

A Historical Experience Surrogate Model Assisted Particle Swarm Optimization for Expensive Black-box Problems

Chongzheng Na ^a, Huixin Liu ^{*}

School of Information and Control, Shenyang Institute of Technology, Fushun, China

^anachongzheng@outlook.com

^{*} Corresponding Author Email: 591667038@qq.com

Abstract. This paper proposed a new historical experience surrogate model assisted particle swarm optimization method. This method extends the particle swarm optimization by adding new surrogate-based phase. In the classic phase, the particle swarm optimization runs the same way as the original algorithm, and the real function value evaluated are collected into the global database. In the surrogate phase, sub-swarms are generated following the distribution of the history data and evaluated by the surrogate model(s). The purpose of the surrogate phase is to explore the possible better solutions of the searching history. Also, the surrogate model(s) have the ability of accelerating the intelligence algorithms. Nevertheless, considering the time complexity of training and evaluating the surrogate model(s), the original problem should be expensive to evaluate or driven by data, which are same as many real-world problems.

Keywords: Black-box optimization, Particle swarm optimization, Surrogate model, Surrogate-assisted algorithm.

1. Introduction

In the recent decades, evolutionary algorithms (EAs), such as particle swarm optimization (PSO), grey wolf optimizer (GWO), differential evolution (DE), teaching-learning-based optimization algorithm, whale optimization algorithms and their variants have been proposed to solve various kinds of real-world optimization problems. Evolutionary algorithms can be known as a bunch of derivative-free optimization algorithms, which are mathematical optimization algorithms that does not use derivative information in the classical sense to find optimal solutions [1]. Applying EAs to the real-world problems sound to be a reasonable solution as the derivative information of most real-world problems can rarely be acquired easily. However, most EAs usually need a lot of function evaluations to find an optimization solution, and this seriously limits their application to solve the expensive problems as a single FE of these problems may take minutes to hours. [2] These expensive problems are typically simulate-based, data-driven or noisy. As the result, a bunch of EAs are proposed, aiming reducing the computing time or reducing the negative impact of uncertainty using surrogate model, which are known as surrogate assisted evolutionary algorithms (SAEAs). Various SAEAs are developed:

[3] proposed a heuristic PSO algorithms with Radial Basis Function Network (RBFN). [4] developed an algorithm framework bases genetic algorithm and surrogate models. [5] extended PSO algorithms with a two-layer surrogate framework. [6] employed the evolutionary algorithms with the classification-based surrogate to solve multi-objective optimization problem. Researchers suggest that SAEAs can greatly improve computationally efficient of original EAs and efficiently solve many complex real-world optimization problems [7].

At the same time, the uncertainty bring by the surrogate model is a double-edged sword. On the one hand, the same number of evaluations could be done to get closer to the global optimum or to solve more difficult optimization problems. On the other hand, the approximation error of approximation models may lead algorithms converge to false optima and weaken the prospecting ability [8] [9]. Hence, making use of the advantage and overcoming the disadvantage are constantly research direction of this topic.

This paper proposed a historical experience surrogate model assisted particle swarm optimization, which extends the particle swarm optimization by adding new surrogate-based phase. In the classic phase, the particle swarm optimization runs the same way as the original algorithm, and the real function value evaluated are collected into the global database. In the surrogate phase, sub-swarms are generated following the distribution of the history data and evaluated by the surrogate model(s). Like people can get benefit from reviewing history, the sub-swarms repeat the converging of the particle swarm using the newest experience. With the surrogate model(s), the sub-swarms can explore the solution space without real function evaluation. It means that the sub-swarms could be much larger swarm than the primary swarm and continuously explore if their approximation error allow. The best results of the surrogate phase will merge into the primary swarm. The two phases repeat until the algorithm end. In section II, a brief overview of techniques related will be discussed. In section III proposed method will be presented. In section IV, the benchmark function is deployed to evaluate the proposed method. In the last section, the paper is concluded, and the future will be discussed.

2. Techniques Related

2.1 Particle Swarm Optimization

Typically, a continuous nonlinear problem can be which can be define as:

$$\text{minimize } f(\mathbf{x}) \quad (1)$$

$$\text{s.t. } \mathbf{x}_{lb} \leq \mathbf{x} \leq \mathbf{x}_{ub} \quad (2)$$

where $\mathbf{x}$ is the decision variable vector, $f(\mathbf{x})$ is the objective function, $\mathbf{x}_{lb}$ and $\mathbf{x}_{ub}$ are the upper and lower boundary vectors of $\mathbf{x}$.

Particle Swarm Optimization, which was firstly proposed in 1995[10], is an evolutionary algorithm as a stylized representation of the movement of organisms in a bird flock or fish school. This algorithm starts randomly with several particles in the search space. The movement of each particle is guided by two special particles: particle best(pbest) and global best(gbest). The pbest records the best position that one particle has found. The gbest records the best position that the whole swarm has found. The movement of each particle can be defined as:

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + c_1 r_1 (\mathbf{pbest}_i^t - \mathbf{x}_i^t) + c_2 r_2 (\mathbf{gbest}^t - \mathbf{x}_i^t) \quad (3)$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \quad (4)$$

where $\mathbf{x}_i^t$ is the position of particle i at iteration t , $\mathbf{v}_i^t$ is the velocity of particle i at iteration t , $\mathbf{pbest}_i^t$ c_1 and c_2 are the acceleration coefficients, r_1 and r_2 are random numbers in the range $(0,1]$, is the best historical position of particle i before iteration t , $\mathbf{gbest}^t$ is the best historical position of the whole swarm before iteration t .

This paper uses the PSO with inertia weight, which is the mostly used PSO variant. The movement of each particle can be defined as:

$$\mathbf{v}_i^{t+1} = w^t \mathbf{v}_i^t + c_1 r_1 (\mathbf{pbest}_i^t - \mathbf{x}_i^t) + c_2 r_2 (\mathbf{gbest}^t - \mathbf{x}_i^t) \quad (5)$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \quad (6)$$

$$w^t = w_{max} - \frac{t(w_{max} - w_{min})}{t_{max}} \quad (7)$$

where w^t is inertia weight, changes linearly, reducing from w_{max} to w_{min} .[11]

2.2 Support Vector Regression

The proposed method choice Support Vector Regression (SVR) as surrogate model. SVR is a regression form of Support vector machine (SVM), which is firstly proposed in 1998[12]. One of the widely used SVR implementation is ϵ -SVR [13], which can be described as:

$$\min_{w, b, \xi, \xi^*} \quad \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i + C \sum_{i=1}^l \xi_i^* \quad (8)$$

$$\text{subject to?} \quad w^T \phi(x_i) + b - z_i \leq \epsilon + \xi_i$$

$$z_i - w^T \phi(x_i) - b \leq \epsilon + \xi_i^*,$$

$$\xi_i, \xi_i^* \geq 0, i = 1, \dots, l.$$

$$\min_{\alpha, \alpha^*} \quad \frac{1}{2} (\alpha - \alpha^*)^T Q (\alpha - \alpha^*) + \epsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l z_i (\alpha_i - \alpha_i^*) \quad (9)$$

$$e^T (\alpha - \alpha^*) = 0$$

$$0 \leq \alpha_i, \alpha_i^* \leq C, i = 1, \dots, l$$

$Q_{ij} = K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$, $K(x, x')$ is the kernel function, which is supposed to be given

priority. The most commonly used kernel function is the Gaussian Kernel: $K(x, x') = e^{-\frac{x-x'^2}{2\sigma^2}}$

Approximate function of the original function is:

$$\hat{f}(x) = \sum_{i=1}^l (-a_i + a_i^*) K(x_i, x) + b \quad (10)$$

3. Historical Experience Surrogate Model Assisted Particle Swarm Optimization

In this paper, Historical Experience Surrogate Model Assisted Particle Swarm Optimization (HESMAPSO) is proposed. This algorithm is divided into two phases: the classic phase and the surrogate phase.

The classic phase is working the same strategy as the original PSO. Besides the searching, there are two purposes of this phase, one is to construct the global database. During the search, real function value is collected.

$$DB_g = \{(x_i, f(x_i)) | i \in G\} \quad (11)$$

Where G is a set of all particles evaluated in the classic phase? The other purpose is to guarantee the propection. As the in beginning and ending of HESMAPSO the uncertainty of the surrogate model is usually large, the classic phase could void the search dropping into the false optima.

In the surrogate phase, sub-swarms are generated following the distribution of the history data and evaluated by the surrogate model, which is trained using the samples in DB_g . This phase can be considered as reviewing search history using historical experience. It is the same concept as people reviewing our history. The subswarm can explore the solution space without real function evaluation. It means that the sub-swarms could be much larger swarm than the primary swarm and continuously explore if their approximation error allow.

Assuming the $x_i \in G_{la}$ obey the gaussian distribution $\mathcal{N}(\mu, \Sigma)$, the new subswarm can be generate:

$$x_{new} \sim (\mu, \Sigma) \quad (12)$$

G_{la} is particles set of last n_{la} generations, μ is the mean vector, Σ is the covariance matrix. This assumption is widely by many optimization algorithms, such as covariance matrix adaptation evolution strategy(CMA-ES), and proved to be valid.[14]

The subswarm explore the solution space using the SVM surrogate, which is constructed by global database DB_g . The best n_{swb} particles of each generation are collected as DB_{sw} . As the search progressing, the differences between particles distribution and training set distribution will be getting bigger. Approximation accuracy of the surrogate model is supposed to be tested. This paper proposed a strategy of evaluating the model accuracy without evaluate real-function:

$$err = \frac{2}{N(N-1)} \sum_{l < i < N} \sum_{i < j < N} ((f(\tilde{x}_i) < f(\tilde{x}_j)) \text{ xor } (\hat{f}(\tilde{x}_i) < \hat{f}(\tilde{x}_j))) \quad (13)$$

$$a \text{ xor } b = \begin{cases} 0 & \text{if } (a = \text{true}, b = \text{true}) \text{ or } (a = \text{false}, b = \text{false}) \\ 1 & \text{if } (a = \text{true}, b = \text{false}) \text{ or } (a = \text{false}, b = \text{true}) \end{cases} \quad (14)$$

N is the particles amount of the subswarm, $\tilde{x}_i$ is the nearest particle to x_i of DB_g in Euclidean distance

$$\tilde{x}_i = \underset{k \in G}{\operatorname{argmin}} (|x_i - x_k|) \quad (15)$$

The surrogate phase will be end and return to the classic phase while $err < err_{sw}$ or after n_{iter} generations. err_{sw} and n_{iter} are hyper-parameters. Then a lean set of DB_{sw} should be constructed. It should include best particles subswarm found and sparse enough avoiding fall into false optima. The i th particle of the selected set DB_{sel} can be described as:

$$x_i = \underset{x_i \in DB_{sw}}{\operatorname{argmin}} \left(f(x_i) + \sum_{l < j < i} \frac{M}{|x_i - x_j|^2} \right) \quad (16)$$

Fig. 1 shows the framework of HESMAPSO:

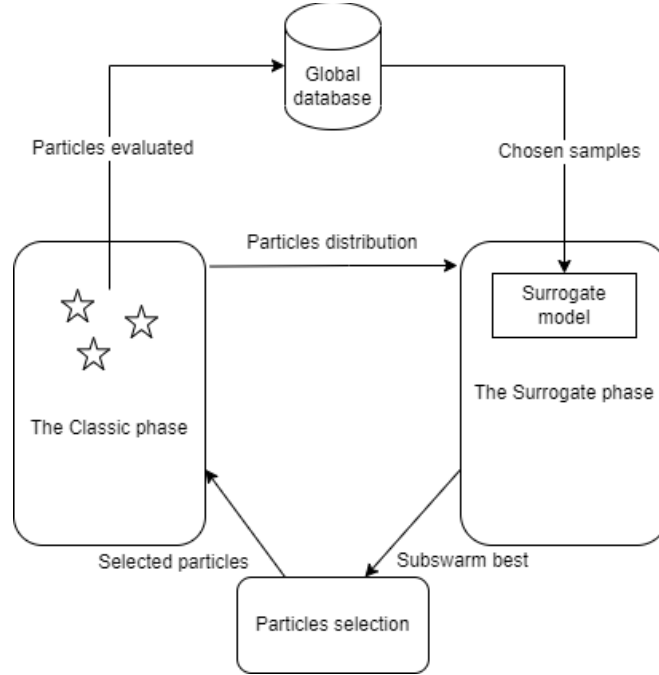


Figure 1. The framework of HESMAPSO

4. Historical Experience Surrogate Model Assisted Particle Swarm Optimization

The Rosenbrock benchmark function is chosen to evaluate the HESMAPSO. The Rosenbrock benchmark function, also referred to as the Valley or Banana function, is one of the most popular test problems for both black-boxed or gradient-based optimization algorithms. This function is unimodal, and the global minimum lies in a narrow, parabolic valley. However, even though this valley is easy to find, convergence to the minimum is difficult.

$$f(x) = \sum_{i=1}^{d-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right) \quad (17)$$

The global minimum is $f(x^*) = 0$ at $x^* = (1, 1, \dots, 1)$.

Table.1 and Figure.2 show the comparison between the original CPSO and HESMAPSO.

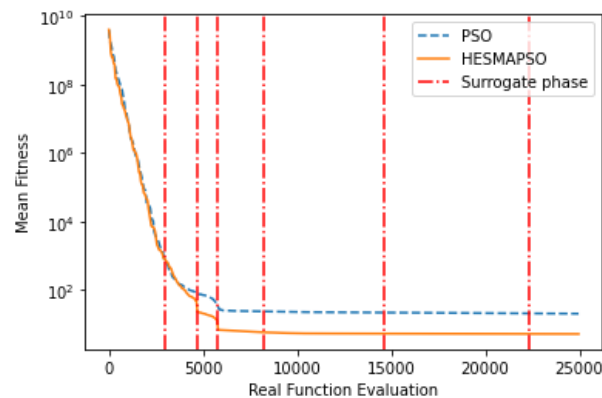


Figure.2 The comparison of HESMAPSO and original PSO

Results are the average of 45 times computations. Notice that the real function evaluations before entering the surrogate phase is different in every computation. Hence the surrogate phase lines are the average value.

Table 1. Results of the original CPSO and the HESMAPSO Table Type Styles

Algorithms	Performance			
	Best	Worst	Average	Std
PSO(t=5000)	1.45e00	2.40e02	6.20e01	5.82e01
HESMAPSO (t=5000)	4.92e00	7.50e01	8.01e00	6.70e00
PSO(t=10000)	9.90e-1	1.02e02	3.50e01	2.00e01
HESMAPSO (t=10000)	4.50e00	9.00e01	5.20e00	2.30e00

Let n presents the real function evaluation. The first surrogate phase starts at average $n = 3000$. Unexpectedly, the surrogate phase is failed to give better solution. The reason might be that at the very beginning, the train samples are sparsely in the solution space. Yet the search process of following classic phase seems to be accelerated. The reason is the solution DB_{sw} that merge into the main swarm prevent early convergence of the PSO. At $n \approx 4600$ and $n \approx 6000$, the surrogate phase is proved to be effectively, better solutions are found without new real function evaluated. At the same time, the prospection ability is enhanced since the average best-solution is remarkably better than the PSO. The last 3 surrogate phases seem to be not obviously influenced. The reason is the global surrogate model is hard to meet the criterion. It becomes increasingly difficult to enter the surrogate phase.

5. Conclusion

The historical experience surrogate model assisted particle swarm optimization, is to emulate human's reviewing history activity using surrogate model. Benchmark experiments prove that the proposed method can enhance the exploration and prospection ability of the original PSO and reducing real function evaluated to acquiring even better solution. The searching process could be accelerated depending on the complexity of the real function, which should be the same or larger than the complexity of training the surrogate model. As a result, the proposed method is appropriately to solve the real-world expensive problem.

Much work remains for future study: the global surrogate seems inefficiently to oversee the beginning and the ending searching stage. Application of local model in the proposed method may be a meaningful topic. Also, multi-subswarm is a direction. Subswarm could be generated by not only one distribution. different subswarms with different distribution might enhance the exploration and prospection ability further than the proposed method.

Acknowledgments

This work was financially supported by Natural Science Foundation of Liaoning Province of China (2019-ZD-0342).

References

- [1] A. Slowik and H. Kwasnicka, "Evolutionary algorithms and their applications to engineering problems," *Neural Comput & Applic*, vol. 32, no. 16, pp. 12363–12379, Aug. 2020.
- [2] J.-S. Pan, N. Liu, S.-C. Chu, and T. Lai, "An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems," *Information Sciences*, vol. 561, pp. 304–325, Jun. 2021.
- [3] H. Yu, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Surrogate-assisted hierarchical particle swarm optimization," *Information Sciences*, vol. 454, pp. 59–72, 2018.
- [4] X. Cai, L. Gao, and X. Li, "Efficient Generalized Surrogate-Assisted Evolutionary Algorithm for High-Dimensional Expensive Problems," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 365–379, Apr. 2020.
- [5] X. Ji, Y. Zhang, D. Gong, and X. Sun, "Dual-surrogate-assisted cooperative particle swarm optimization for expensive multimodal problems," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 794–808, 2021.
- [6] M. Zhao et al., "A classification-based surrogate-assisted multiobjective evolutionary algorithm for production optimization under geological uncertainty," *SPE Journal*, vol. 25, no. 05, pp. 2450–2469, 2020.
- [7] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.
- [8] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing Surrogate-Assisted Evolutionary Computation," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 329–355, Jun. 2010.
- [9] Y. S. Ong, P. B. Nair, A. J. Keane, and K. W. Wong, "Surrogate-Assisted Evolutionary Optimization Frameworks for High-Fidelity Engineering Design Problems," in *Knowledge Incorporation in Evolutionary Computation*, Y. Jin, Ed. Berlin, Heidelberg: Springer, 2005, pp. 307–331.
- [10] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43.
- [11] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII*, Berlin, Heidelberg, 1998, pp. 591–600.
- [12] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.
- [13] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.
- [14] N. Hansen, "The CMA evolution strategy: a comparing review," *Towards a new evolutionary computation*, pp. 75–102, 2006.