A surrogate-assisted hybrid optimization algorithm enhanced by opposition-based learning and its variants

Zhaoqi Ling
Academy of Mathematics and Systems Science, Chinese Academy of Sciences,
Beijing, China
University of the Chinese Academy of Sciences, China
lingzhaorqi18@mails.ucas.edu.cn

Abstract. Surrogate-Assisted Evolutionary Algorithms (SAEAs) are effective approaches to solve computationally expensive optimization problems by remarkably reducing real fitness evaluations. In this work, we proposed a surrogate-assisted hybrid optimization algorithm via combining a famous hierarchical SAEA Framework ESAO and a hybrid teaching-learning based optimization (TLBO) algorithm TLBO-SM. In addition, opposition-based learning (OBL) and its variants are used to enhance the global search ability. Experimental results on benchmark problems show that our method can outperform state-of-the-art SAEAs on most benchmark problems with the significant enhancement made by a recently proposed OBL variant.

1. Introduction
Surrogate models are built on evaluated points to partly replace computationally expensive fitness function for high-dimensional expensive optimization problems (HEOPs). The widely studied surrogate models, including radial basis function (RBF) [1] and Gaussian process (Kriging), yield various surrogate-assisted evolutionary algorithms (SAEAs). Existing SAEAs can be roughly divided into the following three categories: global SAEAs, local SAEAs, and hierarchical SAEAs. Global surrogate models try to approximate the expensive fitness function in the entire search space of the problem. Local surrogate models focus on located local promising regions for the purpose of ensuring the approximation accuracy. SAEAs using a single EA to deal with a complicated problem may follow similar trajectories and fall into a local optimum. Hierarchical SAEAs ensemble the exploration capability of the global surrogate and the exploitation capability of the local surrogate together. SAEAs have got great success in tackling low- and medium-dimensional expensive problems, while only few hybrid SAEAs focus on the HEOPs. A recently proposed hierarchical SAEA called evolutionary sampling assisted optimization (ESAO) [2] builds both a global RBF model and a local RBF model to assist differential evolution (DE) to accordingly conduct global search and local search. In our work, we will separately construct RBF models as global and local surrogates in the proposed SAEA just like ESAO framework.

Teaching-learning based optimization (TLBO) algorithm [3] has been used as a surrogate model in several recent studies. [4] proposes a hierarchical SAEA by combining TLBO and DE. The restart strategy used in their paper is excluded since it’s of no help to our proposed SAHO. [5] presents Kriging-assisted TLBO to solve computationally expensive constrained problems. In order to improve efficiency and avoid trapping into a local optimum, we adopt hybrid TLBO with Nelder–Mead simplex (TLBO-SM) [6] to design a SAEA for solving HEOPs, which can take advantages of the exploration ability of
a population-based metaheuristic TLBO and the exploitation ability of a local refinement procedure Nelder–Mead simplex algorithm [7].

The core idea of opposition-based learning (OBL) strategy [8] is to seek for a better candidate solution by evaluating the opposition solution with respect to the original solution, through which the exploitation space is enriched. However, if there is a local optimum in the space from the current number to the opposite number, OBL strategy have the tendency to converge the search space to this local optimum position. In this study, the recently proposed oppositional mutual learning (OML) strategy [9] is shown to be one of the best OBL variants adopted for improving SAEAs. The algorithm proposed by [10] adopts OBL and a hybrid adaptive strategy with SAEA to improve a multi-objective evolutionary algorithm. In our work, more variants of OBL are considered and the best one is adopted with our SAEA to achieve better performance for HEOPs.

The main work of this paper can be summarized as follows.

- We propose a surrogate-assisted hybrid optimization (SAHO) algorithm via adopting the effective hierarchical SAEA framework ESAO with a hybrid TLBO algorithm TLBO-SM.
- We enhance the performance of SAHO via OBL variants. A recently proposed OBL variant OML works best and helps SAHO to be competitive to state-of-the-art SAEAs and outperform them on most of the benchmark functions.

2. Background

2.1. Teaching-learning based algorithm

TLBO approach is described as follows. In the teaching phase, the teacher \( x_{\text{best}} \), which is the learner with the best fitness, guide the learners \( x_i \) with the following learning law:

\[
x_i' = x_i + r \times (x_{\text{best}} - TF \times x_{\text{mean}})
\]

(1)

where each element of \( r \) is uniformly randomly taken from \([0, 1]\), TF is the teaching factor with each element taken from \([1, 2]\) randomly and \( x_{\text{mean}} \) is the mean of all learners. Then \( x_i' \) will replace \( x_i \) iff \( f(x_i') < f(x_i) \), where \( f \) is the fitness function. In the learning phase, learners \( x_i \) get improvement through interactions among themselves with the following learning law:

\[
x_m' = \begin{cases} x_m + r \times (x_m - x_n) & \text{if } f(x_m) < f(x_n) \\ x_m + r \times (x_n - x_m) & \text{otherwise} \end{cases}
\]

(2)

where \( x_n \) is randomly selected from all the learners except \( x_m \) and the definition of other parameters is independent from the teaching phase.

2.2. Opposition-based learning and its variants

Opposition-based Learning tries to find a better solution by simultaneously considering the original solution \( x \) and its opposite solution \( x' \) with respect to the whole search space:

\[
x_i' = a_i + b_i - x_i
\]

where the \( i^{th} \) element of \( x \) in the whole search space is in \([a_i, b_i]\).

The OBL variant OML considered in this word version paper is described as follows. The generated OML opposite solution \( x_i' \) of \( i^{th} \) individual \( x_i \) in the population is defined as:

\[
x_i' = \begin{cases} x_i + r_i^1 \times (r_i^2 \times (Lu + Ll - x_i) - x_i) & \text{if } \text{rand}(0, 1) < 0.5 \\ x_i + r_i^2 \times (x_m - x_i) & \text{otherwise} \end{cases}
\]

where \( i^{th} \) individual \( x_i \in [Ll, Lu] \), \( Lu \) and \( Ll \) are upper and lower bound vectors for each element of individuals in the population and \( x_m \) is a randomly selected individual in the population except \( x_i \). The random numbers \( r_i \) used here are all independently taken from \([0, 1]\).
3. Proposed Method

Without loss of generality, we denoted opposition-based learning as OBL in the algorithm instead of any special variant we use in the experiment. We will replace OBL with the special variant used in the experiments if necessary.

First, Latin hypercube sampling (LHS) is used to initialize the samples and the true fitness function \( f_{\text{real}} \) is used to evaluate their fitness values. Then, OBL (or one of its variant) is used to generate opposite samples from initial samples. All of them are evaluated with \( f_{\text{real}} \) and stored into the database with their fitness value. Next, we form a population by selecting solutions in the database from best to worst bases on their fitness.

In the stage of global search, a global RBF surrogate model \( f_{\text{global}} \) is constructed with all samples in the database. Then, we perform TLBO algorithm by fitness estimated from \( f_{\text{global}} \). After that, OBL generation is performed, and we select the best one \( x_p \) from all individuals generated in this stage based on the population. Finally, we evaluated \( x_p \) with \( f_{\text{real}} \) and store it into the database. This global search stage will continue if \( x_p \) cannot lead to a better solution ever, otherwise it will switch to the local search stage.

In the stage of local search, a local RBF surrogate model \( f_{\text{local}} \) is constructed with N best samples in the database. Then, we perform Nelder–Mead simplex search starting from the best solution in the database by fitness estimated from \( f_{\text{local}} \) and denote the result as \( x_s \). Finally, we evaluated \( x_s \) with \( f_{\text{real}} \) and store it into the database. This local search stage will continue if \( x_s \) cannot lead to a better solution ever, otherwise it will switch to the global search stage.

The algorithm will stop once the limit of function evaluations is exceeded.

Algorithm 1 presents the pseudocode for our proposed method OBL-SAHO algorithm.

---

**Algorithm 1: Opposition-Based-Learning-enhanced Surrogate-Assisted Hybrid Optimization (OBL-SAHO) algorithm**

**Result:** Output the best sample \( x_b \) in the database

1. Generate the initial samples using LHS, evaluate them with the real fitness function \( f_{\text{real}} \) and put them into the database.
2. Perform OBL generation with samples in the database, get their fitness from \( f_{\text{real}} \) and put them into the database.
3. Use best samples in the database to initialize population and denote the best one as \( x_p \);
4. Set global_search_flag = true.
5. while the limit of function evaluations is not exceeded do
   1. if global_search_flag == true then
      1. Build the global RBF surrogate model \( f_{\text{global}} \) with all samples in the database.
      2. Perform TLBO algorithm (1) and (2) by \( f_{\text{global}} \)-estimated fitness.
      3. if \( \text{rand} < Jr \) then
         1. Perform OBL generation with the current population and select best individuals from them by \( f_{\text{global}} \)-estimated fitness.
         2. Select best individual \( x_p \) from the population by \( f_{\text{global}} \)-estimated fitness, evaluate it with \( f_{\text{real}} \) and put \( x_p \) into the database.
         3. if \( f_{\text{real}}(x_p) \geq f_{\text{real}}(x_b) \) then
            1. reverse global_search_flag.
        else
            1. \( x_b = x_p \)
      else
         1. Build the local RBF surrogate model \( f_{\text{local}} \) with N best samples in the database.
         2. Perform Nelder–Mead simplex search starting from \( x_b \) by \( f_{\text{local}} \)-estimated fitness and denote the result as \( x_s \);
       1. reverse global_search_flag.
   else
      1. Build the local RBF surrogate model \( f_{\text{local}} \) with N best samples in the database.
      2. Perform Nelder–Mead simplex search starting from \( x_b \) by \( f_{\text{local}} \)-estimated fitness and denote the result as \( x_s \);
6. end while

---
Evaluate \( x_s \) with \( f_{\text{real}} \) and put it into the database.

\[
\text{if } f_{\text{real}}(x_s) \geq f_{\text{real}}(x_b) \text{ then}
\]
\[
\text{reverse global_search_flag.}
\]
\[
\text{else}
\]
\[
x_b = x_s
\]
\[
\text{end}
\]

4. Numerical Results

In this section, we adopt six widely used benchmark functions [2] to verify the effectiveness of our proposed method OBL-SAHO and empirically compare it with other state-of-the-art SAEAs. The basic information of the problems is given in Table 1. In our experiments, the maximum number of real fitness evaluations (FEs) for each run is set to 1000. For each algorithm, 30 independent runs are carried out on each function and the statistical results are given. The problems have two kinds of dimensions (denoted as D) 50, 100 considered in the experiments below. The jumping rate \( J_r \) in Algorithm 1 is set to 0.05 according to [9]. The population size is set to \( 100 + \frac{D}{10} \) according to [4]. The size of initial samples is set to the population size and the size \( N \) of samples which local surrogate builds on is set to \( D \). The bias of optimum is removed in the following results. The best values on each line of the following tables are highlighted in bold.

Table 1. Properties of benchmark functions

| Function                  | Optimum | Property          |
|---------------------------|---------|-------------------|
| Ellipsoid                 | 0       | Unimodal          |
| Rosenbrock                | 0       | Multimodal        |
| Ackley                    | 0       | Multimodal        |
| Griewank                  | 0       | Multimodal        |
| Shifted Rotated           | -330    | Very complicated multimodal |
| Rastrigin (F10 in [22]) (SRR) | 10 | Very complicated multimodal |
| Rotated hybrid Composition| 10      | Very complicated multimodal |

4.1. Influence of the choice of OBL variants

The results of comparison between our proposed SAHO enhanced by two OBL variants DOL [12] and OML [9] and the original OBL [8] are shown in the table 2.

OML-SAHO achieves best results on all high-dimensional problems except SRR. This guarantees the advantage of OML when integrated into SAHO to solve HEOPs. As shown in the table 2, the performance of OML-SAHO on SRR is slightly poor but still acceptable which is due to the weakness of OML reported in [9].

Table 2. Comparisons of the statistical results (Mean/Std.) of OBL-SAHO, DOL-SAHO and OML-SAHO on the benchmark functions with 50D and 100D

| Problems | Functions | D | OBL-SAHO | DOL-SAHO | OML-SAHO |
|----------|-----------|---|----------|----------|----------|
|          |           | 50|          |          |          |
|          |           | 100|         |          |          |
### 4.2. Comparison with state-of-the-art algorithms

We further compare the proposed OML-SAHO with state-of-the-art SAEAs including ESAO [2], SA-COSO [13], SHPSO [14] and RPHSA [15] (no results of 50D functions reported for RPHSA) on 50D and 100D problems. To ensure the fairness of the comparison, the results of other SAEAs are all retrieved from their paper with recommended parameter settings used in our SAEA as well.

From the table, we can see OML-SAHO outperforms the other state-of-the-art SAEAs on Rosenbrock, Ackley and RHC among all six benchmark functions. It also achieves competitive results on Griewank while obviously worse on Ellipsoid and SRR. More accurate surrogate models could result in more stable and stronger surrogate-based OBL.

#### Table 3. Comparisons of the statistical results (Mean/Std) of OML-SAHO, ESAO, SA-COSO, SHPSO and RPHSA (no results of 50D functions reported for RPHSA) on benchmark functions with 50D and 100D

| Problems | Algorithms | Functions | OML-SAHO | ESAO | SA-COSO | SHPSO | RPHSA |
|----------|------------|-----------|----------|------|---------|-------|-------|
| Ellipsoid|            | 50        | Mean 7.5388e+01 | 4.0281e+00 | 4.0281e+00 | 4.0281e+00 | 4.0281e+00 |
|          |            | Std. 1.1405e+04 | 2.0599e+00 | 2.0599e+00 | 2.0599e+00 | 2.0599e+00 | 2.0599e+00 |
| Rosenbrock|            | 50        | Mean 4.9375e+01 | 5.0800e+01 | 5.0800e+01 | 5.0800e+01 | 5.0800e+01 |
|          |            | Std. 1.0028e+02 | 1.2014e+02 | 1.2014e+02 | 1.2014e+02 | 1.2014e+02 | 1.2014e+02 |
| Ackley   |            | 50        | Mean 4.8645e+02 | 6.2324e+02 | 6.2324e+02 | 6.2324e+02 | 6.2324e+02 |
|          |            | Std. 8.4176e+04 | 1.5113e+04 | 1.5113e+04 | 1.5113e+04 | 1.5113e+04 | 1.5113e+04 |
| Griewank |            | 50        | Mean 1.3415e+03 | 9.0025e+02 | 9.0025e+02 | 9.0025e+02 | 9.0025e+02 |
|          |            | Std. 2.0575e+04 | 1.4606e+03 | 1.4606e+03 | 1.4606e+03 | 1.4606e+03 | 1.4606e+03 |
| SRR      |            | 50        | Mean 1.3415e+03 | 9.0025e+02 | 9.0025e+02 | 9.0025e+02 | 9.0025e+02 |
|          |            | Std. 2.0575e+04 | 1.4606e+03 | 1.4606e+03 | 1.4606e+03 | 1.4606e+03 | 1.4606e+03 |
| RHC      |            | 50        | Mean 1.3415e+03 | 9.0025e+02 | 9.0025e+02 | 9.0025e+02 | 9.0025e+02 |
|          |            | Std. 2.0575e+04 | 1.4606e+03 | 1.4606e+03 | 1.4606e+03 | 1.4606e+03 | 1.4606e+03 |
5. Conclusion

In this paper, a surrogate-assisted hybrid optimization algorithm OBL-SAHO is proposed to solve HEOPs. OBL is a successful learning strategy to enhance the search capability and works with SAEA effectively. OBL-SAHO integrates TLBO-SM with the SAEA framework ESAO and strengthens its exploration capability with an OBL variant. Thus, a strong OBL variant OML helps OML-SAHO to outperform state-of-the-art SAEs on most of the benchmark functions which is reported in experimental results. However, the performance of OBL-SAHO could be limited by inaccuracy of surrogates. In the future work, we will attempt to investigate the effectiveness and efficiency of variants of opposition-based learning adopted with more accurate surrogate models.

References

[1] Dyn N, Levin D and Rippa S 1986 SIAM Journal on Scientific and Statistical Computing 7
[2] Martin J and Simpson T 2005 Aiaa Journal - AIAA J 43 853–863
[3] Venkata Rao R, Savsani V and Vakharia D 2011 Computer-Aided Design 43 303–315
[4] Pan J S, Liu N, Chu S C and Lai T 2020 Information Sciences 561
[5] Dong H, Wang P, Fu C and Song B 2020 Information Sciences 556
[6] Zhang H, Li B, Zhang J, Qin Y, Feng X and Liu B 2015 Soft Computing 20
[7] Lagarias J, Reeds J, Wright M and Wright P 1998 SIAM Journal on Optimization 9 112–147
[8] Tizhoosh H 2005 Opposition-based learning: A new scheme for machine intelligence vol 1 pp 695 – 701 ISBN 0-7695-2504-0
[9] Xu Y, Xiaofeng Y, Yang Z, Li X, Wang P, Ding R and Liu W 2021 Neurocomputing 435
[10] Cao B, Su Y and Fan S 2019 A surrogate-assisted improved many-objective evolutionary algorithm Advances in Swarm Intelligence ed Tan Y, Shi Y and Niu B (Cham: Springer International Publishing) pp 69–78 ISBN 978-3-030-26354-6
[11] Suganthan P, Hansen N, Liang J, Deb K, Chen Y p, Auger A and Tiwari S 2005 Natural Computing 341-357
[12] Xu Y, Yang Z, Li X, Kang H and Xiaofeng Y 2019 Knowledge-Based Systems 188 104966
[13] Sun C, Jin Y, Cheng R, Ding J and Zeng J C 2017 IEEE Transactions on Evolutionary Computation PP 1–1
[14] Yu H, Tan Y, Zeng J C, Sun C and Jin Y 2018 Information Sciences 454-455