



Algorithm Portfolio for Individual-based Surrogate-Assisted Evolutionary Algorithms

Hao Tong, Jialin Liu, XinYao*

Shenzhen Key Laboratory of Computational Intelligence
University Key Laboratory of Evolving Intelligent Systems of Guangdong Province
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China
htong6@outlook.com, {liujl, xinyao}@sustech.edu.cn

ABSTRACT

Surrogate-assisted evolutionary algorithms (SAEAs) are powerful optimisation tools for computationally expensive problems (CEPs). However, a randomly selected algorithm may fail in solving unknown problems due to no free lunch theorems, and it will cause more computational resource if we re-run the algorithm or try other algorithms to get a much solution, which is more serious in CEPs. In this paper, we consider an algorithm portfolio for SAEAs to reduce the risk of choosing an inappropriate algorithm for CEPs. We propose two portfolio frameworks for very expensive problems in which the maximal number of fitness evaluations is only 5 times of the problem's dimension. One framework named Par-IBSAEA runs all algorithm candidates in parallel and a more sophisticated framework named UCB-IBSAEA employs the Upper Confidence Bound (UCB) policy from reinforcement learning to help select the most appropriate algorithm at each iteration. An effective reward definition is proposed for the UCB policy. We consider three state-of-the-art individual-based SAEAs on different problems and compare them to the portfolios built from their instances on several benchmark problems given limited computation budgets. Our experimental studies demonstrate that our proposed portfolio frameworks significantly outperform any single algorithm on the set of benchmark problems.

KEYWORDS

Algorithm portfolio, surrogate model, evolutionary algorithm, computationally expensive problem

ACM Reference Format:

Hao Tong, Jialin Liu, XinYao. 2019. Algorithm Portfolio for Individual-based Surrogate-Assisted Evolutionary Algorithms. In *Genetic and Evolutionary Computation Conference (GECCO '19)*, July 13–17, 2019, Prague, Czech Republic. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3321707.3321715>

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '19, July 13–17, 2019, Prague, Czech Republic

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6111-8/19/07...\$15.00

<https://doi.org/10.1145/3321707.3321715>

1 INTRODUCTION

Computationally expensive problems (CEPs) are very common in many real-world systems, requiring enormous computational resources to accomplish one fitness evaluation [13]. For instance, one evaluation based on computational fluid dynamic simulations will cost several hours [12]. Obviously, canonical evolutionary algorithms are challenging to handle this kind of problems directly. To overcome this challenge, surrogate-assisted evolutionary algorithms (SAEAs) are developed by applying a much cheaper model to replace the actual expensive fitness evaluation process to reduce the computational cost [11].

Over the past decades, many efficient SAEAs has been proposed and applied into complex real-world applications, such as trauma system [23]. Individual-based model control [10] method is the most effective strategy that a few individuals will be re-evaluated by the actual function in each generation according to different criteria. For example, some criteria like expected improvements (EI), consider the fidelity of surrogate models and the quality of evaluated solutions simultaneously in global optimisation [14]. On the other hand, recent works in individual-based SAEAs proposed new strategies to trade-off exploration and exploitation of optimisation, like active learning based model management [22] and Voronoi-based SAEA framework for very expensive problems [20]. They obtain a better solution in test problems and real-world applications compared with classical efficient global optimisation (EGO) [14].

Even though many model management strategies in individual-based SAEAs are successful in the literature, no free lunch theorems indicate that there is no one best approach appropriate for every problem [24]. For example, EGO is much more powerful than state-of-the-art algorithms in low-dimension cases and algorithm in [26] is an expert in multi-modal expensive problems while Voronoi based SAEA framework is good at uni-modal problems [20]. However, it is hard to determine the optimal algorithm for an unknown problem in practice. In order to address this challenge, algorithm portfolio is employed to reduce the risk of failing to optimise problems in multiple scenarios[9]. For example, the algorithm portfolio obtained an excellent performance in SAT problems [25] and noisy optimization problem [4, 5]. Although many portfolio strategies have been proposed for noise free evolutionary algorithms [3, 16, 21, 27] and Bayesian optimisation [8], to the best of our knowledge, we are the first to apply the algorithm portfolio for individual-based SAEAs which has shown outstanding performance in solving CEPs.

We proposed two algorithm portfolio frameworks in this paper for individual-based SAEAs in very expensive problems [20]. The first framework is motivated from the population-based algorithm portfolio [16], which runs all algorithm candidates simultaneously. In another framework, we employ the technique from reinforcement learning to select relatively “best” algorithm for every generation. Unlike the portfolio for Bayesian optimisation, we directly choose a method to search a solution for re-evaluation instead of generating several solutions simultaneously by different approaches and then evaluating one of them by actual fitness function [17].

The remainder of this paper is structured as follows. Section 2 will present some related work about algorithm portfolio. And then the detail of two algorithm portfolio frameworks will be introduced in Section 3. In Section 4, we will apply some state-of-the-art individual-based SAEAs to the proposed frameworks and test them in a series of benchmark problems. Finally, the paper will end with a brief conclusion and a discussion of future work in Section 5.

2 RELATED WORK

2.1 Portfolio of evolutionary algorithm

In the areas of evolutionary algorithms, algorithm portfolio is applied to increase the probability of finding a better solution by allocating computational resources to several complementary algorithms. The algorithm portfolio frameworks in the literature can be classified into two categories as the parallel-based framework and the sequential-based framework.

For the parallel-based framework, all candidates will run simultaneously in multiple sub-processes. Population-based algorithm portfolio (PAP) is a typical example [16], which allocates computational resources before the optimization according to the prior knowledge. Each algorithm has its own population and evolve independently, but the information is shared among different algorithms by migration strategy. Besides, other parallel-based portfolio frameworks like AMALGAM-SO [21] and the UMOEAs [6] collect the performance of algorithms during the optimisation process and allocate more resources to the better algorithm.

On the other hand, the sequential based framework only runs one algorithm at most of the time during the process of optimisation. Different from the parallel-based algorithm portfolio, this kind of framework try to select the best algorithm in different optimisation stage. The multiple evolutionary algorithm (MultiEA) is one of the state-of-the-art sequential algorithm portfolio frameworks [27]. It utilises the history convergence curve of each algorithm to predict its performance in the near future, and then the best algorithm will be selected to optimise the problem.

Another typical sequential portfolio strategy worthy of mention is an online racing algorithm, max-race portfolio (MRP) [28]. The best algorithm is selected by a statistical test on algorithms’ online performance and when enough statistical evidence indicates that one algorithm is significantly inferior to other algorithms, the worst one will be removed by framework permanently.

2.2 Multi-armed bandit problem

The sequential portfolio framework aims at selecting the best optimisation algorithm in the next generation according to the previous

performance. The computational resources are dynamically allocated to algorithm constituents. It is similar to the multi-armed bandit framework in reinforcement learning, taking actions to maximise the cumulative reward within limited cost [1, 15].

For a K -armed bandit problem, it is basically defined by random variables $\{X_{i,t} | i = 1, 2, \dots, K, t \in \mathbb{N}\}$ where each $X_{i,t}$ represents an independent and identical distribution with an unknown expectation μ_i for each arm of bandit machine in t_{th} successful pull [1]. For any environment state, the action being taken at next time step is determined by a bandit policy π , which is learned according to the actions’ history rewards. The quality of a policy is measured by cumulative regret, which could be defined by Eq. (1):

$$R_n = \mu^* n - \sum_{j=1}^n E[T_j(n)\mu_j] \quad (1)$$

where μ_j is the expected reward of arm j , μ^* is the expectation reward of optimal arm, i.e. $\mu^* \stackrel{\text{def}}{=} \max_{1 \leq j \leq K} \mu_j$ and $T_j(n)$ represents the number of times arm j has been pulled over n trials.

The upper confidence bound (UCB) algorithm is a prevalent and effective method for multi-armed bandit problems to tackle the dilemma between exploitation and exploration [1]. In this paper, the UCB-Tuned (UCB-t) algorithm is applied for algorithm portfolio because there is no additional parameter requiring to be adjusted in the algorithm which is presented in Eq. (2):

$$\pi_{j,n} = \bar{\mu}_j + \sqrt{\frac{\ln n}{T_j(n)}} \cdot \min\left\{\frac{1}{4}, v_j(T_j(n))\right\} \quad (2)$$

and

$$v_j(s) = \frac{1}{s} \sum_{\tau=1}^s \mu_{j,\tau}^2 - \bar{\mu}_{j,s}^2 + \sqrt{\frac{2 \ln n}{s}} \quad (3)$$

where $\bar{\mu}_j$ is the average reward of arm j after n trials. The policy will select the arm with maximal UCB value according to Eq. (2) and (3) for the next generation.

In the literature, there has been some works about bandit framework for algorithm selection. Baudiš and Pošík [3] applied basic UCB in black box optimisation in which they defined the reward by introducing a log-rescaling method to process the raw fitness value. And the value rank [7] as a method of reward definition is also compared in the experiments. The results show that the UCB algorithm is efficient in algorithm selection problems. Also in [18], authors regarded the algorithm selection problem as a non-stationary bandit problem and applied UCB algorithm to be the decision policy.

From this view, it is reasonable to consider the algorithm portfolio problem in the area of reinforcement learning and employ appropriate methods to construct the framework for individual-based SAEAs.

3 ALGORITHM PORTFOLIO STRATEGIES

Individual-based SAEAs re-evaluate a few individuals at each generation and individuals being re-evaluated in the next generation is only determined by the current database. As a sequence, we will introduce two portfolio frameworks as parallel individual-based

SAEAs and UCB for individual-based SAEAs which are motivated from two different aspects as reviewed previously.

3.1 Parallel individual-based SAEAs

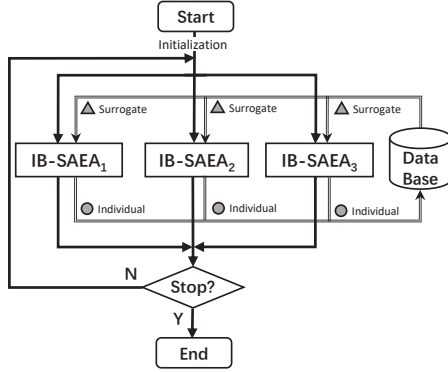


Figure 1: The diagram of the framework: Parallel individual-based SAEAs.

Similar with the algorithm portfolio for canonical evolutionary algorithms, it is intuitive to consider each individual-based SAEA as a simple evolutionary algorithm and embed them into the existing framework, like PAP or MultiEA. From this aspect, the parallel individual-based SAEAs (Par-IBSAEA) framework is proposed that all algorithm candidates run simultaneously at each generation. Nevertheless, it is more convenient than portfolio for canonical evolutionary algorithms because almost all individual-based SAEAs have the same algorithm structure and it does not require a particular design for each algorithm. A brief diagram for Par-IBSAEA with only three algorithm instances is presented in Figure 1 where the double solid line represents the interaction between individual-based SAEAs and the database.

Considering the set of individual-based SAEAs $\mathcal{A} = \{A_1, A_2, \dots, A_n\}$ with n algorithms, each algorithm is independent with each other. The framework starts by the initialization of a database and all SAEAs use the same surrogate model constructed by samples in the database to search for the next re-evaluated solution. Due to different mechanisms for various algorithms, they will obtain several different solutions for re-evaluation as $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$, where the size of k is not completely equal to n because some individual-based SAEAs might get more than one solution. For example, VESAEA will re-evaluate two solutions in one generation when the optimisation in global search stage [20]. The database will be updated by adding evaluated solutions at the end of one iteration, and the whole algorithm will be stopped when the fitness evaluations are exhausted.

3.2 UCB for individual-based SAEAs

On the other hand, the individual-based SAEA portfolio is regarded as a multi-armed bandit problem that an individual-based SAEA is considered as an arm and the quality of solutions could be used to

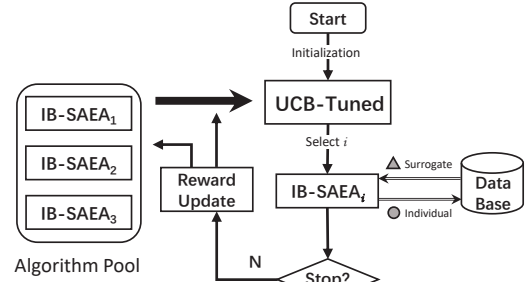


Figure 2: The diagram of the framework: UCB for individual-based SAEAs.

measure the reward of actions. Then, we could apply the UCB algorithm to determine which algorithm is used in the next generation for CEPs.

The key point in UCB for individual-based SAEAs (UCB-IBSAEA) is the definition of reward. According to the condition of UCB, the reward used in UCB policy must be in the range of $[0, 1]$. It is intuitive to define the reward of an algorithm by linear scaling the solution's fitness into the bound of problem. However, we can not obtain the actual bounds due to the problem solved is a black box. And the only information we could use is the evaluated individuals in the database. Therefore, the bounds could be estimated by the samples' fitness in the database. Assuming all solutions' fitness in database in t_{th} generation is $\mathcal{Y}_t = \{y_1, y_2, \dots, y_n\}$, the empirical upper bound eUB_t and empirical lower bound eLB_t could be estimated by Eq. (4):

$$\begin{cases} eUB_t = \max(\mathcal{Y}_t) \\ eLB_t = \min(\mathcal{Y}_t) \end{cases} \quad (4)$$

and the empirical upper bound and lower bound will be updated after updating the database. As a sequence, the reward of one algorithm could be formulated as Eq. (5):

$$\mu_{j,t} = \frac{eUB_t - f_{i,t}}{eUB_t - eLB_t} \quad (5)$$

where $f_{i,t}$ is the actual fitness of solution found by the algorithm in t_{th} iteration.

However, the performance of empirical bound estimated by Eq. (4) will be bad due to the characteristic of the optimisation process. It is obvious that the convergence speed varies with the optimisation stage that the speed is much faster at the beginning and slows down when the optimization is near convergence. Therefore, the empirical bounds are almost fixed and evaluated solutions' fitness are almost all very close to the empirical lower bound in the later stage of optimisation. As a sequence, the reward between different algorithms almost has no significant difference and the UCB policy is not effective any more. Hence, the simple normalization of fitness to be the reward is not appropriate for the UCB policy.

In order to improve the stability of the UCB policy, we introduce an online strategy to update the estimated bound by sliding window.

The main idea is shown in Algorithm 1. Considering all evaluated solutions in the database after t_{th} generation, a sliding window with the size of sw selects best sw solutions to form a subset. The sw is set as $2d$ in this work, where d is the dimension of a problem. If the minimal fitness in the subset is lower than the current eLB which indicates the problem's lower bound could be a much lower value, the eLB will be updated by the newest minimal value. And if the maximal fitness in the subset is lower than current eUB which means the optimisation has entered into another stage compared to the last stage, eUB will be updated to refine the empirical bounds.

Algorithm 1: Update empirical bounds by sliding window.

Input: Database, lower bound eLB_t , upper bound eUB_t

- 1 Size of sliding window: sw
- 2 Sort evaluated samples in database from best to worst;
- 3 The top sw best individuals $\mathcal{S} = \{(x_1, y_1), \dots, (x_{sw}, y_{sw})\}$;
- 4 Collect fitness of individuals in \mathcal{S} : $\mathcal{Y} = [y_1, y_2, \dots, y_{sw}]$;
- 5 **if** $eLB_t > \min(\mathcal{Y})$ **then**
- 6 $eLB_{t+1} = \min(\mathcal{Y})$
- 7 **else**
- 8 $eLB_{t+1} = eLB_t$
- 9 **end**
- 10 **if** $eUB_t > \max(\mathcal{Y})$ **then**
- 11 $eUB_{t+1} = \max(\mathcal{Y})$
- 12 **else**
- 13 $eUB_{t+1} = eUB_t$
- 14 **end**

Output: New empirical bound: eLB_{t+1}, eUB_{t+1}

After defining the algorithm's reward, the UCB-IBSAEA framework is easy to be implemented which is presented in the Figure 2. The UCB-t with no additional parameter in Eq. (2) is used in the framework. The framework starts by initialising a database and then the UCB-IBSAEA will select one best algorithm A_k from the algorithm pool $\mathcal{A} = \{A_1, A_2, \dots, A_n\}$ according to the UCB-t policy for the following generation. The new solution will be added into the database after being evaluated by the actual fitness function. After that, the empirical bounds and reward information for each algorithm will be updated. Finally, the optimisation process will stop after running out of all fitness evaluations.

4 NUMERICAL EXPERIMENTS AND ANALYSIS

The performance of two proposed portfolio frameworks is studied in this section. We choose three state-of-the-art individual-based SAEAs to generate two portfolio instances and evaluate them on a set of benchmark functions.

4.1 Experimental setting

In this work, very computationally expensive problems are considered so that we employ three efficient individual-based SAEAs to assess the efficacy of portfolio frameworks. Three algorithms are expert in different kinds of problems, and a brief introduction for them is presented below:

- EGO-LCB: Efficient global optimisation with lower confidence bound [14] is a very efficient algorithm for low-dimension expensive problems.
- VESAEA: Voronoi-based efficient SAEA employed Voronoi diagram in SAEA framework to assist the local search process [20]. It is good at uni-modal problems even though there are some noises over the landscape.
- GORS-SSLPSO: Despite it is a so-called generation based SAEA [26], we consider it as the individual based SAEA because the solution re-evaluated in next generation is determined by the current database. The algorithm maintains the diversity by PSO operator and guarantees the convergence by a restart strategy. And it is appropriate for solving multi-modal problems.

Table 1: Benchmark Problems. The optimum solution is shifted to another random position in the landscape.

Problem	Dimension	Optimum	Note
Sphere	10,20,30	0	Uni-modal
Griewank	10,20,30	0	Multi-modal
Ackley	10,20,30	0	Multi-modal
Rosenbrock	10,20,30	0	Multi-modal with narrow valley
Rastrigin	10,20,30	0	Very complicated multi-modal

Frameworks are tested on five widely used problems with dimensions $d = 10, 20, 30$ as shown in Table 1. The maximal fitness evaluation is setting as $5d$ where $2d$ is used for initialisation, and all comparisons are based on 25 independent runs. In the following, the comparison between portfolio frameworks and three single SAEAs will be performed firstly and then we will analyse the performance of portfolio frameworks by comparing with another two frameworks: random selection framework and epsilon-greedy selection framework [2].

4.2 Comparative results with single algorithms

The results of two proposed frameworks and three algorithm candidates on benchmark problems over 25 independent runs are presented in Table 2 where figures in each cell denote averaged best fitness and standard deviation, in which bold ones are the best results among five algorithms in one problem. The averaged ranking for algorithm portfolio and three SAEAs are listed on the third row from the bottom, from which we could find the proposed frameworks are much better than all single algorithms. Moreover, the last two rows provide the result of Wilcoxon test with a 0.05 significance level, in which the UCB-IBSAEA and Par-IBSAEA are control methods, respectively and the 'win-draw-lose' represents the control method is superior, not significantly different and inferior to the compared algorithm. Convergence profiles of all algorithms on 15 test problems are plotted in Figs. 3-5, where the x-axis ranges from $2D$ to $5D$ because the first $2D$ fitness evaluations are used for initialization, which is same for all algorithms in each run for fair comparison.

The overall performance of an algorithm portfolio is significantly better than each single algorithm in Table 2. However, the performance of an algorithm portfolio in different problems has a significant difference. For example, in sphere and griewank problems, the portfolio dramatically improves the quality of the final

Table 2: The averaged best results and standard deviation of algorithm portfolio and single algorithm on test problems with 25 independent runs. The ‘win-draw-lose’ represents the control method is superior, not significantly different and inferior to the compared algorithm.

Problem	D	UCB-IBSAEA	Par-IBSAEA	GORS-SSLPSO	VESAEA	EGO-LCB
sphere	10	1421.21 ± 939.89	1039.95 ± 712.93	5483.17 ± 2789.23	1878.56 ± 691.56	5728.52 ± 2791.72
sphere	20	2445.62 ± 1109.81	2417.95 ± 938.53	9604.73 ± 2911.24	10906.42 ± 1487.33	44083.71 ± 7045.51
sphere	30	3606.77 ± 1377.47	3091.30 ± 1171.53	15133.57 ± 4323.70	25736.52 ± 2554.90	81185.30 ± 8810.83
rosenbrock	10	209.05 ± 82.43	234.90 ± 112.40	89.03 ± 46.93	238.17 ± 101.78	1127.73 ± 399.72
rosenbrock	20	349.14 ± 105.95	512.16 ± 144.50	281.95 ± 314.65	936.38 ± 240.64	5281.28 ± 1460.20
rosenbrock	30	418.05 ± 86.88	596.34 ± 150.43	311.92 ± 116.62	2419.77 ± 435.33	14763.02 ± 2247.72
ackley	10	19.46 ± 0.69	19.11 ± 1.39	19.52 ± 0.85	16.61 ± 1.89	15.71 ± 5.35
ackley	20	19.18 ± 0.58	19.27 ± 0.70	19.49 ± 0.51	18.43 ± 0.74	20.48 ± 0.23
ackley	30	19.06 ± 0.59	19.22 ± 0.50	19.52 ± 0.59	19.13 ± 0.37	20.75 ± 0.18
griewank	10	15.18 ± 12.21	12.82 ± 9.30	78.07 ± 37.73	17.83 ± 5.11	46.91 ± 20.39
griewank	20	27.34 ± 12.03	18.84 ± 5.62	143.75 ± 41.18	101.19 ± 11.65	386.30 ± 58.78
griewank	30	39.46 ± 13.21	31.33 ± 12.98	209.52 ± 48.56	226.61 ± 15.51	714.12 ± 92.21
rastrigin	10	73.63 ± 16.69	79.21 ± 18.00	55.79 ± 24.19	81.52 ± 14.91	101.07 ± 20.36
rastrigin	20	117.38 ± 30.23	152.84 ± 34.60	86.66 ± 31.20	165.87 ± 34.84	271.99 ± 19.77
rastrigin	30	143.47 ± 29.35	176.62 ± 26.82	114.12 ± 25.37	235.87 ± 53.24	467.43 ± 28.99
Average-Ranking		2.07	2.20	2.73	3.33	4.67
Wilcoxon-Test		Control method	4-9-2	7-2-6	10-3-2	14-1-0
Wilcoxon-Test		2-9-4	Control method	6-3-6	9-4-2	14-1-0

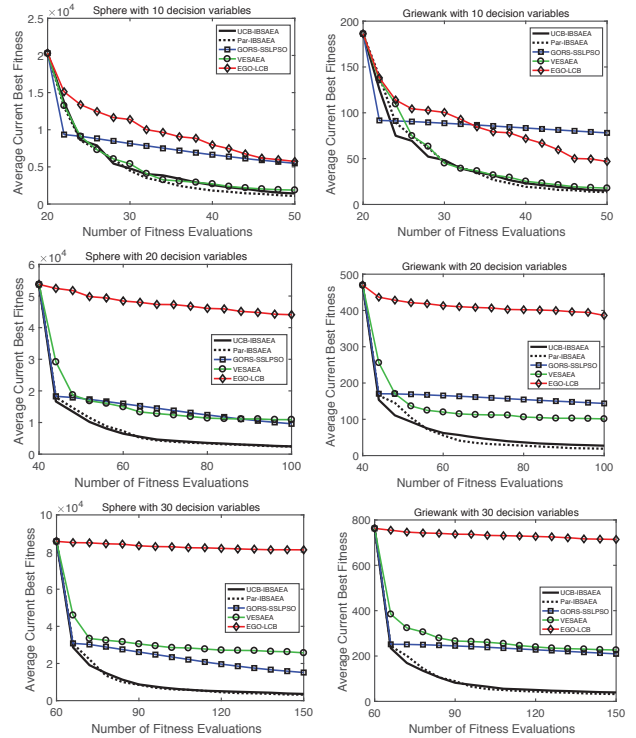
solution. By contrast, they are not always the best in rosenbrock and rastrigin problems, but it still can get an acceptable result compared with the best single algorithm.

The sphere function is a uni-modal problem and griewank function can also be regarded as a uni-modal problem with many small noises in the whole landscape. It is obvious in Table 2 that the portfolio performs much better than every single algorithm on these kind of problems. Nonetheless, we could find the portfolio’s effect is much higher when the dimension equal to 20 and 30 even though there is a significant terrible algorithm in the framework as shown in Figure 3. In 10-dimension case, the portfolio is similar to the performance of VESAEA which is the best algorithm for this kind of problem among three candidates.

Rosenbrock and rastrigin problems are both multi-modal problems which the performance of optimisation algorithms is likely to be hindered by attractive local optimums. GORS-SSLPSO is the most appropriate optimization algorithm for multi-modal problems among three algorithm candidates. And two algorithm portfolio frameworks are a little worse than GORS-SSLPSO, but performs better than other two algorithms as shown in convergence profiles in Figure 4.

In Ackley problems, although the superiority of algorithm portfolio is not obvious, portfolio still obtains relatively better result compared with the worst single algorithm. It is probably due to the little difference between three algorithms for this problem, and the performance of portfolio is restricted by the single algorithm’s ability.

As introduced above, we could obtain the conclusion that two proposed frameworks could improve the performance of single algorithms in uni-modal problems and obtain the similar performance with the most appropriate algorithm candidate in complex multi-modal problems. For uni-modal problems, there is only one attractive global optimal solution over the whole landscape and the

**Figure 3: Convergence curves of comparison algorithms on the Sphere and Griewank function with $D = 10, 20, 30$.**

algorithm portfolio performs better than every single algorithm probably because various algorithms provide more diversity during the optimization process. VESAEA and EGO-LCB are good at exploiting the search space and GORS-SSLPSO could contribute

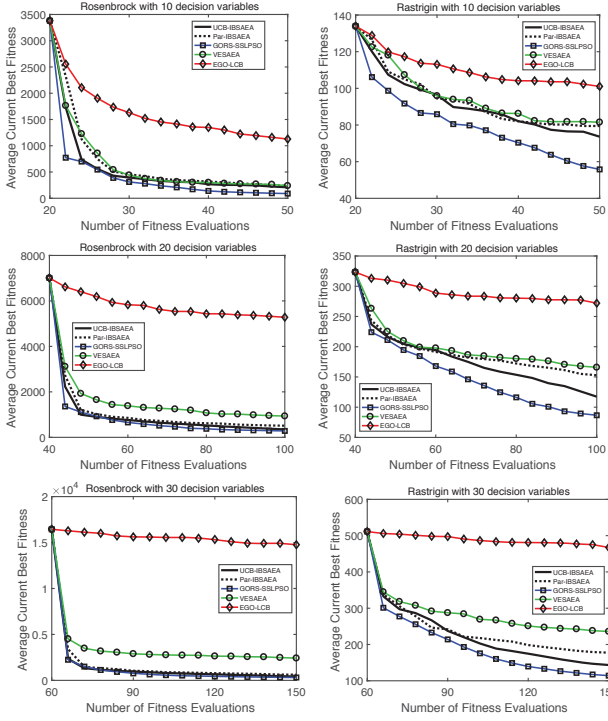


Figure 4: Convergence curves of comparison algorithms on the Rosenbrock and Rastrigin function with $D = 10, 20, 30$.

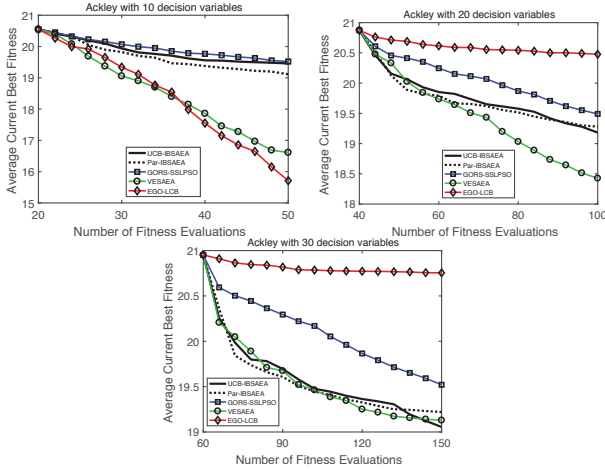


Figure 5: Convergence curves of comparison algorithms on the Ackley function with $D = 10, 20, 30$.

more diversity so that they promote the performance of each other in uni-modal problems. But in the multi-modal problem which has many attractive local optimums, although the GORS-SSLPSO can provide much diversity, other two algorithms are easy to be trapped in the local optimal. As a result, the whole performance of the portfolio is hindered by VESAEA and EGO-LCB so that the result of the portfolio is slightly worse than the GORS-SSLPSO algorithm.

4.3 Comparative results with other frameworks

To further analyse the efficacy of two proposed portfolio frameworks, we compare their performance with other two frameworks: random selection (RS) policy and epsilon-greedy policy with $\epsilon = 0.5$ (EG50). RS policy randomly selects an algorithm at each generation while EG50 policy selects the “optimal” algorithm with probability 0.5 according to the cumulative reward and selects a random algorithm with probability 0.5 [2]. The comparative results on test problems are presented in Table 3.

As shown in the result table, either UCB-IBSAEA or Par-IBSAEA obtain the best result among four frameworks except in 10-dimension ackley problem that RS gets a relatively better solution. Furtherly, the Par-IBSAEA performs better in sphere and griewank problem which are regarded as uni-modal problems. And UCB-IBSAEA is always the best framework among four frameworks in rosenbrock and rastrigin problems which are complex multi-modal problems. And the four frameworks also have the similar performance in Ackley function.

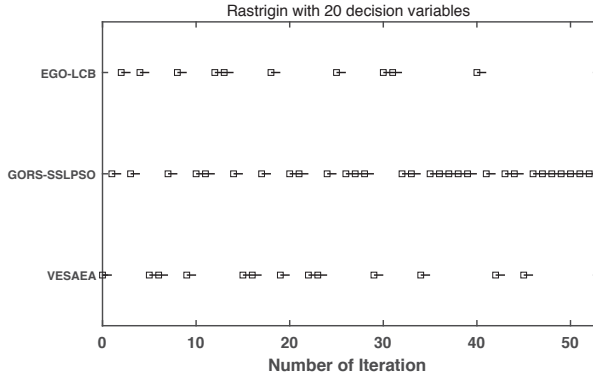
Compared with UCB-IBSAEA, RS, EG50, Par-IBSAEA is the only parallel framework that running algorithm candidate in parallel. It is better than other three sequential frameworks in sphere and griewank problems. As discussed in the above subsection, algorithm portfolio increases the diversity during the optimization process by combining different kinds of algorithms. Par-IBSAEA uses all algorithm simultaneously at each generation, using the same database that algorithm candidates search the next re-evaluated solution from various aspects. This idea is similar to the negative correlation search (NCS) [19], which uses different agents to cover different regions of search space. However, if the number of algorithm candidates increases, the performance of Par-IBSAEA might deteriorate because the limited computational cost cannot afford the huge cost for parallelly running many algorithms simultaneously at each iteration.

Although Par-IBSAEA is outstanding in uni-modal problems, UCB-IBSAEA is superior to other frameworks in multi-modal problems as shown in Table 3. It selects the appropriate algorithm at each generation according to the UCB policy in Eq. (2). The GORS-SSLPSO is the best algorithm among three algorithm candidates and UCB-IBSAEA is likely to detect the fact and allocate more computational budget to this algorithm. Take the 20-dimension rastrigin problem as an example, the behaviour of selection in each iteration of UCB-IBSAEA is plotted in Figure 6. We could find UCB-IBSAEA explores three algorithms at the early stage and then detect that GORS-SSLPSO is better than another two algorithms so that it mostly selects GORS-SSLPSO at the later stage.

On the other hand, we find that there is no big difference between random selection policy and two proposed frameworks in Table 3. This is mainly because there are only three algorithms in our experiments that the best algorithm will be selected at a high probability. Furtherly, the randomness could provide diversity for portfolio framework and avoid framework being trapped in one action so that RS policy might obtain the best result than other frameworks, like 10-dimension ackley problem. Meanwhile, the difference among algorithm candidates is not very large so that random selection will not perform very badly.

Table 3: The averaged best results and standard deviation of four different frameworks: UCB-IBSAEA, Par-IBSAEA, RS, EG50 on test problems with 25 independent runs. The ‘win-draw-lose’ represents the control method is superior, not significantly different and inferior to the compared algorithm.

Problem	D	UCB-IBSAEA	Par-IBSAEA	RS	EG50
sphere	10	1421.21 ± 939.89	1039.95 ± 712.93	1178.42 ± 567.00	1280.76 ± 494.39
sphere	20	2445.62 ± 1109.81	2417.95 ± 938.53	2598.02 ± 1144.57	4455.35 ± 1584.86
sphere	30	3606.77 ± 1377.47	3091.30 ± 1171.53	3134.88 ± 995.80	4980.92 ± 1781.29
rosenbrock	10	209.05 ± 82.43	234.90 ± 112.40	268.05 ± 143.15	264.17 ± 108.04
rosenbrock	20	349.14 ± 105.95	512.16 ± 144.50	467.78 ± 174.07	561.89 ± 183.74
rosenbrock	30	418.05 ± 86.88	596.34 ± 150.43	604.03 ± 134.11	721.90 ± 194.39
ackley	10	19.46 ± 0.69	19.11 ± 1.39	18.66 ± 1.10	18.82 ± 1.35
ackley	20	19.18 ± 0.58	19.27 ± 0.70	19.36 ± 0.57	19.36 ± 0.47
ackley	30	19.06 ± 0.59	19.22 ± 0.50	19.27 ± 0.45	19.17 ± 0.42
griewank	10	15.18 ± 12.21	12.82 ± 9.30	15.07 ± 9.69	20.56 ± 12.21
griewank	20	27.34 ± 12.03	18.84 ± 5.62	20.12 ± 7.59	37.70 ± 19.87
griewank	30	39.46 ± 13.21	31.33 ± 12.98	35.29 ± 9.84	50.00 ± 15.22
rastrigin	10	73.63 ± 16.69	79.21 ± 18.00	84.20 ± 17.47	79.99 ± 18.15
rastrigin	20	117.38 ± 30.23	152.84 ± 34.60	138.08 ± 23.11	155.38 ± 31.97
rastrigin	30	143.47 ± 29.35	176.62 ± 26.82	175.89 ± 26.73	191.12 ± 40.51
Average-Ranking		2.00	1.93	2.60	3.47
Wilcoxon-Test		Control method	4-9-2	5-8-2	9-6-0
Wilcoxon-Test		2-9-4	Control method	0-15-0	7-8-0

**Figure 6: An illustration of selection behaviour for UCB-IBSAEA in 20-dimension rastrigin problem.**

4.4 Risk analysis

The risk of proposed framework for individual-based SAEAs will be analyzed in this subsection. The metric to measure algorithms' risk is employed from the work in [16], in which the comparative risk is estimated by comparing the quality of solutions obtained by two algorithms in a set of test problems with several runs. Considering benchmark problems $\mathcal{F} = \{f_k | k = 1, 2, \dots, n\}$ and algorithm constituents $\mathcal{A} = \{A_j | j = 1, 2, \dots, m\}$, the probability of A_i outperforming A_j as $A_i > A_j$ can be calculated by the following equation:

$$P(A_i > A_j) = \frac{1}{n} \sum_{k=1}^n P(q_{i,k} < q_{j,k} | f_k) \quad (6)$$

where the $q_{i,k}$ denotes the quality of solution obtained by A_i in f_k . And $P(q_{i,k} < q_{j,k} | f_k)$ can be estimated by the following equation:

$$P(q_{i,k} < q_{j,k} | f_k) = \frac{\sum_{s=1}^{s_i} \sum_{t=1}^{s_j} \mathcal{I}(y_{i,k,s} < y_{j,k,t})}{s_i \times s_j} \quad (7)$$

where $y_{i,k,s}$ represents the fitness of solution obtained by A_i in s_{th} trial for f_k problem, s_i, s_j represent the number of trails of each algorithm for one problem and $\mathcal{I}(\cdot)$ denotes the indicator function.

Table 4: The comparison risk of algorithm portfolio and single algorithm. The two figures in each cell stand for the probabilities that the portfolio and the individual-based SAEAs outperformed each other.

	GORS-SSLPSO	VESAEA	EGO-LCB
UCB-IBSAEA	0.60 - 0.40	0.75 - 0.25	0.94 - 0.06
Par-IBSAEA	0.55 - 0.45	0.72 - 0.28	0.94 - 0.06
RS	0.56 - 0.44	0.71 - 0.29	0.94 - 0.06
EG50	0.55 - 0.45	0.68 - 0.32	0.94 - 0.06

The comparative risk between portfolio frameworks and three SAEAs are shown in the Table 4. The four portfolio frameworks are all better than single SAEAs, in which the UCB-IBSAEA is slightly better than other three frameworks. Firstly, we can conclude that the portfolio framework could obviously reduce the risk of failing in optimizing problems. The performance of SAEAs could be promoted by combining various efficient algorithms so that the CEPs could be solved much better within limited computational resources. Secondly, we can find the proposed UCB-IBSAEA is more effective than other three frameworks in terms of optimisation risk so that the UCB-IBSAEA is more appropriate to be used for an unknown CEP.

5 CONCLUSION

In this paper, the algorithm portfolio for individual-based SAEAs is the first time to be considered in solving computationally very expensive black-box problems. We creatively proposed two effective portfolio frameworks as the Par-IBSAEA and UCB-IBSAEA framework. The Par-IBSAEA runs all algorithms in parallel and all algorithm candidates share the same database at each iteration. On the other hand, the UCB-IBSAEA considers the portfolio as a multi-armed bandit problem and employs the UCB-Tuned algorithm to formulate the selection policy. Meanwhile, we design a new efficient reward definition for optimization in the UCB-IBSAEA framework, which normalizes the quality of solution in the scope of a sliding window. Additionally, the numerical experiments are performed to analyze the performance of the proposed frameworks in five commonly used test problems with dimension $d = 10, 20, 30$. The results shows that the Par-IBSAEA can increase the diversity of optimization process and gets a much better performance than all the algorithm constituents in uni-modal problems as well as several multi-model problems even though a significant terrible algorithm is included in the framework, and the UCB-IBSAEA can effectively balance the exploration and exploitation in algorithm selection, which obtains an excellent performance than other frameworks in multi-modal problems.

This work mainly focuses on individual-based SAEAs only for solving CEPs. In the future, the portfolio framework will be improved to suit for more general optimisation algorithms. On the other hand, it is much important to select appropriate algorithm candidates for algorithm portfolio which directly influence the framework's performance, so it is valuable to do more research on how to choose constituent algorithms for algorithm portfolio in SAEAs.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (Grant No. 2017YFC0804003), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2017ZT07X386), Shenzhen Peacock Plan (Grant No. KQTD2016112514355531), the Science and Technology Innovation Committee Foundation of Shenzhen (Grant No. ZDSYS201703031748284, Grant No. JCYJ20180504165652917 and Grant No. JCYJ20170817112421757) and the Program for University Key Laboratory of Guangdong Province (Grant No. 2017KSYS008).

REFERENCES

- [1] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning* 47, 2-3 (2002), 235–256.
- [2] Petr Baudis. 2014. COCOpf: An Algorithm Portfolio Framework. *CoRR* abs/1405.3487 (2014).
- [3] Petr Baudis and Petr Pošík. 2014. Online black-box algorithm portfolios for continuous optimization. In *International Conference on Parallel Problem Solving from Nature*. Springer, 40–49.
- [4] Marie-Liesse Cauwet, Jialin Liu, Baptiste Rozière, and Olivier Teytaud. 2016. Algorithm portfolios for noisy optimization. *Annals of Mathematics and Artificial Intelligence* 76, 1-2 (2016), 143–172.
- [5] Marie-Liesse Cauwet, Jialin Liu, and Olivier Teytaud. 2014. Algorithm portfolios for noisy optimization: Compare solvers early. In *International Conference on Learning and Intelligent Optimization*. Springer, 1–15.
- [6] Saber M. Elsayed, Ruhul A. Sarker, Daryl Leslie Essam, and Noha M. Hamza. 2014. Testing united multi-operator evolutionary algorithms on the CEC2014 real-parameter numerical optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014*. IEEE, 1650–1657.
- [7] Álvaro Fialho, Marc Schoenauer, and Michèle Sebag. 2010. Toward comparison-based adaptive operator selection. In *Genetic and Evolutionary Computation Conference, GECCO 2010, Proceedings, Portland, Oregon, USA, July 7-11, 2010*. 767–774.
- [8] Matthew D Hoffman, Eric Brochu, and Nando de Freitas. 2011. Portfolio Allocation for Bayesian Optimization. In *UAI*. Citeseer, 327–336.
- [9] Bernardo A Huberman, Rajan M Lukose, and Tad Hogg. 1997. An economics approach to hard computational problems. *Science* 275, 5296 (1997), 51–54.
- [10] Yaochu Jin. 2005. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing* 9, 1 (2005), 3–12.
- [11] Yaochu Jin. 2011. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation* 1, 2 (2011), 61–70.
- [12] Yaochu Jin and Bernhard Sendhoff. 2009. A systems approach to evolutionary multiobjective structural optimization and beyond. *IEEE Computational Intelligence Magazine* 4, 3 (2009).
- [13] Yaochu Jin, Handing Wang, Chugh Tinkle, Guo Dan, and Miettinen Kietinen. 2018. Data-Driven Evolutionary Optimization: An Overview and Case Studies. *IEEE Transactions on Evolutionary Computation* (2018), 1–1. <https://doi.org/10.1109/TEVC.2018.2869001>
- [14] Donald R Jones, Matthias Schonlau, and William J Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 13, 4 (1998), 455–492.
- [15] Michael N. Katehakis and Arthur F. Veinott Jr. 1987. The Multi-Armed Bandit Problem: Decomposition and Computation. *Math. Oper. Res.* 12, 2 (1987), 262–268.
- [16] Fei Peng, Ke Tang, Guoliang Chen, and Xin Yao. 2010. Population-based algorithm portfolios for numerical optimization. *IEEE Transactions on evolutionary computation* 14, 5 (2010), 782–800.
- [17] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. 2016. Taking the human out of the loop: A review of bayesian optimization. *Proc. IEEE* 104, 1 (2016), 148–175.
- [18] David Lupien St-Pierre and Jialin Liu. 2014. Differential Evolution algorithm applied to non-stationary bandit problem. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014*. IEEE, 2397–2403.
- [19] Ke Tang, Peng Yang, and Xin Yao. 2016. Negatively Correlated Search. *IEEE Journal on Selected Areas in Communications* 34, 3 (2016), 542–550.
- [20] Hao Tong, Changwu Huang, Jialin Liu, and Xin Yao. 2019. Voronoi-based Efficient Surrogate-assisted Evolutionary Algorithm for Very Expensive Problems. (2019). [arXiv:1901.05755](https://arxiv.org/abs/1901.05755)
- [21] Jasper A. Vrugt, Bruce A. Robinson, and James M. Hyman. 2009. Self-Adaptive Multimethod Search for Global Optimization in Real-Parameter Spaces. *IEEE Transactions on Evolutionary Computation* 13, 2 (2009), 243–259.
- [22] Handing Wang, Yaochu Jin, and John Doherty. 2017. Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Transactions on Cybernetics* 47, 9 (2017), 2664–2677.
- [23] Handing Wang, Yaochu Jin, and Jan O Jansen. 2016. Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system. *IEEE Transactions on Evolutionary Computation* 20, 6 (2016), 939–952.
- [24] David H Wolpert and William G Macready. 1997. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation* 1, 1 (1997), 67–82.
- [25] Lin Xu, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. 2008. SATzilla: portfolio-based algorithm selection for SAT. *Journal of artificial intelligence research* 32 (2008), 565–606.
- [26] Haibo Yu, Ying Tan, Chaoli Sun, and Jianchao Zeng. 2019. A generation-based optimal restart strategy for surrogate-assisted social learning particle swarm optimization. *Knowledge-Based Systems* 163 (2019), 14–25.
- [27] Shiu Yin Yuen, Chi Kin Chow, Xin Zhang, and Yang Lou. 2016. Which algorithm should I choose: an evolutionary algorithm portfolio approach. *Applied Soft Computing* 40 (2016), 654–673.
- [28] Tiantian Zhang, Michael Georgiopoulos, and Georgios C. Anagnostopoulos. 2014. Online model racing based on extreme performance. In *Genetic and Evolutionary Computation Conference, GECCO '14, Vancouver, BC, Canada, July 12-16, 2014*. ACM, 1351–1358.