world engineering. Hence there is a considerable need to develop efficient method for multimodal optimization.

It has been proved useful to build surrogate models for the computationally expensive analysis and to use it for the purpose of carrying out optimization [11,12]. In recent years, a number of surrogate-assisted methods have been proposed for solving this problem. They include, such as, a hybrid GA method consisting of artificial neural network (ANN) algorithms (GA-ANN)[13], surrogate-assisted evolutionary algorithms (SAEA) [14] and a cellular particle swarm optimization (CPSO) that combines cellular automata (CA) with particle swarm optimization (PSO) [15].However, there are still some dilemmas for these methods. Hereunder are (1) to the surrogate models, the accuracy of regression is low and the adaptability is weak for polynomial response surface. It is always difficulty to find an appropriate network topology and the optimum hyperparameters for ANN; (2) to strategies for optimization, surrogates should be used together with the real fitness function for learning samples. Then the surrogate models such as polynomial response surface or ANN are established by learning samples to take the place of real fitness function. However, this optimization strategy is too dependent on learning samples that are determined before optimization and remain unchanged during the whole optimized process. If the learning samples are not representative or deviation from the optimal value, the prediction accuracy is so bad that leads to the searching accuracy is reduced and easy to fall into local optimum solution.

In this paper, a Gaussian process (GP) surrogate-assisted PSO algorithm is proposed for solving computationally expensive multimodal optimization problems. For convenience, the proposed method will be called PSO-GP. Compared to GA and other stochastic search methods, PSO has characters of simple computation and rapid convergence capability. Compared to ANN, GP has the main advantage is its simplicity: no network size or topology must be chosen. Beside that, GP owns a theoretical framework for obtaining the optimum hyperparameters self-adaptively. Furthermore, in order to solve the (2) dilemma, the method of dynamically update training datasets is used that sorts

datasets and replaces the worst dataset during iterative process.

II. PSO ASSISTED BY GP

A. General

In order to reduce the total number of real function evaluation, an optimization procedure based on PSO using GP as an inexpensive model of the objection function is proposed. PSO is used to randomly generate control variables treated as random solution. Then the fitness of the solution can be obtained by function evaluation. The random solution and corresponding fitness is treated as a sample. After collection of some samples, a GP model is constructed for the given dataset. Once the trained model is gotten, we can directly use the model instead of real function to forecast more adaptive control variables. PSO will generate new generation as new random solution based on the zone. Finally, the current best solution is evaluated on the real function and added to the training dataset instead of the worst particle. If the fitness is unsuccessful, a new loop starts until the required minimal solution is found.

B. Fitness Approximation with GP

In this method, GP is used as approximation of the fitness function, which has the capacity to elegantly handle the uncertainty about the real fitness function, especially for noisy training data. A detailed description for GP is given in references [16]. We provide a brief introduction of GP.

GP is a collection of random variables, any finite set of which has a joint Gaussian distribution and is completely specified by its mean function $m(x)$ and the covariance function $k(x, x')$:

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (1)$$

There is a training set \mathbf{D} of m observations, $\mathbf{D} = \{(\mathbf{x}_i, y_i) | i=1, \dots, m\}$, where \mathbf{x} denotes an input vector, y denotes a scalar output or target. The column vector inputs for all m cases are aggregated in the $D \times m$ design matrix X , and the targets are collected in the vector \mathbf{y} . The goal of Bayesian forecasting is to compute the distribution $p(y_* | \mathbf{x}_*, \mathbf{D})$ of output y^* given a test input \mathbf{x}^* and a set of training points \mathbf{D} . Using Bayesian rule, the posterior distribution for the Gaussian process outputs y^* can be obtained. By conditioning on the observed targets in the training set, the predictive distribution is Gaussian:

$$\mathbf{y}_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y} \sim N(\hat{y}(\mathbf{x}_*), \hat{\sigma}(\mathbf{x}_*)) \quad (2)$$

where the mean and variance are given by:

$$\hat{y}(\mathbf{x}_*) = \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} \quad (3)$$

$$\hat{\sigma}(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_* \quad (4)$$

where a compact form of the notation setting for matrix of the covariance functions are: $k_* = \mathbf{K}(\mathbf{x}, \mathbf{x}_*)$, $\mathbf{K} = \mathbf{K}(\mathbf{X}, \mathbf{X})$, σ_n^2 is the unknown variance of the Gaussian noise.

In this method, the first two generations particles have been evaluated by the real function, the trained GP model can be established. Once the GP model is obtained, the function value is evaluated using trained GP model instead of real function evaluation. It is implemented as the following: firstly, it makes full use of the first two generations particle information to construct the training datasets. Next, GP is used to learn the training datasets and preliminarily establish nonlinear mapping relationship between the decision variables and function values. Finally, the trained GP model is used to evaluate the fitness instead of real function evaluation. Through this measure, the number of real function evaluation for PSO is clearly reduced.

C. Dynamically Update the Optimum Particle

In order to accelerate the velocity of finding the best solution, many generations particle cycled produce, and repeatedly use GP model to evaluate the fitness that has the advantage of evaluating the fitness without calling real function. After cycled production, the particle with the smallest fitness is find that is maybe the optimal solution in the global space to predict, and put the optimal solution to the real function to obtain its true fitness. At the same time, compared to the optimal solution in the history optimal solution, the one of the smaller fitness becomes the optimal particle of the current iteration step, which appointed to the lead particles of next step in order to guide flight direction of the particle swarm before flight.

D. Dynamically Update the Training Datasets

To avoid the algorithm too dependent on initial learning samples and to improve approximation quality of GP model, the measure of dynamically update training datasets is used that can realize by the following two ways:(1) without wasting one time evaluation to the true fitness of the smallest particles of fitness from cycled generation particles involved in part C, the individual information of this particle is served as new knowledge source to update the training datasets instead of the worst particle's information; (2) when it does not satisfy the requirement of accuracy and put into outer cycle, outer particle swarm structure produces a new generation particles under the guidance of the optimal particles that evaluated by the real function. Then all the samples between this generation particles and the original training datasets are arranged from small to large based on the fitness and the particles of smaller fitness are chosen as many as the original training datasets to build new training datasets. Finally, the new training datasets is learned by using GP machine learning method. Maintain training datasets capacity in the same level and enhance the quality

of training datasets can continuously improve the predictive power of GP model and the precision in the iteration process.

E. The Implementary Provisions

The structure of the proposed hybrid method is given in Fig. 1. Its frame mainly consists of two cycles. The cycle depending on “while” is called outer cycle and the cycle depending on “for” is called inner cycle. The maximum number of outer cycle is p. Meanwhile, the maximum number of inner cycle is q. Specific implementation steps are shown as follows:

Step 1: Determine the dimension of PSO according to the number of the decision variables. The particles are evaluated by the real function. If the minimum fitness meets the requirement of accuracy, the algorithm is stopped. The particle of the minimum fitness is the optimal solution. Otherwise, the algorithm goes into the outer cycle and the 1th generation particles of outer cycle are randomly initialize as random solutions in accordance with the particle swarm rules. Then the random solutions are imported real function for fitness evaluation. The particle and its fitness are regarded as the information of itself. The original training datasets is built based on the information of two generations particles.

Step 2: The algorithm goes into the inner cycle. According to PSO rules, it can produce nth generation particles of inner cycle. The nonlinear mapping relationship between the decision variables and function values is

```

Begin
Set parameters of GP and PSO. Randomly generate N particles and evaluate their fitness by the real function;
Find the optimum particle  $p_{id}$  and global optimum particle  $p_{gd}$ ;
 $p = 0$ ;
While (the accuracy is not met)
According to PSO rule, generate N particles and evaluate the real fitness;
Find the optimum particle  $p_{id}$  and global optimum particle  $p_{gd}$ ;
If  $p \neq 0$ 
All the samples are arranged from small to large based on the fitness;
The particles of smaller fitness are chosen as many as the number of  $2*N$ ;
End if
Establish the training datasets based on  $2*N$  particles information;
For  $n=1$  to  $q$ 
Generate N particles according to PSO rule;
Train GP by the training datasets;
Evaluate the fitness of N particles using trained GP;
End For
Find the best particle with the minimum fitness in  $q$  generation;
Evaluate its fitness using real function and replace its fitness evaluated using trained GP;
Compare its fitness to  $gbest$ , select the smaller one as global optimum particle  $p_{gd}$  and update  $gbest$ ;
Replace the worst particle with the maximum fitness in training datasets by this particle and its fitness;
Find the optimum particle  $p_{id}$  and global optimum particle  $p_{gd}$ ;
 $p = p+1$ ;
End While
End

```

Figure 1. PSO-GP algorithmic framework.

In order to implement the method proposed above, a

established by using GP machine learning method through studying the training datasets. And then use learned GP model replacing real function evaluation to evaluate the fitness. Through circulating, the particle of minimum fitness for q times is found, that is chosen as the optimal solution based on GP predictive value.

Step 3: It is considered that there must be certain error between the predictive value of the optimal solution and real solution. So the particle corresponding to the optimal solution is put into real function to obtain the real fitness. Compare the real fitness just obtained with the history optimal solution $gbest$, the smaller one is the optimal solution of current iteration step. Without wasting this real function evaluation, the information of this particle replace the worst particle's information in the training datasets to update the training datasets.

Step 4: Judge the precision of the current optimal solution. If the precision satisfies the requirement of accuracy, the algorithm is stopped. Otherwise, the 1+pth generation particles of outer cycle are produced and evaluated by the real function. Then all the samples between this generation particles and the original training datasets are arranged from small to large based on the fitness and the particles of smaller fitness are chosen as many as the original training datasets to build new training datasets. Finally, the algorithm goes to step 2.

the proposed method are discussed next with some benchmark functions

III. EXPERIMENTAL RESULTS AND DISCUSSION

To analyze the performance of the algorithms simulations were performed for 3 well known 20 dimension multimodal test functions. For each case the proposed algorithm was compared to the standard PSO algorithm by evaluating convergence velocity and efficiency. In order to avoid unnecessary time costs, the interval is confined to [-2, 2]²⁰. The task is to solve the minimum value of these benchmark functions. All considered functions have a minimal function value of 0. The known global minimum is located in the origin.

The common parameters for PSO are population size(N), lower (lb) and upper (ub) bounds of the search space, maximum number of iterations (Maxit) that is the total number of cycle for PSO-GP, inertia parameter (w), maximum velocity of particle (vmax), and acceleration constants(c1 and c2).The configuration settings used for performing the experiments are described as follows: N =30, lb=-2, ub=2, vmax=1, c1= c2=2, Maxit=8000 for Ackley and Griewank functions while Maxit=20000 for Rastrigin function. The w is decreased as proposed by (5). The maximum number of inner cycle for PSO-GP is q=10.

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{t_{\max}} t \quad (5)$$

where t is the current iteration step, tmax is the maximum iteration step, wmax is the maximum inertia weight, wmin is the minimum inertia weight. Usually, wmax=0.9, wmin=0.4.

For the comparisons to be fair, all algorithms are forced to use the same precision for the same function and the results of all experiments are averaged over 30 independent runs. There are different precision for different functions. That is to say, the error between optimized results and real minimal value is different. The precision for Ackley and Griewank functions are e=1e-3 and for Rastrigin function is e=30, that means the run is stopped and regarded as successful, when the function value is smaller than e.

A. Ackley Function

Ackley function, originally proposed by Ackley [17], has an exponential term that covers its surface with numerous local minima. Its number of local minima increases exponentially with the problem dimension. Ackley's function has a global optimum with very strong local features. The graph of the function in the two-dimensional case is shown in Fig. 2. It is defined as follows:

$$f_1(x) = 20 + e - 20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) \quad (6)$$

$$x = [0, 0]^{20}, f_{\min} = 0$$

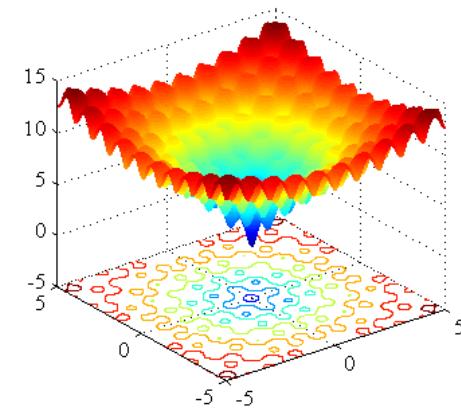


Figure 2. The graph of the 2-dimensional Ackley function.

The total number of the real function evaluation using PSO-GP and PSO are 14817 and 112470 respectively. Our algorithm has a significant advantage on cost compared with the original algorithm, which is nearly 8 times lower than PSO. In order to give a visualized and detailed comparison of the velocity of convergence, the graph compared the performances of PSO-GP and PSO versus iteration is shown in Fig. 3.

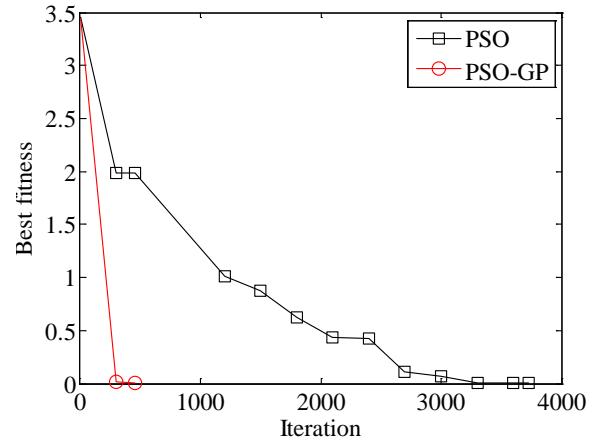


Figure 3. Convergence trends for 20-dimensional Ackley function compared with PSO and PSO-GP.

B. Griewank Function

Griewank function, first introduced by Griewank [18], has been employed as a test function for global optimization algorithms in many papers. The function has a very large number of local minima, exponentially increasing with dimension [19]. Fig. 4 shows the graph of the function in the 2-dimensional case. It is defined as follows:

$$f_2(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (7)$$

$$x = [0, 0]^{20}, f_{\min} = 0$$

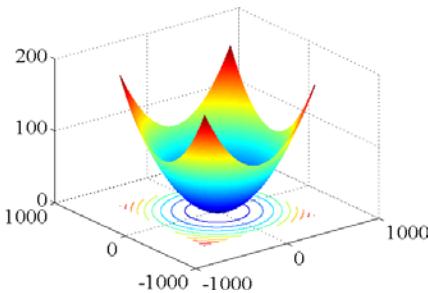


Figure 4. The graph of the 2-dimensional Griewank function.

Both PSO and PSO-GP can find the minimum value at the above settings. The total number of fitness evaluation required for PSO is 73620 while for PSO-GP is 4711, that is nearly 16 times. The averaged convergence trends obtained using PSO-GP and PSO are shown in Fig. 5.

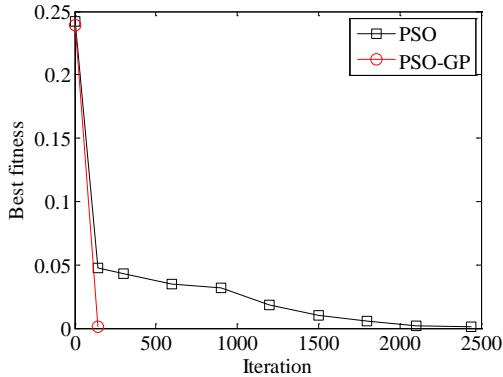


Figure 5. Convergence trends for 20-dimensional Griewank function compared with PSO and PSO-GP.

C. Rastrigin Function

Rastrigin function is a typical example of non-linear multimodal function. It was first proposed by Rastrigin [20] as a 2-dimensional function and has been generalized by Mühlenbein [21]. Rastrigin function is an important highly multimodal function which is used in global optimization because it has the property to have large search space and many local minima but only one global minimum. The graph of the function in the 2-dimensional case is shown in Fig. 6 and it is defined as follows:

$$f_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (8)$$

$$x = [0, 0]^3, f_{\min} = 0$$

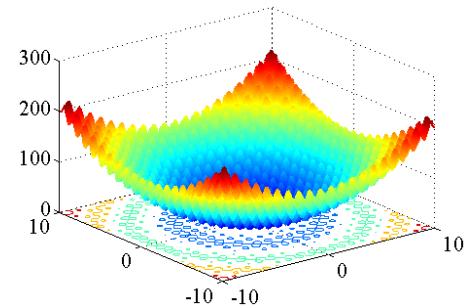


Figure 6. The graph of the 2-dimensional Rastrigin function.

At the mentioned settings, PSO and PSO-GP both can find the minimum value. But there is a significant difference in their total number of fitness evaluation. The total number of fitness evaluation for PSO is 114780, while, the total number of fitness evaluation for PSO-GP is merely 867. The total number of real function evaluation required for GP-PSO is remarkably less than that required for PSO, that is more than 132 times. Fig. 7 shows performances of PSO-GP and PSO for averaged convergence trends.

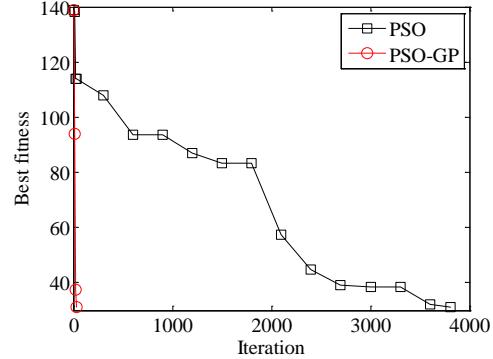


Figure 7. Convergence trends for 20-dimensional Rastrigin function compared with PSO and PSO-GP.

Fig. 3, Fig. 5 and Fig. 7 show that how PSO-GP converged towards the optimal solution faster than PSO algorithm. This indicates that the proposed PSO-GP is much more efficient than PSO to achieve reasonable accuracy. In addition, both PSO-GP and PSO can achieve expected accuracy after the program runs over 30 times for all experiments. This demonstrates that the robustness of PSO-GP is as same as PSO. It owes to the basic evolution rule of particles in PSO-GP that does not change comparing to PSO, just the search speed of PSO is accelerated through searching the promising new solution using GP approximation. Thus, the capability of searching the global optimum is same between PSO-GP and PSO, but the efficiency of PSO-GP is much higher than PSO.

IV. CONCLUSIONS

In this paper an optimization algorithm based on PSO and GP is proposed for optimizing multimodal functions. The algorithm combines the PSO with GP, PSO is treated as an optimization technique to build the main structure and GP is used to predict the fitness instead of some real evaluation. The algorithm involves approximation of the fitness

evaluation by the trained GP model and the dynamically update training datasets that introduces new approach for reducing the real fitness evaluation.

The new algorithm is tested on 3 benchmark functions. The dimensions of these problems are 20. The experimental results have clearly demonstrated that the new algorithm is more efficient than PSO. However, the experiment is just a mathematical test that can not obviously reflect the superiority of PSO-GP. If there is a complicated optimization problem in realistic applications which evaluate the fitness one time consuming many hours, PSO-GP will play an important role in cutting computational cost. Future research therefore will put PSO-GP into the complicated realistic applications.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (51409051) and the Natural Science Foundation of Guangxi (2014GXNSFBA118256). The authors would like to thank the Colleges of Guangxi scientific research project (YB2014156) and Scientific Research Allowance of Guilin University of Technology (002401003413).

REFERENCES

- [1] A. Georgieva and I. Jordanov, "Global optimization based on novel heuristics, low-discrepancy sequences and genetic algorithms," *Eur. J. Oper. Res.* Amsterdam, vol. 196, pp. 413–422, July 2009.
- [2] C. T. Luo, S. L. Zhang, C. Wang, and Z. L. Jiang, "A metamodel-assisted evolutionary algorithm for expensive optimization," *J. Comput. Appl. Math. Netherlands*, vol. 236, pp. 759–764, October 2011.
- [3] D. Bunnag and M. Sun, "Genetic algorithm for constrained global optimization in continuous variables," *Appl. Math. Comput. America*, vol. 171, pp. 604–636, December 2005.
- [4] L. N. Xing, Y. W. Chen, and H. P. Cai, "An intelligent genetic algorithm designed for global optimization of multi-minima functions," *Appl. Math. Comput. America*, vol. 178, pp. 355–371, July 2006.
- [5] A. W. Mohamed, H. Z. Sabry, and M. Khorshid, "An alternative differential evolution algorithm for global optimization," *J. Adv. Res. Cairo*, vol. 3, pp. 149–165, April 2012.
- [6] S. Kitayama, M. Arakawa, and K. Yamazaki, "Differential evolution as the global optimization technique and its application to structural optimization," *Appl. Soft. Comput. Netherlands*, vol. 11, pp. 3792–3803, June 2011.
- [7] M. A. Behrang, E. Assareh, A. R. Noghrehabadi, and A. Ghanbarzadeh, "New sunshine-based models for predicting global solar radiation using PSO (particle swarm optimization) technique," *Energy England*, vol. 36, pp. 3036–3049, May 2011.
- [8] M. M. Noel, "A new gradient based particle swarm optimization algorithm for accurate computation of global minimum," *Appl. Soft. Comput. Netherlands*, vol. 12, pp. 353–359, January 2012.
- [9] M. D. Toksari, "Minimizing the multimodal functions with Ant Colony Optimization approach," *Expert Syst. Appl. America*, vol. 36, pp. 6030–6035, April 2009.
- [10] C. W. Leung, T. N. Wong, K. L. Mak, and R. Y. K. Fung, "Integrated process planning and scheduling by an agent-based ant colony optimization," *Comput. Ind. Eng. England*, vol. 59, pp. 166–180, August 2010.
- [11] Y. Jin, M. Olhofer, and B. Sendho, "A framework for evolutionary optimization with approximate fitness functions," *IEEE T. Evolut. Comput. America*, vol. 6, pp. 481–494, December 2002.
- [12] Y. S. Ong, P. B. Nair, and A. J. Keane, "Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling," *AIAA J. America*, vol. 41, pp. 687–696, April 2003.
- [13] J. Cheng, "Hybrid genetic algorithms for structural reliability analysis," *Comput. Struct. England*, vol. 85, pp. 1524–1533, October 2007.
- [14] Y. C. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *IEEE T. Evolut. Comput. America*, vol. 4, pp. 62–76, August 2009.
- [15] Y. Shi, H. C. Liu, L. Gao, and G. H. Zhang, "Cellular particle swarm optimization," *Inf. Sci. America*, vol. 181, pp. 4460–4493, October 2011.
- [16] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, Massachusetts, 2006.
- [17] D. Ackley, *Genetic Algorithms and Simulated Annealing*. Morgan Kaufman Publishers, Los Altos, 1987.
- [18] A. O. Griewank, "Generalized descent for global optimization," *J. Optimiz. Theory App. America*, vol. 34, pp. 11–39, May 1981.
- [19] M. Locatelli, "A Note on the Griewank Test Function," *J. Global Optim. Netherlands*, vol. 25, pp. 169–174, February 2003.
- [20] A. Törn and A. Zilinskas, *Lecture Notes in Computer Science*. Morgan Springer-Verlag, Berlin, 1989.
- [21] H. Mühlenbein, D. Schomisch, and J. Born, "The parallel genetic algorithm as function optimizer," *Parallel Comput. Netherlands*, vol. 17, pp. 619–632, September 1991.