On Composing an (Evolutionary) Algorithm Portfolio

Shiu Yin Yuen
Department of Electronic Engineering
City University of Hong Kong
Hong Kong, China
kelviny.ee@cityu.edu.hk

Xin Zhang
Department of Electronic Engineering
City University of Hong Kong
Hong Kong, China
xinzhang9-c@my.cityu.edu.hk

ABSTRACT
In this paper, we propose a general methodology to automatically compose a good portfolio from a set of selected EAs. As a single EA is a degenerate portfolio, our method also provides an answer to when a portfolio of two or more EAs are beneficial. Our method has the nice property of being parameter-less; it does not introduce extra parameters. Hence there is no need for parameter control, which is well known to be a thorny research issue. To illustrate our idea, we show how a portfolio that is constructed by considering five state of the art EAs as candidates is automatically constructed from ten CEC 2005 benchmark functions. It is found that the resulting portfolio enjoys excellent, and equally importantly, stable ranking. Thus the new portfolio algorithm has the property of being a robust algorithm, which is a highly desirable property in practical applications.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Heuristic methods*; G.1.6 [**Numerical Analysis**]: Optimization – *Global optimization*.

General Terms
Algorithms, Experimentation

Keywords
Evolutionary algorithm, portfolio, global optimization

1. INTRODUCTION
In the companion paper [1], we report a novel algorithm portfolio approach called Multiple Evolutionary Algorithm (MultiEA). The portfolio is composed ABC, CMA-ES, CoDE, PSO and SaDE.

The present paper is a novel extension of MultiEA. We report a method that chooses algorithms to compose the portfolio *automatically*, in a *parameter-less* manner. Note that our method will automatically find out the situation if there is no need for a portfolio, i.e., only one EA is enough or equivalently, a portfolio of one algorithm is the best.

Both our ideas of MultiEA and the novel portfolio composing method are *generic*. They can be applied to any algorithms that can compute a predicted performance, be they EAs or non-EAs, though in this paper our portfolio is exclusively composed of EAs; hence the bracket ( ) in the title.

2. A NOVEL ALGORITHM THAT COMPOSES A PORTFOLIO AUTOMATICALLY
Given q EAs, this novel algorithm returns a portfolio AP that contains either one algorithm that presents on average the best performance or a MultiEA that presents the best average performance by combining several EAs.

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.
ACM 978-1-4503-1964-5/13/07.

What is the "best" algorithm depends on the requirements of applications. In this paper, we use a simple heuristic that the best algorithm is one that gives the lowest average rank on a suite of benchmarks.

We run the given q EAs on the suite and then rank them based on the procedure of Friedman's test. The full algorithm is as follows:

Algorithm (Portfolio composition algorithm).
Input:
  A set of q EAs $\{A_1, \dots, A_q\}$; a set of benchmark functions S with dimension D; maximum number of evaluations N.
Step 1:
  Run $A_i, i = 1,2, \dots, q$ on benchmark set S for N evaluations.
  Calculate the ranks of all algorithms on S.
Step 2:
  $AP = \{PA1\}$, where $PA1$ is the algorithm with the lowest average rank.
Step 3:
  Calculate the $q \times q$ covariance matrix Cov of ranks.
Step 4:
  Find $PA2$ such that $Cov(PA1, PA2)$ is the smallest.
Step 5:
  Set up MultiEA= $\{PA1, PA2\}$. Treat this algorithm as a new individual algorithm A_{q+1}.
  Run A_{q+1} on benchmark set S.
Step 6:
  Calculate the ranks of all $q + 1$ algorithms on S.
  If A_{q+1} is the algorithm with the lowest average rank
    Set $PA1 = A_{q+1}$.
    Set $q \leftarrow q + 1$ and go to Step 3.
  Otherwise return $AP = PA1$.
Output:
  Portfolio AP.

3. NUMERICAL EXPERIMENTS
Functions $f_1 - f_{10}$ in the CEC 2005 are taken and D is set to 30. 25 independent runs are conducted for each algorithm and each test function. The termination criteria are to terminate the algorithm when one of the following conditions is satisfied:

1. The maximum number of function evaluations (maxFE) is reached (maxFE $= 10000 \times D$);
2. $|f(\boldsymbol{x}) - f(\boldsymbol{x}^*)| \le 10^{-6}$.

Based on the experimental results of the 5 EAs, the ranks are shown in the table below. The best results are shown in bold.

function	ABC	CMA-ES	CoDE	PSO	SaDE
f_1	3	**1**	5	2	4
f_2	5	**1**	3	2	4

f_3	5	**1**	2	3	4
f_4	5	**1**	2	3	4
f_5	5	**1**	2	4	3
f_6	3	**1**	2	5	4
f_7	3	**1**	2	5	4
f_8	3	5	**1**	2	4
f_9	**1**	5	3	4	2
f_{10}	5	4	**1**	3	2
Avg	3.8	**2.1**	2.3	3.3	3.5
Var	1.96	3.21	1.34	1.34	**0.72**

It is clearly observed that CMA-ES performs badly for functions $f_8 - f_{10}$; whereas CoDE and ABC are remarkably good on these functions. Hence, for a problem which we have no knowledge about, recommending CMA-ES to the user may not be a good strategy. Our proposed algorithm can remedy this by finding a portfolio consisting of more than one EA.

The covariance matrix of the ranks of the five algorithms on functions $f_1 - f_{10}$ is shown below. The entry with the smallest correlation with CMA-ES is shown in bold.

Algorithm	ABC	CMA-ES	CoDE	PSO	SaDE
ABC	1.96	-1.20	-0.49	-0.49	0.22
CMA-ES	**-1.20**	3.21	-0.70	-0.37	-0.94
CoDE	-0.49	-0.70	1.34	-0.32	0.17
PSO	-0.49	-0.37	-0.32	1.34	-0.17
SaDE	0.22	-0.94	0.17	-0.17	0.72

Based on Step 4 of the proposed algorithm, CMA-ES and ABC are chosen and constitute a MultiEA= $\{AP1, AP2\} =$ {CMA-ES, ABC}. We denote it by MultiEA1. The ranks of the six algorithms are given below.

	ABC	CMA-ES	CoDE	PSO	SaDE	MultiEA1
f_1	4	**1**	6	3	5	2
f_2	6	**1**	4	3	5	2
f_3	6	**1**	3	4	5	2
f_4	6	**1**	3	4	5	2
f_5	6	**1**	3	5	4	2
f_6	4	**1**	3	6	5	2
f_7	4	**1**	3	6	5	2
f_8	4	6	**1**	3	5	2
f_9	**1**	6	4	5	3	2
f_{10}	6	5	**1**	4	2	3
Avg	4.7	2.4	3.1	4.3	4.4	**2.1**
Var	2.68	5.16	2.10	1.34	1.16	**0.10**

Seen from the table, CMA-ES is still the algorithm with the most 1st rank test functions compared with the other algorithms. However, MultiEA1 obtains the lowest average rank and the smallest variance of ranks on functions $f_1 - f_{10}$. Thus MultiEA1 is remarkably robust on this test function set. It is easy to explain why MultiEA1 has no 1st rank case. It includes as one of the algorithms CMA-ES, and it would take time to discover CMA-ES as the algorithm to run during its execution for the cases of $f_1 - f_7$. Similar arguments hold for ABC for the case of f_9. Although MultiEA1 has no 1st rank case amongst the 10 test functions, it obtains a much more robust and consistent rank on all test functions than the other five EAs. As MultiEA1 is the algorithm with the lowest average rank (step 6), the algorithm goes to step 3 and obtains the covariance matrix below:

	ABC	CMA-ES	CoDE	PSO	SaDE	MultiEA1
ABC	2.68	-1.87	-0.63	-0.57	0.24	0.14
CMA-ES	-1.87	5.16	-1.60	-0.47	-1.51	0.29
CoDE	-0.63	-1.60	2.10	-0.14	0.51	-0.23
PSO	-0.57	-0.47	-0.14	1.34	-0.13	-0.03
SaDE	0.24	-1.51	0.51	-0.13	1.16	**-0.27**
MultiEA1	0.14	0.29	-0.23	-0.03	-0.27	0.10

Only the information in the last column is used, and MultiEA2= {MultiEA1, SaDE}={CMA-ES, ABC, SaDE}. Running this algorithm (step 5), the results are shown below. The algorithm stops and MultiEA1 is chosen (step 6).

	ABC	CMA-ES	CoDE	PSO	SaDE	Multi-EA1	Multi-EA2
f_1	4	**1**	7	3	5	2	6
f_2	7	**1**	5	4	6	2	3
f_3	7	**1**	4	5	6	2	3
f_4	7	**1**	4	5	6	2	3
f_5	7	**1**	4	6	5	2	3
f_6	5	**1**	4	7	6	2	3
f_7	5	**1**	4	7	6	2	3
f_8	5	7	**1**	3	6	2	4
f_9	**1**	7	5	6	4	2	3
f_{10}	7	6	**1**	5	2	4	3
Avg	5.5	2.7	3.9	5.1	5.2	**2.2**	3.4
Var	3.83	7.57	3.21	2.10	1.73	**0.40**	0.93

As shown in the above table, MultiEA1 attains a much more robust and stable rank on all test functions than the other six EAs. Therefore the experimental results have demonstrated that the proposed algorithm can suggest a suitable portfolio algorithm for optimization problems. If we apply MultiEA1 to an unknown problem P, we have reason to expect both an outstanding and a consistent performance.

4. CONCLUSIONS

In this paper, a novel method to compose an evolutionary algorithm (EA) portfolio is reported. We offer a solution of *when* an algorithm portfolio is useful and *when* it is not, and a solution on how to compose methodically such a portfolio. It has the advantages of being parameter-less and generic. Given a set of benchmark problems as input, it delivers a portfolio algorithm which has better and more stable rank than any individual algorithm running on the benchmark set.

5. ACKNOWLEDGMENTS

The work described in this paper was supported by a grant from CityU (7002746).

6. REFERENCES

[1] Which algorithm should I choose at any point of the search: an evolutionary portfolio approach. *GECCO 2013*.