Multipopulation Evolution Framework for Multifactorial Optimization

Genghui Li
City University of Hong Kong
Hong Kong
genghuili2-c@my.cityu.edu.hk

Qingfu Zhang
City University of Hong Kong, Hong Kong
City University of Hong Kong Shenzhen Research Institute, China
qingfu.zhang@cityu.edu.hk

Weifeng Gao
Xidian University
Xi'an, China
gaoweifeng2004@126.com

ABSTRACT

Multifactorial [1] Optimization (MFO) has been attracting considerable attention in the evolutionary computation community. In this paper, we propose a general multi-population evolution framework (MPEF) for MFO, wherein each population has its own random mating probability (*rmp*) and is used for its own task. The benefits of using MPEF are twofold: 1) Various well-developed evolutionary algorithms (EAs) can be easily embedded into MPEF for solving the task(s) of MFO problems; 2) Different populations can implement different genetic material transfers. Moreover, for instantiation, we embed a powerful differential evolution algorithm, namely SHADE, into MPEF to form a multipopulation DE algorithm (MPEF-SHADE) for solving MFO problems. The experimental results on nine MFO benchmark problems show that MPEF-SHADE is significantly better than or at least competitive with other multifactorial evolution algorithms, such as MFEA, MFDE, MFPSO and AMA.

CCS CONCEPTS

• Computing methodologies → Artificial intelligence → Search methodologies → **Heuristic function construction**

KEYWORDS

Multifactorial optimization, multipopulation evolution framework, multitasking environment, differential evolution

1 INTRODUCTION

Recently, in the community of optimization and computational intelligence, Gupta et al. [1] established a new multifactorial optimization (MFO) paradigm for evolutionary multitasking as a third category of optimization problems, MFO asks to navigate multiple search spaces that are associated with different optimization problems (or tasks) concurrently and simultaneously. However, MFEA yet not performs well in solving MFO problem when its constitutive tasks have a very low similarity (Spearman's rank correlation) [2], or the dimensions of the search space of its inter-tasks are not the same [3], or the optimums of its inter-tasks lie in different locations [3]. This paper aims to establish a multipopulation evolution framework (MPEF) so that various well-developed population-based search algorithms can be easily employed to solve the task(s) of MFO problems. Moreover, genetic material transfer can be implemented and controlled by exchanging information between populations in an effective manner. To crystalize our idea, we embed a well-developed DE method, namely SHADE [4], into our MPEF and therefore propose a multipopulation DE algorithm (MPEF-SHADE) for single objective MFO problems. The experimental results on nine single objective MFO benchmark problems show that MPEF-SHADE is able to obtain significant better solutions for most of these benchmark problems.

2 PROBLEM DEFINITION

Assume that in a MFO problem [1], K optimization tasks are need to be optimized simultaneously, and each task T_k $(k=1, 2, \cdots, K)$ is denoted by a single objective function. Without loss of generality, all tasks are supposed to be minimization problems. Mathematically, a single objective K-factorial optimization problem can be defined as follows.

$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_K\} = \arg\min\{f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_K(\mathbf{x})\} \tag{1}$$

where $f_k : \mathbf{\Omega}_k \to \mathbb{R}$ $(k=1, 2, \cdots, K)$ is a single objective function with search space $\mathbf{\Omega}_k$, $\mathbf{x}_k \in \mathbf{\Omega}_k$.

3 MPEF-SHADE

We first propose an explicit multipopulation evolution framework (MPEF) for MFO, in which each task T_k $(k=1, 2, \cdots, K)$ possesses an independent population P_k, and a particular search engine A_k is used to solve the task T_k. When use A_k to solve T_k by evolving P_k, the genetic material of other population P_r $(r=1, 2, \cdots, K, r \neq k)$ can also be exploited. Moreover, in MPEF, each task T_k has its own random mating probability rmp_k. In addition, the random mating probability rmp_k will be

---

[1] Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205761

adaptively adjusted based on the evolution status of P_k. The pseudo-code of a general MPEF for a K-factorial optimization is provided in **Algorithm 1**. Moreover, the pseudo-code of the update of rmp_k is given in **Algorithm 2**, in which tsr_k denotes the success rate of that offspring generated with genetic material transfer is better than its parent.

Algorithm 1 Basic Structure of the MPEF

Generate an initial population P_k in the unified search space for task T_k $(k=1,2,\cdots,K)$
Set $rmp_k = 0$ $(k=1,2,\cdots,K)$
Evaluate all individuals in population P_k based on the task T_k $(k=1,2,\cdots,K)$
while (stopping conditions are not satisfied) **do**
  for each task T_k
    for each individual $\mathbf{x}_i^k$ in P_k
      if $rand < rmp_k$
        Use A_k generate offspring $\mathbf{o}_i^k$ with genetic material transfer (e.g., Eq.(3))
      else
        Use A_k generate offspring $\mathbf{o}_i^k$ without genetic material transfer (e.g., Eq.(2))
      end
      Select the better one between $\mathbf{o}_i^k$ and $\mathbf{x}_i^k$
    end for
    Calculate the success rate sr_k of the population
    if $sr_k < 1/5$
      Update rmp_k (refer to **algorithm 2**)
    end
  end for
end while

Algorithm 2 Update of rmp_k

if all offspring are generated without genetic transfer
  $rmp_k = \min\{rmp_k + c\cdot(1-sr_k), 1\}$
else
  if $tsr_k > sr_k$
    $rmp_k = \min\{rmp_k + c\cdot tsr_k, 1\}$
  else
    $rmp_k = \max\{rmp_k - c\cdot(1-tsr_k), 0\}$
  end if
end if

In MPEF, we choose SHADE [4] as a search engine for all tasks. Moreover, we exploit the mutation operator of SHADE to transfer genetic material among tasks. The original mutation operator (*without genetic material transfer*) of SHADE is defined as follows.

$$\mathbf{v}_i^k = \mathbf{x}_i^k + F_i\cdot(\mathbf{x}_{pbest}^k - \mathbf{x}_i^k) + F_i\cdot(\mathbf{x}_{r1}^k - \mathbf{x}_{r2}^k) \quad (2)$$

Eq. (2) uses only the genetic material of one population to create offspring, therefore to make genetic material transfer among different populations, we devise a new mutation operator (*with genetic materials transfer*) as follows.

$$\mathbf{v}_i^k = \mathbf{x}_i^k + F_i\cdot(\mathbf{x}_{pbest}^r - \mathbf{x}_i^k) + F_i\cdot(\mathbf{x}_{r1}^r - \mathbf{x}_{r2}^r) \quad (3)$$

where r represents a randomly selected population ($r \neq k$).

4 EXPERIMENTAL RESULTS

MPEF-SHADE is compared with four state-of-the-art EAs, namely MFEA [1], MFDE [5], MFPSO [5] and AMA [6] on nine single objective MFO problems. The details of these benchmark problems can be referred to the technical report [2]. For MPEF-SHADE, the population size for each task is set to 100, learning parameter c is set to 0.3, other parameters are kept the same as those of SHADE [4]. To make a fair comparison, 100,000 total function evaluations (each task has 50,000 function evaluations) is adopted as the termination condition, and each algorithm conducts 20 independent runs on each problem. The experimental results are provided in Table 1, and the best results are highlighted in bold font. It can be seen from Table 1 that MPEF-SHADE performs better than the competitors.

Table 1: mean (rank) performance of MFEA, MFPSO, MFDE, AMA and MPEF-SHADE on all tasks

problem	Task	MFEA	MFPSO	MFDE	AMA	MPEF-SHADE
CI+HS	T1	0.3631(5)	0.21(4)	0.001(3)	2.4e-04(2)	**1.33e-09 (1)**
	T2	191.27(5)	8.11(4)	2.608(2)	6.5999(3)	**3.36e-06(1)**
CI+MS	T1	4.5495(5)	0.06(3)	0.001(2)	3.5506(4)	**3.21e-06(1)**
	T2	212.51(5)	6.26(3)	0.003(2)	181.68(4)	**3.19e-08(1)**
CI+LS	T1	20.208(3)	**5.60(1)**	21.20(5)	20.0312(2)	21.144(4)
	T2	3717.4(3)	2226.3(2)	11843(5)	**3336.0(1)**	5653.7(4)
PI+HS	T1	574.2(5)	205.73(2)	**78.26(1)**	351.99(4)	256.63(3)
	T2	9.3383(4)	3841.8(5)	2.2e-05(3)	2.9e-13(2)	**1.18e-13(1)**
PI+MS	T1	3.5657(4)	3.58(5)	0.001(2)	2.3692(3)	**3.63-05(1)**
	T2	646.18(5)	123.67(4)	60.34(3)	**19.090(1)**	48.602(2)
PI+LS	T1	20.004(5)	0.01(2)	0.46(3)	17.691(4)	**5.92e-06(1)**
	T2	20.304(5)	0.05(2)	0.22(3)	12.856(4)	**5.17e-04(1)**
NI+HS	T1	1017.8(5)	43.92(3)	89.28(4)	**14.824(1)**	42.731(2)
	T2	273.68(5)	39.58(4)	20.54(2)	**15.322(1)**	37.729(3)
NI+MS	T1	0.4264(4)	0.48(5)	2.0e-03(3)	**2.2e-11(1)**	1.68e-09(2)
	T2	27.824(5)	12.07(3)	2.97(2)	22.659(4)	**1.6382 (1)**
NI+LS	T1	621.9(5)	332.64(3)	**96.15(1)**	417.24(4)	265.93(2)
	T2	3573.9(2)	9256.2(5)	3938.2(3)	**3063.34(1)**	5982.4(4)
Ave-rank		4.44	3.28	2.78	2.50	**2.00**

REFERENCES

[1] A. Gupta, Y. S. Ong, and L. Feng. 2016. Multifactorial evolution: toward evolutionary multitasking. *IEEE Transaction on evolutionary computation* 20, 3 (2016), 343-357.

[2] B. S. Da, Y.S. Ong, L. Feng, A. K. Qin, A. Gupta, Z. X. Zhu, C. K. Ting, K. Tang, and X. Yao. 2016. Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metrics and baseline results. *Technical Report, Nanyang Technological University*, 2016.

[3] J. L Ding, C. Yang, Y.C Jin, and T.Y Chai. 2018. Generalized multi-tasking for evolutionary optimization of expensive problems. *IEEE Transaction on evolution computation*. DOI: http://dx.doi.org/10.1109/TEVC.2017.2785351

[4] R. Tanabe, and A. Fukunaga. 2013. Success-history based parameter adaptation for differential evolution. In *IEEE congress on Evolutionary Computation (CEC)*, (2013), 71-78.

[5] L. Feng, W. Zhou, L. Zhou, S.W. Jiang, J.H. Zhong, B.S. Da, Z.X. Zhu, and Y. Wang. 2017. An empirical study of multifactorial PSO and Multifactorial DE. In *IEEE congress on Evolutionary Computation (CEC),* (2017), 1658-1665.

[6] Q. J Chen, X. L Ma, Y.W. Sun, and Z.X Zhu. 2017. Adaptive memetic algorithm based evolutionary multi-tasking single-objective optimization. In *Asia-Pacific Conference on simulated Evolution and Learning (SEAL),* (2017), 462-472.