# Enhancing Cooperative Coevolution for Large Scale Optimization by Adaptively Constructing Surrogate Models

Bei Pang, Zhigang Ren
Autocontrol Institute, Xi'an Jiaotong University
Xi'an, Shaanxi, 710049, P. R. China
beibei@stu.xjtu.edu.cn, renzg@mail.xjtu.edu.cn

Yongsheng Liang, An Chen
Autocontrol Institute, Xi'an Jiaotong University
Xi'an, Shaanxi, 710049, P. R. China
liangyongsheng, chenan123@stu.xjtu.edu.cn

## ABSTRACT

It has been shown that cooperative coevolution (CC) can effectively deal with large scale optimization problems (LSOPs) through a divide-and-conquer strategy. However, its performance is severely restricted by the current context-vector-based sub-solution evaluation method because this method requires too many computation resources. To alleviate this issue, this study proposes an adaptive surrogate model assisted CC framework which adaptively constructs surrogate models for different sub-problems by fully considering their characteristics. By this means, the computation cost could be greatly reduced without significantly sacrificing evaluation quality. Empirical studies on IEEE CEC 2010 large scale benchmark suit show that the concrete algorithm based on this framework performs well.

## CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; *Search methodologies*

## KEYWORDS

cooperative coevolution, surrogate model

## 1 INTRODUCTION

Large-scale optimization problems (LSOPs) are becoming more and more popular in scientific research and engineering applications. Due to the curse of dimensionality, conventional evolutionary algorithms (EAs) lose their efficiency in solving this kind of problems. Taking the idea of 'divide-and-conquer', cooperative coevolution (CC) provides a natural way for solving LSOPs. It first decomposes the original LSOP into several smaller and simpler sub-problems, and then solves the LSOP by cooperatively optimizing all the sub-problems with a conventional EA.

It is known that CC mainly focuses on black-box LSOPs. This means that all the sub-problems obtained through decomposition do not own separate or explicit objective functions. To evaluate the sub-solutions, now all the CC algorithms adopt a context-vector-based method [1]. But generally, a very limited number of solution simulations are allowed for a practical LSOP since even a single simulation is relatively time-consuming. With so little computation resource, it is challenging for a CC algorithm to produce high-quality solutions. To tackle this issue, this study proposes an Adaptive Surrogate Model Assisted CC framework (ASMCC) which can greatly reduce the requirement of function evaluations (FEs) in CC by adaptively constructing surrogate models for the sub-problems with different characteristics.

## 2 DESCRIPTION OF THE ASMCC ALGORITHM

ASMCC solves the LSOP by optimizing the separable and nonseparable sub-problems in different ways. In this paper, we adopt success-history based adaptive differential evolution (SHADE) [2] as the basic EA, polynomial regression (PR) and radial basis function (RBF) [3] as the surrogate models. The procedure of the ASMCC algorithm is presented in Algorithm 1.

**Algorithm 1**: ASMCC

1. Generate a decomposition $\boldsymbol{x} \rightarrow \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_k\}$, randomly initialize $\boldsymbol{x}^c$;
2. **for** each 1-dimensional sub-problem $g$ **do** $(x^*_g) = PR(g)$;
3. Initialize the best overall solution $\boldsymbol{x}^*$ based on the obtained best solutions of the 1-dimensional sub-problems;
4. Initialize the parameters, randomly initialize population $P_g$ and database $D^n_g$ for all of the nonseparable sub-problems;
5. **while** the termination condition is not met **do**
6. Determine the sub-problem $g$ to be optimized;
7. $(P_g, D^n_g, \boldsymbol{x}^*) \leftarrow RBF-SHADE(P_g, D^n_g, \boldsymbol{x}^*, g)$;
8. Output $\boldsymbol{x}^*$.

In Algorithm 1, the separable sub-problems are further divided into several 1-dimensional sub-problems first (step1), and then steps 2-4 optimize the 1-dimensional sub-problems with the two-layer PR search process, where a fixed context vector $\boldsymbol{x}^c$ is used for simplicity (step 1). After that, the nonseparable sub-problems are optimized by RBF-SHADE (steps 4-7) with the rest of the computation resources, and the best overall solution $\boldsymbol{x}^*$ is used as the context vector (step 3). Step 6 selects the nonseparable sub-problems with a round-robin method. Algorithm 2 and Algorithm 3 present the details of the two-layer PR process and RBF-SHADE algorithm, respectively.

In Algorithm 2, the first layer (steps 1-3) finds out a small region which covers the optimal solution of sub-problem $g$ by constructing a global PR model, and then the second layer (steps 4-7) finds out the final optimal solution of sub-problem $g$ within the small region by constructing several local PR models. In the first layer, the fitness distance correlation (FDC) is used to determine the degree of the PR model ($N$) (step 2).


*GECCO'18 Companion*, July 15-19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205797

**Algorithm 2**: $(x_g^*) = PR(g)$

1. Initialize $D_g^s$ with $d^s$ uniformly generated sub-solutions $x_g$ within $[lb_g, ub_g]$ and evaluate them with $e(x_g) = f(\boldsymbol{x}^c) - f(\boldsymbol{x}^c \mid x_g)$;
2. **if** $\left|FDC_g\right| > \varepsilon$, $N$=2; **else** $N$=5;
3. Build $P(x_g)$ with $D_g^s$, find out the best solution $x_g^*$ of the $g$th sub-problem, and define the new search region $[lb_g', ub_g']$, where $lb_g'(ub_g') = lb_g(ub_g)/r + x_g^*$;
4. Generate new $D_g^s$, divide $[lb_g', ub_g']$ into $\lfloor d^s/6 \rfloor$ sub-regions;
5. **for** sub-region $i$=1:$\lfloor d^s/6 \rfloor$ **do**
6. Build $P_i(x_g)$ based on $D_g^s$ ($N$=5), find out $x_g^{i^*}$ of each sub-region $i$;
7. $i = \arg\max_{i=1:\lfloor d^s/6 \rfloor} (P_i(x_g^{i^*}))$, **if** $f(x^c \mid x_g^{i^*}) < f(x^c \mid x_g^*)$ **then** $x_g^* \leftarrow x_g^{i^*}$;
8. Return $x_g^*$;

**Algorithm 3**: $(P_g, D_g^n, \boldsymbol{x}^*) \leftarrow RBF - SHADE(P_g, D_g^n, \boldsymbol{x}^*, g)$

1. Build a RBF model for the $g$th sub-problem with $D_g^n$;
2. **for** each sub-solution $\boldsymbol{x}_g^i \in P_g$ **do**
3. Evaluate $\boldsymbol{x}_g^i$ and $\boldsymbol{u}_g^i$ with $\overline{e}(\boldsymbol{x}_g^i)$ and $\overline{e}(\boldsymbol{u}_g^i)$ provided by RBF;
4. Select $q$ best trial vectors from the group of $\boldsymbol{u}_g^i, i = 1,2,\cdots,p$ and store them into $Q_g$;
5. Reevaluate each $\boldsymbol{u}_g^i \in Q_g$ with $e(\boldsymbol{u}_g^i)$ and modify $\overline{e}(\boldsymbol{u}_g^i) \leftarrow e(\boldsymbol{u}_g^i)$;
6. Update the parameters of SHADE based on $\overline{e}(\boldsymbol{x}_g^i)$ and $\overline{e}(\boldsymbol{u}_g^i)$;
7. Update $D_g^n$ and $P_g$ with $Q_g$, find out the best sub-solution $\boldsymbol{x}_g^b$ in $P_g$;
8. **if** $e(\boldsymbol{x}_g^b) > 0$ **then** Update $f(\boldsymbol{x}^*) \leftarrow f(\boldsymbol{x}^* \mid \boldsymbol{x}_g^b)$, $\boldsymbol{x}^* \leftarrow \boldsymbol{x}^* \mid \boldsymbol{x}_g^b$;
9. Return $P_g, D_g^n, \boldsymbol{x}^*$;

In Algorithm 3, the population $P_g$ records $p$ best sub-solutions obtained so far, and the database $D_g^n$ records $d^n$ most recently real evaluated sub-solutions. For the generation of the trial vectors $\boldsymbol{u}_g^i$ and concrete update rule of the parameters of SHADE, readers can refer to [2].

## 3 EXPERIMENTAL AND CONCLUSION

Functions $F_1$-$F_{13}$ in CEC 2010 large scale benchmark suite were adopted in our experiments, and two decomposition methods, i.e., ideal decomposition and VGDA-D [4] were used in our experiments. The result of each algorithm on a function was calculated based on 25 independent runs, and a maximum number of $3.0\times10^5$ FEs was used as the termination condition of a run of an algorithm unless otherwise mentioned.

There are several parameters in ASMCC. For the two-layer PR search process, we suggest setting $d^s$ to 100, $\varepsilon$ to 0.8. As for $r$, it is set to 10 and 15 when the fifth-order and second-order PR model are used, respectively. As for RBF-SHADE, the achieve size $d^n$ is set to $5D$, where $D$ is the dimensionality of the corresponding sub-problem, and the population size $p$ was set to 100. As for $q$, our prior experiments suggest setting it to 10.

To show the superiority of ASMCC, we compared it with three other CC algorithms and one non-CC algorithm. Table 1 summarizes the results obtained by the algorithms, where SHADE-CC is a conventional CC algorithm which adopts SHADE as the optimizer, and PS-CC is a CC algorithm which adopts the two-layer PR process and SHADE to solve the separable sub-problems and the non-separable sub-problems, respectively. The results of CC-I [5] and MA-SW-Chains [6] were directly taken from their original papers, and the number of FEs they consumed are $3.0\times10^6$, which is ten times as much as ASMCC, PS-CC and SHADE-CC.

The last row of Table 1 lists the ranking of the four algorithms according to Friedman test, it can be seen that ASMCC performs best under both kind of decomposition methods. From the results of SHADE-CC, PS-CC and ASMCC, it can be observed that the two-layer PR process and the RBF model are really feasible and efficient. And it is clearly that ASMCC can obtain better results than CC-I and MA-SW-Chains even with less FEs.

**Table 1: The average function values ± standard deviations obtained by the algorithms on CEC 2010 functions $F_1$-$F_{13}$**

| $F$ | CC-I | SHADE-CC-I | PS-CC-I | ASMCC-I | MA-SW-Chains | SHADE-CC-D | PS-CC-D | ASMCC-D |
|---|---|---|---|---|---|---|---|---|
| $F_1$ | 3.50e+11 ± 2.0e+10$^-$ | 1.05e+06 ± 1.03e+05$^-$ | **7.05e-14 ± 1.59e-15**$^\approx$ | **7.05e-14 ± 1.59e-15** | **2.10e−14 ± 1.99e−14**$^+$ | 1.16e+06 ± 8.08e+04$^-$ | 7.05e-14 ± 1.56e-15$^\approx$ | 7.05e-14 ± 1.53e-15 |
| $F_2$ | 9.40e+03 ± 2.1e+02$^-$ | 6.51e+03 ± 5.69e+01$^-$ | **7.32e-06 ± 4.44e-07**$^\approx$ | **7.32e-06 ± 4.44e-07** | 8.10e+02 ± 5.88e+01$^-$ | 6.51e+03 ± 7.29e+01$^-$ | **7.19e-06 ± 5.10e-07**$^\approx$ | **7.19e-06 ± 5.11e-07** |
| $F_3$ | 2.00e+01 ± 4.4e−02$^-$ | 1.52e+01 ± 1.76e-01$^-$ | **5.61e-03 ± 2.51e-02**$^\approx$ | **5.61e-03 ± 2.51e-02** | **7.28e−13 ± 3.40e−13**$^+$ | 1.49e+01 ± 3.69e-01$^+$ | 1.50e+01 ± 3.69e-01$^+$ | – |
| $F_4$ | 3.40e+14 ± 7.5e+13$^-$ | 6.92e+13 ± 1.15e+13$^-$ | 6.05e+11 ± 2.66e+11$^-$ | **8.67e+10 ± 3.94e+10** | 3.53e+11 ± 3.12e+10$^-$ | 7.23e+13 ± 1.34e+13$^-$ | 8.21e+11 ± 3.30e+11$^-$ | **7.86e+10 ± 2.90e+10** |
| $F_5$ | 4.90e+08 ± 2.4e+07$^-$ | 4.01e+08 ± 1.91e+07$^-$ | 1.26e+08 ± 1.46e+07$^-$ | **1.12e+08 ± 2.61e+07** | 1.68e+08 ± 1.04e+08$^-$ | 4.04e+08 ± 1.83e+07$^-$ | 1.27e+08 ± 1.49e+07$^-$ | **1.18e+08 ± 1.87e+07** |
| $F_6$ | 1.10e+07 ± 7.5e+05$^-$ | 1.05e+06 ± 1.81e+05$^-$ | **1.11e-02 ± 3.54e-02**$^+$ | 3.87e+05 ± 7.55e+05 | 8.14e+04 ± 2.84e+05$^+$ | 1.08e+06 ± 2.43e+05$^-$ | **2.36e-02 ± 5.36e-02**$^+$ | 9.18e+05 ± 1.18e+06 |
| $F_7$ | 7.70e+10 ± 9.6e+09$^-$ | 2.68e+10 ± 5.27e+09$^-$ | 7.91e+04 ± 8.93e+04$^-$ | **1.63e-03 ± 2.62e-03** | 1.03e+02 ± 8.70e+01$^-$ | 2.51e+10 ± 4.73e+09$^-$ | 1.78e+05 ± 3.72e+05$^-$ | **2.26e-03 ± 7.02e-03** |
| $F_8$ | 1.80e+14 ± 9.3e+13$^-$ | 3.46e+09 ± 2.39e+09$^-$ | 3.51e+07 ± 2.51e+07$^-$ | **9.57e+05 ± 1.74e+06** | 1.41e+07 ± 3.68e+07$^-$ | 3.50e+09 ± 1.79e+09$^-$ | 3.08e+07 ± 1.72e+07$^-$ | **7.17e+05 ± 1.48e+06** |
| $F_9$ | 9.40e+08 ± 7.1e+07$^-$ | 6.13e+08 ± 4.14e+07$^-$ | 3.34e+08 ± 2.29e+07$^-$ | **1.14e+07 ± 1.55e+06** | 1.41e+07 ± 1.15e+06$^-$ | 6.52e+08 ± 4.83e+07$^-$ | 3.65e+08 ± 2.56e+07$^-$ | **1.17e+07 ± 1.07e+06** |
| $F_{10}$ | 4.80e+03 ± 6.7e+01$^-$ | 7.29e+03 ± 8.80e+01$^-$ | 3.75e+03 ± 5.53e+01$^-$ | **1.10e+03 ± 7.03e+01** | 2.07e+03 ± 1.44e+02$^-$ | 7.35e+03 ± 5.69e+01$^-$ | 3.80e+03 ± 4.72e+01$^-$ | **1.11e+03 ± 9.68e+02** |
| $F_{11}$ | 4.10e+01 ± 1.5e+00$^-$ | 2.74e+01 ± 8.35e-01$^-$ | **9.00e-01 ± 1.16e-01**$^+$ | 6.00e+00 ± 2.33e+00 | 3.80e+01 ± 7.35e+00$^-$ | **1.63e+01 ± 3.41e-01**$^+$ | 1.63e+01 ± 4.21e-01$^+$ | 2.33e+01 ± 4.44e+00 |
| $F_{12}$ | 4.90e+05 ± 3.4e+04$^-$ | 2.98e+05 ± 1.20e+04$^-$ | 1.76e+05 ± 8.98e+03$^-$ | **1.82e+03 ± 1.27e+03** | **3.62e−06 ± 5.92e−07**$^+$ | 3.09e+05 ± 1.71e+04$^-$ | 2.00e+05 ± 1.13e+04$^-$ | 1.70e+03 ± 7.31e+02 |
| $F_{13}$ | 1.50e+07 ± 4.1e+06$^-$ | 3.28e+04 ± 5.77e+03$^-$ | 3.76e+03 ± 1.46e+03$^-$ | **6.47e+02 ± 1.97e+02** | 1.25e+03 ± 5.72e+02$^-$ | 1.25e+05 ± 2.11e+04$^-$ | 1.03e+04 ± 2.20e+03$^-$ | **8.01e+02 ± 3.50e+02** |
| +/≈/− | 0/0/13 | 0/0/13 | 2/3/8 | – | 4/0/9 | 2/0/11 | 3/2/8 | – |
| Ranking | 3.9231 | 3.0769 | 1.7692 | 1.2308 | 2.1667 | 3.7500 | 2.5000 | 1.5833 |

"**−**", "**+**" and "**≈**" respectively denote that the performance of the corresponding algorithm is worse than, better than or similar to that of ASMCC according to Cohen's $d$ effect size.

## ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 61105126, in part by the Postdoctoral Science Foundation of China under Grants 2014M560784 and 2016T90922, and in part by the Postdoctoral Science Foundation of Shaanxi Province.

## REFERENCES

[1] F. van den Bergh and A. P. Engelbrecht. 2004. A Cooperative Approach to Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, 8 (3), 225–239.

[2] R. Tanabe and A. Fukunaga. 2013. Success-History Based Parameter Adaptation for Differential Evolution. In *Proceedings of IEEE Congress on Evolutionary Computation* (CEC'13). Cancun, Mexico, 71-78.

[3] A. Díaz-Manríquez, G. Toscano and C. A. C. Coello. 2017. Comparison of Metamodeling Techniques in Evolutionary Algorithms. *Soft Computing,* 21, 5647-5663.

[4] Z. Ren, A. Chen, L. Wang, Y. Liang and B. Pang. 2017. An Efficient Vector-Growth Decomposition Algorithm for Cooperative Coevolution in Solving Large Scale Problems. In *Proceedings of the genetic and evolutionary computation conference* (GECCO'17). Berlin, German, 41-42.

[5] M. Yang, M. N. Omidvar, C. Li, X. Li, Z. Cai and B. Kazimipour. 2017. Efficient Resource Allocation in Cooperative Co-evolution for Large-Scale Global Optimization. *IEEE Transactions on Evolutionary Computation,* 21(4), 493–505.

[6] D. Molina, M. Lozano, and F. Herrera. 2010. MA-SW-Chains: Memetic Algorithm Based on Local Search Chains for Large Scale Continuous Global Optimization. In *Proceedings of IEEE Congress on Evolutionary Computation* (CEC'10), 1-8.