

An Evolutionary Algorithm using GP surrogate model for expensive constrained optimization problems

Meiyi Li, Hai Zhang, Rong Lv

The College of Information Engineering*

Xiang Tan University

Xiang Tan, China

meiy_li@yahoo.com.cn, focus_zh@yeah.net, lvrong0409@yahoo.cn

Abstract: In expensive constrained optimization problems, the evaluation of candidate solutions could be extremely computationally and/or financially expensive. This paper proposes a method, called DyHF-GP, for reducing computation costs and raising optimization efficiency, by combining Gaussian stochastic process model(GP) with DyHF(Dynamic Hybrid Framework). In DyHF-GP, the Latin Hypercube Sampling(LHS) is used to sample points, then the true function is surrogated by GP. In evolutionary processes, the sample points and the GP are updated by retention and replacement mechanism. The using of GP and true function is controlled by error among several neighbor generations. The 13 standard test functions show that DyHF-GP has higher accuracy and retrieval efficiency. The number of FES is reduced by about 60% on average within 10^{-4} error, which diminishing the computation costs of the objective functions greatly.

Key words: Gaussian stochastic process model; Expensive constrained optimization; Surrogate model; Fitness evaluation; DyHF

I. INTRODUCTION

Evolutionary algorithms, because of their implicit parallelism, strong robustness and other characteristics, are widely used to solve single/multiple objective optimization problems. However, in the course of evolution, EAs need a great number of fitness evaluations and comparison for candidate solutions of objective functions in order to achieve acceptable solutions. But for complex optimization problems, one objective function evaluation would need to consume a large amount of time or computation costs (Expensive Problems). Therefore, using EAs based on the original functions to search for solutions may be costly.

For expensive problems, researchers have emerged in a lot of outstanding methods. Strategies or approaches to deal with expensive fitness functions can be distinguished into three main types [1]: Problem Approximation, Functional Approximation and Evolutionary Approximation. Their main motivation is to reduce the number of evaluations directly to the original objective function. For example, Response Surface Method (RSM) [2], which belongs to Functional Approximation, comprises three main components: regression surface fitting, design of experiments and optimizations using the approximated responses. Fitness

Inheritance is a technique of which the mechanism works as follows: when assigning the fitness to an individual, some times we evaluate the objective function as usual, but the rest of the time, we assign fitness as an average of the fitness of the parents. In Problem Approximation, to save the cost of the experiments, numerical simulations instead of physical experiments are used to pseudo-evaluate the performance of a design.

In many science and engineering disciplines, it is not uncommon to face constrained optimization problems (COPs). Traditional methods to solve this kind of problems are usually local searching methods based on gradient, such as Gradient projection method, penalty functions, Lagrangian, quadratic programming method and so on. But these methods depend on the initial points and the function forms excessively. They are powerless for problems whose feasible region are not connected, who are non differentiable or don't have explicit mathematical expressions. Compared with traditional methods, EAs don't need too much prior knowledge of the problems, which are more conducive to deal with COPs. Camponogara and Talukdar transform the COP into a 2-objective optimization problem using Pareto strategy. Its feature is calculating the improvement direction from the Pareto assemblage. The CW [6], proposed by Cai and Wang, is robust and efficient when handling linear/nonlinear equality/inequality constraints. Ray and Liew [7] proposed a society and civilization model. In this model, a society contains a cluster of individuals in the search space, and a civilization is a set of such societies. During the evolution, intrasociety information exchange and intrasociety information exchange are used. These COEAs have certain effects for general COPs, but they don't involve expensive COPs.

Although EAs have been widely used in expensive optimization problems and non-expensive COPs, domestic researches and applications in the field of expensive COPs, which are widespread in engineering practices, have not been reported. With motivation of solving expensive COPs, an algorithm, named DyHF-GP, is proposed based on GP model, which is used as an approximate model to predict the objective functions. DyHF-GP uses DyHF as the search mechanism and can reduce the FES. The performance of DyHF-GP is tested on 13 benchmark test functions from CEC2006. The experimental results show that the method has certain feasibility and validity.

This work was supported in part by National Natural Science Foundation of China, under Grant No. 61070232.

* The college now has the provincial key subject of Computer Software and Theory.

II. PROBLEM DESCRIPTION AND SURROGATE MODEL

A. Constrained Optimization Problem

Without loss of generality, the constrained optimization problems(COPs) can be formulated as follows:

$$\begin{aligned} & \text{minimize} && f(\bar{x}) \quad \bar{x} = (x_1, x_2, \dots, x_n) \in \mathfrak{R}^n \\ & \text{subject to} && g_j(\bar{x}) \leq 0, j = 1, \dots, l \\ & && h_j(\bar{x}) = 0, j = l + 1, \dots, m \end{aligned} \quad (1)$$

where $\bar{x} \in \Omega \subseteq S$, Ω is the feasible region, and S (decision space) is an n-dimensional rectangle space in \mathfrak{R}^n defined by the parametric constraints $L_i \leq x_i \leq U_i, 1 \leq i \leq n$.

In general, the degree of constraint violation of a vector \bar{x} on the jth constraint is defined as

$$G_j(\bar{x}) = \begin{cases} \max\{0, g_j(\bar{x})\}, & 1 \leq j \leq l \\ \max\{0, |h_j(\bar{x}) - \delta|\}, & l + 1 \leq j \leq m \end{cases} \quad (2)$$

where δ is a positive tolerance value for equality constraints. Then, $G(\bar{x}) = \sum_{j=1}^m G_j(\bar{x})$ reflects the degree of constraint violation of the vector \bar{x} , and represents the distance of the individual \bar{x} from the boundaries of the feasible set.

B. Surrogate Model

Surrogates or meta-models[1] is a method of functional approximation. A new expression is constructed for the objective function based on previous data obtained from the real objective functions. RSM, Gaussian stochastic process(GP), Radial Basis Functions, Neural Networks, Support Vector Machines and Artificial Immune Systems are usually used surrogates.

In these methods, the GP[8] has special characteristics. The most important characteristic is that it can provide an error limit for each prediction. The capacity to overcome excessive fitting, a limited number of meaningful and adjustable parameters, and real time data addition without needing to optimize the model parameters, are the reasons why to choose GP.

GP model can be described as follows:

(1) Assumptions: To build a cheap surrogate model for an expensive function $y = g(x), x \in R^n$, Gaussian stochastic process modeling makes the following assumptions[9]:

- a) For any x , the prior distribution of $g(x)$ is Gaussian with constant mean μ and constant variance σ^2 .
- b) For any $x, x' \in R^n, c(x, x')$, the correlation between $\mathcal{E}(x)$ and $\mathcal{E}(x')$, depends only on $x - x'$, more precisely

$$c(x, x') = \exp[-d(x, x')], d(x, x') = \sum_{i=1}^n \theta_i |x_i - x'_i|^{p_i}, \theta_i > 0, 1 \leq p_i \leq 2 \quad (3)$$

(2) Hyperparameter Estimation: Giver K points $x^1, \dots, x^K \in R^n$ and their g-function values y^1, \dots, y^K , the

hyperparameters $\mu, \sigma, \theta_1, \dots, \theta_n, p_1, \dots, p_n$ can be estimated by maximizing the likelihood

$$\frac{1}{(2\pi\sigma^2)^{K/2} \sqrt{\det(C)}} \exp\left[-\frac{(y - \mu\mathbf{1})^T C^{-1} (y - \mu\mathbf{1})}{2\sigma^2}\right] \quad (4)$$

where C is a $K \times K$ matrix, $C_{i,j} = c(x^i, x^j)$, $y = (y^1, \dots, y^K)^T$ and $\mathbf{1}$ is a K -dimensional column vector of ones.

To maximize (4), the values of μ and σ must be

$$\hat{\mu} = \frac{\mathbf{1}^T C^{-1} y}{\mathbf{1}^T C^{-1} \mathbf{1}} \quad (5)$$

$$\hat{\sigma}^2 = \frac{(y - \mathbf{1}\hat{\mu})^T C^{-1} (y - \mathbf{1}\hat{\mu})}{K} \quad (6)$$

Substituting (5) and (6) into (4) eliminates the unknown parameters μ and σ from (4). As a result, the likelihood function depends only on θ_i and p_i . An optimization method(like DE) can then be used for maximizing (4) to obtain estimates $\hat{\theta}_i$ and \hat{p}_i . Then estimates $\hat{\mu}$ and $\hat{\sigma}^2$ can be readily obtained from (5) and (6).

(3)The Best Linear Unbiased Prediction and Predictive Distribution[10]: Given hyperparameter estimates $\hat{\theta}_i, \hat{p}_i, \hat{\mu}$ and $\hat{\sigma}^2$, one can predict $g(x)$ at any untested point x based on the g-function values y^i at x^i . The best linear unbiased predictor of $g(x)$ is

$$\hat{y}(x) = \hat{\mu} + r^T C^{-1} (y - \mathbf{1}\hat{\mu}) \quad (7)$$

and its mean squared error is

$$s^2(x) = \hat{\sigma}^2 \left[1 - r^T C^{-1} r + \frac{(1 - \mathbf{1}^T C^{-1} r)^2}{\mathbf{1}^T C^{-1} \mathbf{1}}\right] \quad (8)$$

where $r = (c(x, x^1), \dots, c(x, x^K))^T \cdot N(\hat{y}(x), s^2(x))$ can be regarded as a predictive distribution for $g(x)$.

III. GP MODEL FOR SOLVING EXPENSIVE COP

A. Constrained Surrogate Model

The main characteristic of using multiobjective constraint-handling technique to treat constraints is redefining COPs as MOPs. Generally, all constraints are treated as one objective. The original problem is redefined in such a manner that two objectives are considered: the first is to optimize the original objective function, and the second is to minimize the degree of constraint violation. In COPs, number of constraints are usually large, which may easily lead to the solutions outside the feasible region(especially in the boundary of the feasible region), when using GP model to predict all objective function and constraints, because of

the error of GP. Therefore, in order to guarantee the effectiveness of the algorithm, DyHF-GP uses GP only for the objective function and uses the original expression for the constraint conditions.

B. Combining GP model with DyHF

DyHF[11] using global search model and local search model for solving COPs dynamically according to the feasibility proportion of the current population. The method of combining DyHF with GP model, which may reduce the FES, is using GP instead of the original function to predict a part of the fitness of individuals during the evolution. Framework schematic diagram of the combination of DyHF and GP is shown in Figure 1.

The accuracy of the surrogate model for approximating fitness functions relies on the sampling technology and the sample updating technology. DyHF-GP uses Latin hypercube sampling(LHS) [12], which can ensure that all portions of the vector space is represented. In order to use the real values of original function effectively during the evolution, a dynamic sample capacity of points is set. In the course of evolution, the sample-set is needed to be update dynamically by retention and replacement mechanism: Find out points in population P whose G-values are 0, if the number of sample point is less than upper limit, then add these points into sample-set, otherwise, replace some points who have the maximum G-value by these points. By this mechanism, with the process of evolution, sample points will become more concentrated near the feasible region. GP model would predict the values more accurate and converge faster in a small region.

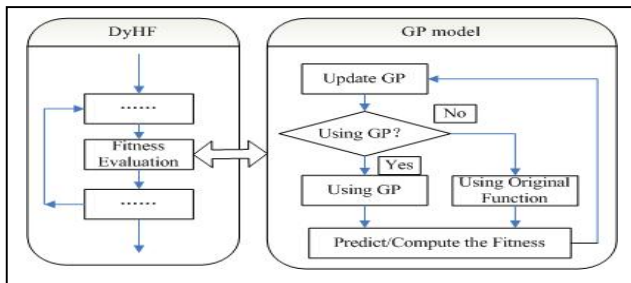


Figure 1. Framework of the combination of DyHF and GP

After updating the sample-set, GP model should be updated based on the new sample-set in order to improve the prediction precision. If the frequency of updating is too low, the prediction model will run out large errors and converge slowly; on the contrary, the updating may consume a large amount of computational costs. DyHF-GP uses a method to update GP model dynamically. The number of new

replacement sample points is as the basis of judgment. Updating the model when the number reaches the upper limit set in advance, while the upper limit changes with the evolution and relate to the sample capacity, feasible rate and function dimension. In the initial stage of evolution, the feasible rate is low, so the frequency of updating should be increased in order to make individuals evolve to feasible region. With the increase of feasible solutions, frequent updating the model has little effect on predicting, instead it may costs much more. In addition, the frequency should be lower when the sample capacity is larger and the dimension of function is lower.

When evaluating the fitness, too much usage of GP model may cause harder to obtain the exact optimal ever can not converge because of the superposition of errors in the model. While too much usage of original objective functions will fail to effectively reduce the number of FES. In order to maintain a balance, DyHF-GP counts the errors between objective functions and Optimal values are less than a threshold in continuous generations. If all the errors are less than a threshold(like 10^{-8}), then the original objective function is used to evaluate the fitness, otherwise, the GP model is used.

C. Algorithm Flowchart and Steps

The flowchart of DyHF-GP is shown in Figure 2.

DyHF-GP adopts the following steps for solving expensive COPs:

Step 1: Initializing. Sample points as the sample-set from search space S using LHS and generate an initial population P by choosing NP points from sample points, evaluate the f-value and the G-value for each point, and compute the number of feasible solutions(NF) in P, Establish the GP model. Set $count_diff=0$, which means the errors between objective functions and Optimal values are less than a threshold in continuous $count_diff$ generations. Set $count_max$, which means the maximum of $count_diff$. Set the number of updating(increasing or replaced) sample points $sample_len=0$.

Step 2: Choosing the search model. If $rand < (NP - NF) / NP$, where $rand$ is a uniformly distributed random number between 0 and 1, then the local search is applied, otherwise, the global search is applied.

Step 3: Local/Global search model: if $count_diff > count_max$, then using the original objective function to compute the fitness of the new individuals generated during the search and updating the new individuals, otherwise, using GP model to predict the fitness. Update $sample_len$.

Step 4: Compute the number of feasible solutions (NF) in P. Replace individuals in sample-set who have the

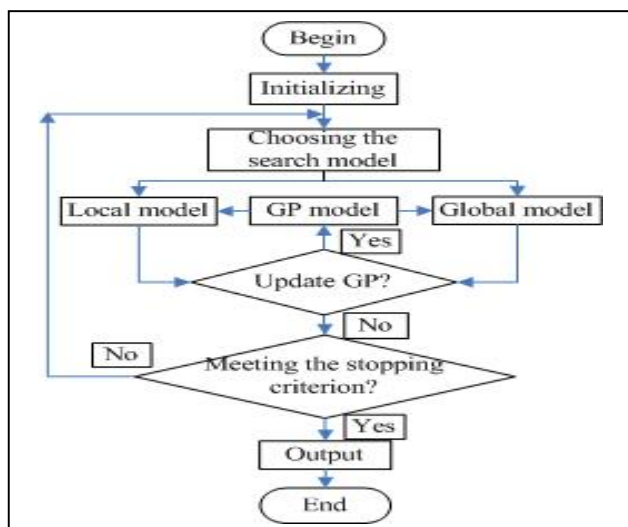


Figure 2. The flowchart of DyHF-GP

maximum G-value by individuals in P whose G-value equal to 0, or add these individuals in P into the sample-set. Update *sample_len* and *count_diff*.

Step 5: If $sample_len \geq (m/(n*n))*(NF/NP)$, then update the GP model.

Step 6: If the stopping criterion is met, stop and output the best solution in P, else go to Step 2.

IV. EXPERIMENTAL ANALYSIS

The performance of DyHF is assessed on 13 benchmark test functions. For each test function, 25 independent runs are performed. The actual parameter settings are $m=400$, and $extra_m=100$.

All data tables are shown on the last page of this document.

Table 1 shows the number of FES to achieve the success condition: $f(\bar{x}) - f(\bar{x}^*) \leq 0.0001$ and the feasible rate.

As shown in Table I, feasible rate in the population is not reached 100% when the DyHF-GP achieves the success condition, but the FES is reduced 66.4%-98% compared with DyHF. Hence one can see that DyHF-GP can reduce the FES significantly for expensive COPs and save a lot of computational overhead.

Table II-IV shows the Function Error values between objective functions and Optimal values and Feasible Rate when $\max FES=1*10^3$ and $FES = 5*10^3$.

From Table II-IV, compared with DyHF, DyHF-GP can obtain a higher accuracy for most test functions under the condition of the same or less FES, which means that the DyHF-GP can be accurate for solving expensive COPs.

From the experiments results, we can know that expensive COPs are very complex. For g09,g13,etc., there decision spaces are small. Although the feasible rates are outstanding, they are difficult to converge to the exact optimal solutions; To the contrary, for g05,g17,etc., there decision space are two small. A small number of sample

points could not obtain an accurate prediction model, which may cause large errors.

V. CONCLUSION

For expensive COPs, computing the objective functions and the degree of constraint violation requires high computational costs. How to make the objective and constraint violation function assessment become efficient is a significant research topic. This paper proposes the DyHF-GP, which can effectively reduce the number of FES, utilizing GP model to predict/approximate the original objective functions and update the model dynamically. A comprehensive inspection of the algorithm is shown by some test functions. Expensive COPs in the real-world are very complex. Thus, constructing a prediction model of less error and improving the accuracy of the algorithm are still in need of further research.

REFERENCES

- [1] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Comput. Fusion Found. Method. Appl.*, vol. 9, no. 1, pp. 3-12, Jan. 2005.
- [2] ROMMEL G. REGIS, and CHRISTINE A. SHOEMAKER, "Constrained Global Optimization of Expensive Black Box Functions Using Radial Basis Functions," *Journal of Global Optimization* 31: 153-171.
- [3] Robert E. Smith, B. A. Dike, and S. A. Stegmann, "Fitness inheritance in genetic algorithms," *SAC 1995: Proceedings of the 1995 ACM symposium on Applied computing*, pp. 345-350. ACM Press, New York (1995)
- [4] Y. Wang, Z.X. Cai, Y.R. Zhou, and C.X. Xiao, "Constrained optimization evolutionary algorithms," *Journal of Software*, Vol.20, No.1, January 2009, pp.11-29.
- [5] Camponogara E., and Talukdar S. N., "A Genetic Algorithm for Constrained and Multiobjective Optimization," in *3rd Nordic Workshop on Genetic Algorithms and Their Applications*. Jarmo T A, editor, Vaasa, Finland, August 1997: 49-62.
- [6] Zixing Cai, and Yong Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 658-675, 2006.
- [7] Ray T., and Liew K. M., "Society and civilization: an optimization algorithm based on the simulation of social behavior," *IEEE Trans. Evol. Comput.*, 7(4), 2003: 386-396.
- [8] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive Multiobjective Optimization by MOEA/D with Gaussian Process Model," *IEEE Trans on Evolutionary Computation*, vol. 14, no.3, pp 456-474, 2010.
- [9] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments(with discussion)," *Statist. Sci.*, vol. 4, no. 4, pp. 409-423, Nov. 1989.
- [10] Carl Edward Rasmussen, and Christopher K. I. Williams, "Gaussian Processes for Machine Learning," *The MIT Press*, 2006. ISBN 0-262-18253-X..
- [11] Yong Wang, and Zixing Cai, "A dynamic hybrid framework for constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 1, pp. 203-217, 2012.
- [12] M.D. McKay, W.J. Conover, and R.J. Beckman, "A comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, vol. 21, no. 2, 1979.

TABLE I. NUMBER OF FES TO ACHIEVE THE SUCCESS CONDITION, FEASIBLE RATE

Algorithm	Prob.	Best	Mean	Feas.Rate	Prob.	Best	Mean	Feas.Rate
DyHF-GP	g04	2360	2997	0.904	g15	49	121	0.273
DyHF		34300	40235	1		15540	23038	1
DyHF-GP	g05	1418	1537	0.929	g16	1897	2386	0.951
DyHF		41580	47236	1		25620	30300	1
DyHF-GP	g06	1800	2090	0.776	g18	1728	1853	0.964
DyHF		27580	37720	1		76300	88379	1
DyHF-GP	g08	367	411	0.694	g21	2537	2768	0.764
DyHF		840	1223	1		68180	102180	1
DyHF-GP	g10	3194	3467	0.986	g23	1517	1678	0.921
DyHF		121380	142652	1		134260	161145	1
DyHF-GP	g11	18	40	0.123	g24	872	1048	0.783
DyHF		3500	5768	1		11620	14284	1
DyHF-GP	g12	81	191	0.575				
DyHF		420	3012	1				

TABLE II. FUNCTION ERROR VALUES AND FEASIBLE RATE FOR TEST FUNCTION G04,G05,G06,G08,G10,G11

Algorithm	DyHF-GP	DyHF	DyHF-GP	DyHF	DyHF-GP	DyHF
Prob.	g04		g05		g06	
Max FES	$5*10^3$	$5*10^3$	$3*10^3$	$5*10^3$	$5*10^3$	$5*10^3$
Best	0.0000e+00	4.5215e+01	1.6358e-05	9.3455e+00	2.0476e-07	3.2389e+00
Mean	0.0000e+00	1.6291e+02	3.5289e-05	2.1005e+02	5.2381e-06	9.4927e+01
Feasi. Rate	1	0.693	0.936	0	0.921	0.286
Prob.	g08		g10		g11	
Max FES	$1*10^3$	$5*10^3$	$5*10^3$	$5*10^3$	$1*10^3$	$5*10^3$
Best	1.7539e-08	1.9667e-08	-6.3665e-12	4.2652e+03	2.9022e-08	2.0256e-05
Mean	6.3215e-08	2.5249e-08	-3.2548e-11	8.9299e+03	6.4857e-07	2.0494e-04
Feasi. Rate	0.829	0.329	0.764	0	0.707	0.093

TABLE III. FUNCTION ERROR VALUES AND FEASIBLE RATE FOR TEST FUNCTION G12,G15,G16,G18,G21,G23

Algorithm	DyHF-GP	DyHF	DyHF-GP	DyHF	DyHF-GP	DyHF
Prob.	g12		g15		g16	
Max FES	$1*10^3$	$5*10^3$	$1*10^3$	$5*10^3$	$4*10^3$	$5*10^3$
Best	0.0000e+00	2.6445e-06	1.6741e-05	7.3492e-02	1.3574e-12	6.0594e-02
Mean	0.0000e+00	5.2996e-05	8.2475e-04	2.4183e-01	5.1285e-11	1.4394e-01
Feasi. Rate	1	0.707	0.857	0	0.936	0.186
Prob.	g18		g21		g23	
Max FES	$3*10^3$	$5*10^3$	$4*10^3$	$5*10^3$	$3*10^3$	$5*10^3$
Best	2.2204e-16	5.4452e+00	2.0233e-05	1.1590e+02	4.5869e-05	-2.2944e+02
Mean	7.2642e-15	4.1775e-01	3.2791e-04	1.6414e+02	2.1248e-05	-2.9184e+02
Feasi. Rate	0.886	0	0.936	0	0.948	0

TABLE IV. FUNCTION ERROR VALUES AND FEASIBLE RATE FOR TEST FUNCTION G24

Prob. Error	g24	
	DyHF-GP	DyHF
Max FES	$2*10^3$	$5*10^3$
Best	1.9654e-07	1.9054e-03
Mean	7.2421e-06	1.3918e-02
Feasible Rate	0.900	0.764