Composing Algorithm Portfolio with Problem Set of Unknown Distribution

Wenwen Liu, Shiu Yin Yuen, Chi Wan Sung
Department of Electrical Engineering
City University of Hong Kong
Kowloon, Hong Kong, China
Email: wenwen.liu@my.cityu.edu.hk
kelviny.ee@cityu.edu.hk
albert.sung@cityu.edu.hk

***Abstract*—Portfolio approaches aim to combine different algorithms and take advantages of their strengths since it is challenging for a single algorithm to solve a problem set with all optimization problems. When there are many algorithms to choose from, there are various algorithm combinations of possibilities. Selecting a well-composed algorithm portfolio becomes essential to solve a problem set efficiently. In this paper, we propose a problem set dependent method to automatically and generically accomplish portfolio construction. The problem set can be composed of any problem, and come from any distribution or any benchmark. To decide which algorithm should be added in the portfolio, we utilize the average rank of results from solving problems to find the best-performing algorithm. Then we select complementary algorithms for the portfolio by applying Pearson correlation coefficient of fitness values. The method then iterates to compose more and more complex portfolios until there is no more improvement. This method is tested under three different problem sets. The experimental result shows the good ability of this approach to detect well-cooperated algorithms; and the composed portfolio is proved to have good adaptability to the problem set.**

1. Introduction

Numerous algorithms are proposed to solve numerical optimization problems in recent decades. Evolutionary Algorithms (EA) is a class of stochastic optimization techniques that have shown their abilities to solve various kinds of optimization problems. Typical EAs, for example Artificial Bee Colony algorithm (ABC) [1], Particle Swarm Optimization (PSO) [2] and Differential Evolution (DE) [3], are inspired by biological operations in nature. Besides, there are other effective algorithms to solve optimization problems, such as Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [4] and Broyden–Fletcher–Goldfarb–Shanno (BFGS) method [5]. Experiments show that they have competitive abilities in solving various classes of numerical optimization problems. Without loss of generality, we assume that we are solving single objective minimizaiton problems in this paper.

Algorithms have different strengths and weaknesses in solving optimization problems. The No Free Lunch (NFL) theorems [6] reveal that there is no so-called best algorithm that can defeat other algorithms on all types of problems. Therefore, it is natural to compose an algorithm portfolio to solve a problem set with a variety of optimization problems effectively. Since each algorithm in the portfolio can utilize its advantages on its well-performed problems, it is reasonable to infer that the algorithm portfolio should have greater confidence in achieving better average performance on a problem set. Indeed, previous works have shown that the cooperative multiple algorithms in the portfolio is better than a single algorithm [7] [8]. Typical portfolio approaches are classified into no information exchange methods [9] [10] and information exchange methods [11] [12]. These methods use different strategies to combine algorithms to search the landscape. Experimental results confirm that a portfolio outperforms single individual algorithms.

Based on the observation that algorithm portfolio methods are beneficial to solve a set of problems, the construction of the portfolio is also important. Hong and Page state the core of constructing an algorithm portfolio is to choose algorithms with diversity [13]. Hart and Sim prove that complementary algorithms combinations are more likely to improve overall performance [14]. There are also some practical schemes proposed to guide how to compose an algorithm portfolio. Tang et al. extend Population-based Algorithm Portfolios (PAP) [12] to PAP based on Estimated Performance Matrix (EPM-PAP) [15]. They introduce a portfolio construction method by choosing the combination with minimal risk. They utilize EPM, which is obtained by running individual algorithm j on each of the n problems r times, to formulate the probability that algorithms in the portfolio outperform others. The risk that the portfolio fails to win the individual algorithms is estimated by the probability. In this way, the construction of the portfolio is converted to the object of finding the least risky combination. Independently, Yuen and Zhang offer a parameterless plan to compose the portfolio generically. They apply a ranking method to select algorithms [16]. Given q algorithms, the average rankings of 25 classes of functions in CEC benchmark are used to find the best performing

2020 IEEE Symposium Series on Computational Intelligence (SSCI)
December 1-4, 2020, Canberra, Australia
978-1-7281-2547-3/20/$31.00 ©2020 IEEE

algorithms, and covariance of rankings, which is a q by q matrix, is utilized to find complementary algorithms. After forming a portfolio with the lowest-rank algorithm and its most uncorrelated algorithm, they apply MultiEA [9] on this portfolio to solve problems. The portfolio is considered as a new algorithm. Then, the loop of finding the best performing algorithm and its complementary algorithm is repeated until the best portfolio is found.

Meanwhile, other approaches of portfolio construction, which do not specify algorithm interaction method like PAP and MultiEA, are also proposed [17] [18]. With rapidly improved computing resource, cooperation mechanism between algorithms, and allocation of evaluation budget are not most important for algorithms in parallel portfolios. All algorithms in the portfolio run simultaneously until one solves the problem. Therefore, finding representative instances for problem sets are the core for these approaches. For example, the Identification of Complementary Algorithms by Uncovered Sets (ICARUS) proposed by Muñoz and Kirley, use voting system theory to determine algorithm combinations [17]. This approach attempts to find the smallest algorithm set, which can cover all instances in the problem set. Recently, Liu et al. propose a new Generative Adversarial Solver Trainer (GAST) to construct parallel portfolios. Except for using the problems of the set as training instances, the trainer generates new instances to increase the diversity of the training set. With more training instances, the selected algorithm portfolio better generalizes [18].

To evaluate their algorithm portfolio composition approaches, researches apply their method on solving various optimization problems. For example, ICARUS is tested on BBOB benchmarks, and GAST is tested on TSP and SAT problems. In algorithm design and evaluation, the IEEE Congress on Evolutionary Computation (CEC) benchmarks [19] and the black-box optimization benchmark (BBOB) [20] are widely used as training and test suite. However, when solving the practical problems, the winners of these benchmarks are not always perfectly customized. This is because in the real world, it rarely exists that every problem in the benchmark has the same probability of occurrence. Additionally, some real-world problems even do not have a representative in the popular benchmarks. To supplement the baseline problems, some tunable benchmark generators are also designed to offer multiple landscape choices. For example, the Max-Set of Gaussian (MSG) landscape generator [21] and the multi-modal landscape generator [22] provide diversified problem landscapes generated synthetically. Some methods form benchmark test suite based on whether algorithms solve the problems easily [23]. Therefore, how to construct algorithm portfolios for non-uniformly distributed problem sets, which include problems that occur with unequal probabilities and problems from infrequent using benchmarks, are worth discussing.

In this paper, we propose a problem set dependent algorithm portfolio construction method and apply the portfolio with MultiEA to solve the optimization problem. The new approach can be flexibly applied to problems with any unknown probability distribution of functions occurrence, as long as the distribution is independent and identically distributed (i.i.d). Moreover, the problems can be selected from any benchmarks. The relaxed requirements of problems offer an algorithm portfolio composing scheme for diverse problem sets. To determine the complementary algorithms, we utilize Pearson correlation coefficient of fitness values solved by algorithms. This is better than the covariance matrix approach in [16], as the Pearson correlation coefficient normalize the variance of the two random variables and values are normalized to lie between $[-1, 1]$. Thus, it removes the bias due to the variances. In experiments, we test our composed portfolio with MultiEA by allocating the same evaluation budget as single algorithms under three scenarios. In former two experiments, we test our approach with uniformly and non-uniformly sampled problems from CEC 2013 benchmarks. In the last experiment, we test with function-less problems from MSG landscape generator. The experimental results show that our approach performs well on finding complementary algorithms and composing the portfolios.

The rest of this paper is organized as follows: Section 2 introduces MultiEA and Pearson correlation coefficient, and states our new algorithm portfolio composition approach. Section 3 reports and discusses our experiment results of three experiments. The last section contains the conclusion and future direction.

2. Algorithm Design

2.1. MultiEA

In this design, we adopt multiEA [9] as the method to implement the algorithm portfolio. The same evaluation budget as single algorithms is allocated to MultiEA. With the given algorithm portfolio, MultiEA dynamically selects suitable algorithms to solve the optimization problem generation by generation.

In the first few generations, all algorithms use the same evaluation budget to set up the initial iterations. Then, the data of fitness values and iterations return the convergence curves for each EA in the portfolio. According to the convergence curves, MultiEA predicts the fitness value in the nearest future based on the bootstrap probability. At a common future point, all anticipated results are compared, and the algorithm with the best-predicted value is selected to run one generation until the evaluation budget is used up. In detail, the steps of MultiEA are summarized as:

1) An algorithm portfolio with complementary elements is given.
2) Algorithms set up initial iterations with the same evaluation budgets.
3) With historical data of iterations, each algorithm returns convergence curves.
4) Set a common nearest future point; each algorithm predicts future fitness values based on the convergence curves.

5) For each algorithm, sampling a result based on the bootstrap probability of predicted fitness values.
6) Select the algorithm with the best result to run one iteration.
7) Update the historical iteration data, repeat steps from 4) to 7) until the evaluation budget is used up.

MultiEA has been shown to be an effective algorithm portfolio method from experimental results in [9]. With the operation of dynamically selecting suitable algorithms, it allocates the evaluation budget automatically. Besides, the parameter-less character provides a lot of convenience for the combination of algorithms. It should be noted that MultiEA does not exchange information among algorithms. Therefore, the wanted components of algorithms in the portfolio is supposed to have diverse performance in solving different kinds of problems. Similar performances are likely to make algorithms compete with each other, wasting computing resources. For example, in the case that several algorithms has close fitness values in every iteration, they are likely to share the budget uniformly. If so, no algorithm has enough evaluation budget to converge. Therefore, the right strategies for detecting complementary algorithms for MultiEA are essential.

2.2. Pearson correlation coefficient

Pearson correlation coefficient [24] is commonly used in data science and engineering. It is an index to represent the correlation relationship between two random variables. The range of the coefficient is from -1 to 1. A positive number implies a positive correlation relationship between two variables, that is, one variable increases or decreases as the other one increases or decreases respectively, while a negative number vice versa. The larger the number, the stronger the correlation between the two variables.

A positive correlation coefficient implies that two algorithms have similar trends in solving the problem set, which means two algorithms are likely to be good at the same types of problems. On the contrary, a negative correlation coefficient means algorithms are likely to have inverse performance on some problems, which are the characteristics for complementary algorithms. For example, algorithm a solves a set of 4 problems with fitness value $f_a = [1, 2, 1, -1]$ while algorithm b returns $f_b = [-0.7, -2, 2, 1]$. If the objective is to get the global minimum value of the problem, algorithm b performs better on the first two problems, and algorithm a is good for solving the last two problems. The correlation coefficient $\rho = -0.5$ between these two groups of data shows that a and b are complementary to a certain extent. However, in case that $f_a = [1, 2, 1, -1]$ and $f_b = [1.2, 2.5, 1, 1]$, algorithm b obtains worse or equal results compared to algorithm a in solving every problem in the set. The correlation coefficient is $\rho = 0.6$, which reflects that these two algorithms are in a competitive relationship.

2.3. Composing algorithm portfolios

Assume a set of optimization problems $F = \{f_i | i = 1, 2, ..., n\}$ with n problems and m candidate algorithms set $A = \{a_j | j = 1, 2, ..., m\}$ are given. We aim to select the best algorithm set $AP \subseteq A$, and $AP = \{PA_k | k = 1, 2, ..., l\}$ to apply MultiEA to solve F. For m candidate algorithms in the algorithm pool, the total number of combinations of subsets $AP \subseteq A$ is $2^m - 1$. With increasing number of candidate algorithms, the number of subsets exponentially increases. Among so many combinations, choosing a group of algorithms to collaboratively complete a high-performance optimized search is the object of this new proposed method.

We first choose the best performing algorithm for the problem set, then progressively expand the algorithms portfolio by adding the complementary algorithms. Since the evaluation of the “best” algorithm depends on the requirement of the problems, we use average rank of problems to represent the abilities of algorithms to solve the set. It should be noted that the lower the rank is, the better the algorithm performs. The calculation of rank is based on the procedure of Friedman’s test [25].

In the algorithm design, the first step is to select a simulation problem suite F with n problems. This problem suite can be made up from any kinds of problems. Problems sampled from any i.i.d distribution of functions are acceptable. Moreover, there is no limitation of the benchmark source for problems as well. Then we run all the m algorithm candidates $\{a_1, a_2, ..., a_m\}$ on the chosen problem suite and record the results of each problem. After running the suite, for each algorithm, a fitness value array of length n is recorded. Algorithm PA_1 with the lowest average rank should be firstly added to the portfolio $AP = \{PA_1\}$. It is reasonable to infer that PA_1 may not be the best algorithm for some of the problems in the suite. Thus, the next step is to find the algorithm PA_2 to add into the portfolio with the smallest Pearson correlation coefficient between AP and PA_2. Then, portfolio $AP = \{PA_1, PA_2\}$ is implemented by MultiEA. AP can also be considered as a new individual algorithm a_{m+1}. We can repeat this process until the performance of portfolio starts to deteriorate.

The method is similar to the composing method of Yuen and Zhang [16], but two key differences: 1) The distribution is relaxed. It does not need to be uniform. Any i.i.d distribution suffices; 2) The more unbiased Pearson correlation coefficient replaces the covariances.

The full algorithm is depicted in pseudo code of Algorithm 1.

3. Experiments

To verify the feasibility of the idea to compose the problem set dependent algorithm portfolio, we test the approach both with CEC 2013 benchmarks and tunable benchmark MSG generator. CEC 2013 benchmarks have 28 functions, containing different types of problems, covering three groups: unimodal functions, multimodal functions,

Algorithm 1: Compose an algorithm portfolio

input : A set of optimization problems $F = \{f_i | i = 1, 2, ..., n\}$ with dimension D and evaluation budget $maxFE$; m candidate algorithms $A = \{a_j | j = 1, 2, ..., m\}$

output: Best performing algorithm portfolio $AP = \{PA_1, PA_2, ..., PA_k\} \subseteq A$ on solving problem set F

```
Run a_j, j = 1, 2, ..., m on set F with budget maxFE;
Calculate average ranks for A;
Let PA_1 = a_j with lowest average rank;
Let AP = {PA_1};
s = 1;
while true do
    Find smallest ρ(AP, PA_{s+1});
    Let portfolio AP^ = {AP, PA_{s+1}};
    Apply MultiEA on AP^ to solve F with budget maxFE;
    Calculate average ranks for A and AP^;
    if AP^ doesn't have lowest rank then
        break;
    end
    AP = AP^;
    Take AP as a_{m+s}, adding to algorithm set A;
    s = s + 1;
end
```

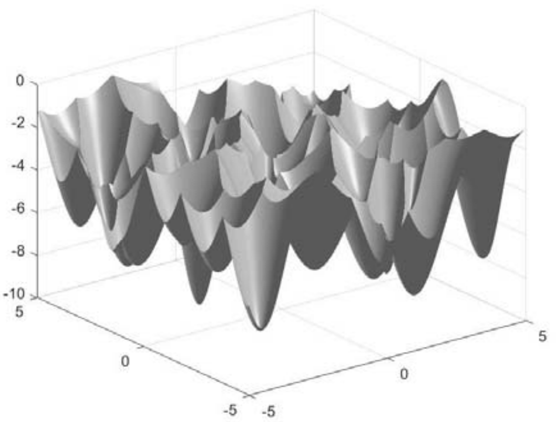

Figure 1. An example of MSG generator landscape with dimension $D = 2$ and $nGaussian = 100$

and composition functions [19]. MSG generator is used to generate continuous, bound-constrained optimization problems with the Multi-Gaussian mixed landscape. A typical example for MSG function with dimension $D = 2$ and number of Gaussian $nGaussian = 100$ is Figure 1.

According to the settings of benchmarks, we apply our approach to three different problem sets with the following scenarios to test its adaption.

1) A problem set with uniformly sampled CEC 2013 benchmark functions.
2) A problem set whose problems are mainly sampled from unimodal functions and rest of the problems are from other functions of the CEC 2013 benchmark.
3) A problem set sampled from tunable benchmark MSG generator.

We select 10 popular algorithms in the EC community. These algorithms are classical evolutionary strategy CMA-ES [4], influential swarm intelligence methods: ABC [1] and PSO [2], Global and Local real-coded genetic algorithms (GL-25) [26], powerful DE algorithms: jDE [27], Self-adaptive Differential Evolution algorithm (SaDE) [28], Adaptive Differential Evolution With Optional External Archive algorithm (JaDE) [29], Composite DE (CoDE) [30], L-Shade [31], and hill climbing optimization technique BFGS [5]. In all experiments, we adopt default settings of algorithms.

All algorithms and benchmarks in the experiments are coded by Matlab. We run on a PC with Intel 3.4GHz Core I7 processor with 24GB RAM.

3.1. Uniformly sampled problems from CEC 2013 benchmark

In the first experiment, we randomly sample a fixed number of problems from CEC 2013 benchmark as the problem set. Let all 10 algorithms solve the problems in the set until the default maximum number of function evaluations ($maxFE = 10000D$) is achieved. In this experiment, problem number $n = 300$. Each problem is uniformly sampled from 28 functions of CEC 2013 benchmark. The problems in this test suite are uniformly sampled from 28 functions in CEC 2013 benchmark. After running the suite, we calculate the average rank of the 10 algorithms and select the one with the lowest rank. Table 1 shows that L-Shade is the best performing algorithm. The bold number is the value of lowest average rank. Then, the next step is to find the complementary algorithm for L-Shade according to Pearson correlation coefficient. After searching the matrix in Table 2, ABC is the most complementary algorithm for L-Shade with $\rho = -0.134$ in bold. Therefore, the portfolio composed for the first round is AP = {L-Shade and ABC}.

Next, we use MultiEA with portfolio AP to solve the problem set and re-rank the results with 10 individual algorithms. The average rank for this round is in Table 3. Note that we record the portfolio composed in round1 as $AP1$. The results show that current algorithm portfolio $AP1$ = {L-Shade, ABC} is the best-performing. Then we follows the steps in Algorithm 1 to keep finding the complementary algorithm in Table 4. Because this correlation matrix has many overlapping parts with Table 2, and the complementary algorithms of non-optimal algorithms are not important, we

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO
Average rank	5.577	7.943	6.805	3.337	4.965	3.590	9.030	**3.273**	4.238	6.242

TABLE 1. AVERAGE RANK FOR 10 ALGORITHMS IN EXPERIMENT OF UNIFORMLY SAMPLING PROBLEMS FROM CEC 2013

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO
BFGS	1.000	-0.164	0.828	0.568	-0.076	0.498	-0.111	0.587	-0.093	-0.048
ABC		1.000	-0.165	-0.132	0.296	-0.142	0.439	**-0.134**	0.426	0.025
CMA-ES			1.000	0.541	-0.077	0.460	-0.111	0.550	-0.096	-0.037
CoDE				1.000	-0.057	0.790	-0.088	0.863	-0.064	-0.039
GL25					1.000	-0.063	-0.018	-0.058	0.433	0.004
JaDE						1.000	-0.097	0.911	-0.066	-0.042
jDE							1.000	-0.091	0.056	0.255
L-Shade								1.000	-0.060	-0.039
SaDE									1.000	0.031
SPSO										1.000

TABLE 2. PEARSON CORRELATION COEFFICIENT MATRIX FOR 10 ALGORITHMS IN EXPERIMENT 1-ROUND 1

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO	AP1
Average rank	5.978	8.618	7.305	4.267	5.983	4.412	9.570	4.172	4.962	6.722	**4.012**

TABLE 3. AVERAGE RANK FOR 11 ALGORITHMS IN EXPERIMENT 1-ROUND 2

just attached the Pearson correlation coefficient matrix of the optimal algorithms in the following sections. It is observed that the Pearson correlation coefficient ρ between portfolio algorithm and other individual ones decreases, which indicates the algorithm portfolio reduces the diversities. With $\rho = -0.044$, the next algorithm that should be added into the portfolio is BFGS algorithm. After continuing to use MultiEA with AP= {L-Shade, ABC, BFGS} to apply the algorithm portfolio to solve the problem set, the performance of the three-algorithm portfolio is in Table 5. Note that $AP1$ and $AP2$ represent the portfolio of $AP1$ = {L-Shade, ABC} and $AP2$ = {L-Shade, ABC, BFGS}. The complementary algorithm for the current best performing algorithm is in Table 6. Both BFGS algorithm and CMA-ES return the same Pearson correlation coefficient. Since BFGS is already in the portfolio, we add CMA-ES to compose $AP3$ = {L-Shade, ABC, BFGS, CMA-ES} in this round.

Seen from Table 7, $AP3$ has the best performance. So we can continue to add the new complementary algorithm following the steps. From Table 8, the most uncorrelated algorithm is CMA-ES. Because CMA-ES is already in the portfolio, we add the next uncorrelated algorithm jDE into the portfolio. It is shown in Table 9 that $AP4$ = {L-Shade, ABC, BFGS, CMA-ES, jDE} performs worse than $AP3$ = {L-Shade, ABC, BFGS, CMA-ES}. It is because there is an exploration-exploitation trade-off of the portfolio as the evaluation budget for MultiEA is limited. The more algorithms in the portfolio, the more evaluation budget spent on exploration is needed. Thus, the algorithms in the portfolio do not have enough budget to converge. In this experiment, we find that adding new algorithms for $AP3$ deteriorates the performance, so the best algorithm portfolio for the problem set is $AP3$ = {L-Shade, ABC, BFGS, CMA-ES}.

3.2. Non-uniformly sampled problems from CEC 2013 benchmark

In this section, we evaluated the approach with a biasedly sampled problem set with i.i.d problems. We investigate the case that most of the problems are from the unimodal group while a small part is from other problem groups. As we know, method like BFGS, which utilizes the gradients of sampled points, can find the global optimum of unimodal functions very fast. But it is very easy to get trapped in local optimum when solving multi-modal problems. If there is a problem set with problems mostly sampled from unimodal functions and a few multi-modal problems, a well-composed algorithm portfolio including BFGS is likely to improve the performance compared to the individual algorithms. Therefore, we verify the ability of our composition approach. In this scenario, we note that both [15] [16] assume uniform sampled problems and cannot handle this situation.

Since the setting of $D = 10$ and $maxFE = 10000D$ to solve unimodal problems provides more than sufficient computing resources, every algorithm has enough budget to find the global optimum for the unimodal problems. Then almost all algorithms get equal performances. So we reduce the evaluation budget to $maxFE = 500D$ to solve this problem set. We also set problem number $n = 200$ and 95% of the problems to be unimodal problems, which is f1 to f5 in CEC 2013 benchmark, while the rest of the problems (5%) are randomly selected from other problems from f6 to f28.

As shown in Table 10 and Table 11 as expected, the best performing algorithm is BFGS and its complementary algorithm is CMA-ES. After composing the algorithm portfolio $AP1$ = {BFGS, CMA-ES} for this round, the performance and the correlation matrix are in Table 12 and Table 13. It can be observed that L-Shade is with the smallest correlation coefficient with the current best algorithm portfolio $AP1$ =

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO	AP1
ρ for current best algorithm	**-0.044**	0.105	-0.043	-0.033	-0.012	-0.038	0.304	-0.036	-0.009	0.003	1.000

TABLE 4. Pearson correlation coefficient matrix for current best performed algorithms in Experiment 1-round 2

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO	AP1	AP2
Average rank	5.970	5.887	6.023	6.793	7.570	6.337	7.140	6.933	6.730	6.497	6.280	**5.840**

TABLE 5. Average rank for 12 algorithms in experiment 1-round 3

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO	AP1	AP2
ρ for current best algorithm	**-0.057**	0.130	**-0.057**	-0.045	-0.015	-0.051	0.394	-0.049	-0.012	0.010	0.964	1.000

TABLE 6. Pearson correlation coefficient matrix for current best performed algorithms in Experiment 1-round 3

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO	AP1	AP2	AP3
Average rank	6.723	8.642	8.133	6.370	7.740	6.665	10.097	6.303	6.720	7.700	5.327	5.452	**5.128**

TABLE 7. Average rank for 13 algorithms in experiment 1-round 4

{BFGS, CMA-ES}, but the value of ρ is positive. As we know, positive correlation coefficient represents the same trend of two groups of fitness values, in which one can infer that the two algorithms are not complementary. Therefore, the newly added algorithm is likely to be redundant, wasting the evaluation budget. To verify the hypothesis, in Table 14, $AP2$ = {BFGS, CMA-ES, L-Shade} has higher average rank than $AP1$ = {BFGS, CMA-ES}. In fact, $AP2$ does badly with an average rank of 6.42. This suggests that competing algorithms are in the present portfolio, which is counter predictive to the optimization process. Therefore, the best-found portfolio for this problem set is $AP1$.

3.3. Problems from MSG landscape generator

In this experiment, we investigate the ability of this approach to find portfolio for other problem sets. We apply the MSG landscape generator to provide the optimization problems. We estimate the setting for the problem set with problem number $n = 100$, dimension $D = 10$, input range for each dimension is $[-5, 5]$, evaluation budget $maxFE = 1000D$. Because there is no equations for problems in the benchmark, the settings of the functions is that all mixed Gaussian functions have magnitude of $[-10, 0]$, the mean and standard deviation are arbitrarily selected, and the number of Gaussian $nGaussin$ is an integral randomly selected from $[2, 200]$.

Seen from Table 15 and Table 16, the algorithm with lowest rank is L-Shade. CMA-ES is added in the portfolio as its companion.

As shown in Table 17, the composed portfolio $AP1$ = {L-Shade,CMA-ES} ranks 1st among all algorithms with its correlation matrix in Table 18.

In Table 19, the rank of portfolio $AP2$ = = {L-Shade, CMA-ES, JaDE} is not lower than $AP1$ = {L-Shade, CMA-ES}. Therefore, the best portfolio for this problem set is $AP1$ = {L-Shade, CMA-ES}. Again, $\rho = 0.033$ in Table 18 gives an early warning of competing algorithms within the portfolio.

This experiment is an example of applying our portfolio composition method on function-less benchmark and problems. The performance of the portfolio shows the approach can well customize to suit optimization needs for problem sets with unique problems. In the real world, in the case that a known problem set is well simulated unsolved practical problems, this approach may offer some inspiration to determine the algorithm portfolios.

4. Conclusion

In recent decades, various types of algorithms have exhibited their strong abilities in solving numerical optimization problems. However, how to take advantages of their strengths require efficient strategies to compose the algorithms portfolio.

Assume that there are the hidden relationships between algorithms and problem sets, which reflect by the solved fitness values, we attempt to find a general method to compose the algorithm portfolio. In this paper, we propose a problem set dependent method to compose algorithm

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO	AP1	AP2	AP3
ρ for current best algorithm	0.492	**-0.117**	0.521	0.616	-0.059	0.642	**-0.064**	0.619	-0.051	-0.032	-0.027	0.692	**1.000**

TABLE 8. Pearson correlation coefficient matrix for current best performed algorithms in Experiment 1-round 4

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO	AP1	AP2	AP3	AP4
Average rank	7.142	9.320	8.658	6.678	8.260	7.177	10.760	7.075	7.338	8.662	6.108	6.248	**5.742**	5.832

TABLE 9. Average rank for 14 algorithms in experiment 1-round 5

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO
Average rank	**2.87**	9.255	3.39	6.33	4.365	4.475	9.29	3.35	4.825	6.85

TABLE 10. AVERAGE RANK FOR 10 ALGORITHMS IN EXPERIMENT 2-ROUND 1

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO
ρ for current best algorithm	1.000	0.271	**-0.008**	0.122	0.053	0.080	0.098	0.034	0.060	0.116

TABLE 11. PEARSON CORRELATION COEFFICIENT MATRIX FOR CURRENT BEST PERFORMED ALGORITHMS IN EXPERIMENT 2-ROUND 1

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO	AP1
Average rank	4.003	10.090	3.823	7.400	4.698	5.095	10.375	4.310	5.735	7.845	**2.938**

TABLE 12. AVERAGE RANK FOR 11 ALGORITHMS IN EXPERIMENT 2-ROUND 2

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO	AP1
ρ for current best algorithm	0.365	0.569	0.259	0.382	0.095	0.187	0.495	**0.056**	0.656	0.480	1.000

TABLE 13. PEARSON CORRELATION COEFFICIENT MATRIX FOR CURRENT BEST PERFORMED ALGORITHMS IN EXPERIMENT 2-ROUND 2

portfolios automatically and use MultiEA as the portfolio strategy. Compared to the previous approaches in [15] [16], the number of functions does not necessarily to be known and the functions does not need to specify probability of occurrence in the problem set. Pearson correlation coefficient, which is less biased than the covariance in [16], is also introduced. This new method is more flexible to adapt to all types of problem sets.

With 10 algorithms in the algorithm pool, we test this approach under three different scenarios. Three experiments are conducted. The first experiment shows that the method can compose a portfolio when the problem distribution is uniform. The second experiment shows that when the problem distribution is unknown, the method can compose a better portfolio that varies with the problem distribution to improve average performance. The last experiment demonstrates that the method works when the problem class is not known to the portfolio in advance. The result shows the Pearson correlation coefficient can find complementary algorithms. Moreover, we demonstrate that different portfolios are generated when the probability distribution is changed, adapting to the changed problem sets. Furthermore, it is shown that the composed portfolio has superior performance compared with individual algorithms.

In future work, we plan to further investigate this method with other portfolio strategies and investigate its ability in real-world optimization problems. Moreover, though this paper has assumed that the unknown distribution of problems are i.i.d., it is straight forward to extend the approach to time varying distribution. In this scenario, number of problems is the time window used to compose the portfolio. As time progresses, the time window with size of problem set will slide and the corresponding portfolio constructed will be time varying. Further work will investigate how to select best problems.

Acknowledgments

Wenwen Liu is supported by the Institutional Postgraduate Studentship of City University of Hong Kong.

References

[1] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer ..., 2005.

[2] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," Swarm intelligence, 2007.

[3] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," IEEE Trans. Evol. Comput., vol. 15, no. 1, pp. 4–31, Feb. 2011.

[4] N. Hansen, "The CMA Evolution Strategy: A Comparing Review," in Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms, J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 75–102.

[5] R. Fletcher, Practical Methods of Optimization. John Wiley & Sons, 2013.

[6] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," IEEE Trans. Evol. Comput., vol. 1, no. 1, pp. 67–82, Apr. 1997.

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO	AP1	AP2
Average rank	4.29	10.955	4.08	8.02	5.095	5.535	11.29	4.72	6.26	8.55	**3.095**	6.42

TABLE 14. AVERAGE RANK FOR 12 ALGORITHMS IN EXPERIMENT 2-ROUND 3

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO
Average rank	5.785	8.86	6.3	4.08	3.745	4.48	9.61	**3.725**	3.935	4.48

TABLE 15. AVERAGE RANK FOR 10 ALGORITHMS IN EXPERIMENT 3-ROUND 1

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO
ρ for current best algorithm	0.341	0.225	**-0.095**	0.278	0.374	0.308	-0.019	1.000	0.520	0.403

TABLE 16. Pearson correlation coefficient matrix for current best performed algorithms in Experiment 3-round 1

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO	AP1
Average rank	6.045	9.94	7.3	4.22	5.1525	5.0725	10.605	4.35	4.2475	4.94	**4.1275**

TABLE 17. Average rank for 11 algorithms in experiment 3-round 2

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO	AP1
Pearson coefficient	0.153	0.131	0.237	0.146	0.361	**0.033**	0.075	0.308	0.276	0.137	1.000

TABLE 18. Pearson correlation coefficient matrix for current best performed algorithms in Experiment 3-round 2

Algorithms	BFGS	ABC	CMA-ES	CoDE	GL25	JaDE	jDE	L-Shade	SaDE	SPSO	AP1	AP2
Average rank	7.065	11.13	8.065	5.07	5.665	5.095	11.53	5.715	4.75	5.05	**4.155**	4.71

TABLE 19. Average rank for 12 algorithms in experiment 3-round 3

[7] E. K. Burke et al., “Hyper-heuristics: a survey of the state of the art,” J. Oper. Res. Soc., vol. 64, no. 12, pp. 1695–1724, Dec. 2013.

[8] O. Ertenlice and C. B. Kalayci, “A survey of swarm intelligence for portfolio optimization: Algorithms and applications,” Swarm and Evolutionary Computation, vol. 39, pp. 36–52, Apr. 2018.

[9] S. Y. Yuen, C. K. Chow, X. Zhang, and Y. Lou, “Which algorithm should I choose: An evolutionary algorithm portfolio approach,” Appl. Soft Comput., vol. 40, pp. 654–673, Mar. 2016.

[10] Y. He, S. Y. Yuen, Y. Lou, and X. Zhang, “A sequential algorithm portfolio approach for black box optimization,” Swarm and Evolutionary Computation, vol. 44, pp. 559–570, Feb. 2019.

[11] J. A. Vrugt, B. A. Robinson, and J. M. Hyman, “Self-Adaptive Multimethod Search for Global Optimization in Real-Parameter Spaces,” IEEE Trans. Evol. Comput., vol. 13, no. 2, pp. 243–259, Apr. 2009.

[12] F. Peng, K. Tang, G. Chen, and X. Yao, “Population-Based Algorithm Portfolios for Numerical Optimization,” IEEE Trans. Evol. Comput., vol. 14, no. 5, pp. 782–800, Oct. 2010.

[13] L. Hong and S. E. Page, “Groups of diverse problem solvers can outperform groups of high-ability problem solvers,” Proc. Natl. Acad. Sci. U. S. A., vol. 101, no. 46, pp. 16385–16389, Nov. 2004.

[14] E. Hart and K. Sim, “On Constructing Ensembles for Combinatorial Optimisation,” Evol. Comput., vol. 26, no. 1, pp. 67–87, Spring 2018.

[15] K. Tang, F. Peng, G. Chen, and X. Yao, “Population-based Algorithm Portfolios with automated constituent algorithms selection,” Inf. Sci. , vol. 279, pp. 94–104, Sep. 2014.

[16] S. Y. Yuen and X. Zhang, “On composing an algorithm portfolio,” Memetic Computing, doi: 10.1007/s12293-015-0159-9.

[17] M. A. Muñoz and M. Kirley, “ICARUS: Identification of complementary algorithms by uncovered sets,” in 2016 IEEE Congress on Evolutionary Computation (CEC), Jul. 2016, pp. 2427–2432.

[18] S. Liu, K. Tang, and X. Yao, “Generative Adversarial Construction of Parallel Portfolios,” IEEE Trans Cybern, Apr. 2020, doi: 10.1109/TCYB.2020.2984546.

[19] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, “Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization,” Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report, vol. 201212, no. 34, pp. 281–295, 2013.

[20] “start [COmparing Continuous Optimisers: COCO],” Mar. 06, 2019. http://coco.gforge.inria.fr/

[21] M. Gallagher and Bo Yuan, “A general-purpose tunable landscape generator,” IEEE Trans. Evol. Comput., vol. 10, no. 5, pp. 590–603, Oct. 2006.

[22] J. Rönkkönen, X. Li, V. Kyrki, and J. Lampinen, “A framework for generating tunable test functions for multimodal optimization,” Soft Computing, vol. 15, no. 9, pp. 1689–1706, 2011.

[23] Y. Lou and S. Y. Yuen, “On constructing alternative benchmark suite for evolutionary algorithms,” Swarm and Evolutionary Computation, vol. 44, pp. 287–292, Feb. 2019.

[24] J. Benesty, J. Chen, Y. Huang, and I. Cohen, “Pearson Correlation Coefficient,” in Noise Reduction in Speech Processing, I. Cohen, Y. Huang, J. Chen, and J. Benesty, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1–4.

[25] S. García, D. Molina, M. Lozano, and F. Herrera, “A study on the use of non-parametric tests for analyzing the evolutionary algorithms’ behaviour: a case study on the CEC’2005 special session on real parameter optimization,” Journal of Heuristics, vol. 15, no. 6, p. 617, 2009.

[26] C. García-Martínez, M. Lozano, F. Herrera, D. Molina, and A. M. Sánchez, “Global and local real-coded genetic algorithms based on parent-centric crossover operators,” Eur. J. Oper. Res., vol. 185, no. 3, pp. 1088–1113, Mar. 2008.

[27] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, “Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems,” IEEE Trans. Evol. Comput., vol. 10, no. 6, pp. 646–657, Dec. 2006.

[28] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization,” IEEE Trans. Evol. Comput., vol. 13, no. 2, pp. 398–417, Apr. 2009.

[29] J. Zhang and A. C. Sanderson, “JADE: Adaptive Differential Evolution With Optional External Archive,” IEEE Trans. Evol. Comput., vol. 13, no. 5, pp. 945–958, Oct. 2009.

[30] Y. Wang, Z. Cai, and Q. Zhang, “Differential Evolution With Composite Trial Vector Generation Strategies and Control Parameters,” IEEE Trans. Evol. Comput., vol. 15, no. 1, pp. 55–66, Feb. 2011.

[31] R. Tanabe and A. S. Fukunaga, “Improving the search performance of SHADE using linear population size reduction,” in 2014 IEEE Congress on Evolutionary Computation (CEC), Jul. 2014, pp. 1658–1665.