第48卷第1期 中国科学技术大学学报 Vol.48,No.1
2018年1月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Jan. 2018

文章编号:0253-2778(2018)01-0007-13

基于信息共享的组合演化算法框架

许 晗,刘伟明,李 斌

(中国科学技术大学信息科学技术学院,安徽合肥 230026)

摘要:提出一种用于集成多种启发式算法的通用框架.该框架的每个组成算法各自拥有自属种群并独立演化,因而可以保持各算法的特性和演化过程的连续性.算法间的信息交互仅通过外设的archive结构完成,即每隔一定的迁移间隔,各算法和archive之间进行一定数量的个体迁移.私有种群和信息共享的组合框架可以方便地集成现有的启发式搜索算法,具有很高的普适性.实验选取了五种算法作为子算法,共组成26个组合算法实例,测试了26个组合算法的性能,验证了基于信息共享的组合演化算法框架(EAP_IS)的有效性,并进一步将EAP_IS与其他组合框架进行了对比,实验结果表明,所提出的框架可以有效提高组成算法的综合性能.

关键词:演化算法;多算法组合;信息共享;私有种群

中图分类号:TP301 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2018.01.002

引用格式:许晗,刘伟明,李斌. 基于信息共享的组合演化算法框架[J]. 中国科学技术大学学报,2018,48(1):7-19.

XU Han, LIU Weiming, LI Bin. Evolutionary algorithm portfolios based on information sharing[J]. Journal of University of Science and Technology of China, 2018,48(1):7-19.

Evolutionary algorithm portfolios based on information sharing

XU Han, LIU Weiming, LI Bin

(*University of Science and Technology of China, School of Information Science and Technology, Hefei*, 230026)

Abstract: A general framework for combining multiple evolutionary algorithms EAP_IS is proposed. Each of the constituent algorithms in this framework has its own population to maintain its characteristic and the continuity of the evolution process. EAP_IS runs each constituent algorithm with a part of the given time budget and encourages information sharing among the constituent algorithms. The effectiveness of EAP_IS has been verified by investigating 26 instantiations of it on 25 benchmark functions, and further comparisons of EAP_IS with other combinatorial frameworks have been conducted. Experimental results show that the proposed framework can improve the performance of constituent algorithms effectively.

Key words: evolutionary algorithm; algorithm portfolios; information sharing; private population

0 引言

基于群体的演化算法被用于求解复杂的优化问题已有多年历史,并在实际工程应用中取得了良好的效果.代表性算法包括遗传算法(GA)[1]、粒子群算法(PSO)[2]、差分演化算法(DE)[3]、分布估计算

收稿日期:2017-04-14;**修回日期**:2017-08-05

基金项目:国家自然科学基金(61473001,71071045,71131002)资助.

作者简介:许晗,女,1992年生,硕士生,研究方向:演化计算. E-mail: han0115@mail.ustc.edu.cn

通讯作者:李斌,博士/教授. E-mail: binli@ustc.edu.cn

法(EDA)[4]、人工蜂群算法(ABC)[5]等.演化算法采用种群的方式组织搜索,种群中个体之间可以进行信息交互,并在搜索过程中自适应调整,具有很强的全局优化能力.

随着应用的发展,不同特性的优化问题层出不穷,演化算法的选择对最终的优化结果有很大影响.演化算法在解决不同特性问题时表现出来的性能可能会有很大的差别.因为目前尚无成熟的算法选择准则,所以人们只能借助于算法的口碑或实验方法来进行算法选择.基于口碑的选择方法存在很大的选择风险,而基于实验的选择方法往往需要消耗大量的计算资源,导致在一些实际应用中并不可行.

多算法组合(algorithm portfolios)是改进算法选择所带来的高风险和低效率的一条有效途径.一方面,通过结合不同特点的算法可以覆盖更多问题类型,因而可以提高算法的普适性,降低算法在大部分问题上的总体风险,并省去算法选择时间;另一方面,不同优化算法的组合往往会产生协同效应,使其在一些复杂问题上取得比任一子算法更好的性能.演化算法组合方法属于混合元启发式方法(hybrid metaheuristic).根据文献[6],演化算法组合方法属于高层混合方法(high-level hybird),根据组成算法的组合方法以及种群关系,又可以分为串行架构(relay)、并行架构(teamwork)以及共享种群(shared population)和私有种群(private population)方法.近年来,许多演化算法组合(algorithm portfolios)框架被陆续提出,最有代表性的有AMALGAM-SO[7],MultiEA[9],PAP[11]等.其中,AMALGAM-SO采用并行结构,共享种群,各子算法通过竞争自适应获取子代.该方法最早由Vrugt于2007年提出,并被用应用于多目标优化问题,被称为AMALGAM[8]方法.其优点是结构简单,易于实现,并能够根据子算法产生子代的适应度调整不同子算法产生的子代个数.因为不同子算法共享种群,所以存在破坏子算法自洽性的可能,不利于子算法的演进.MultiEA采用串行结构,私有种群,每一代预测各子算法未来的性能,预测性能最好的算法运行产生下一代.该方法的优点是私有种群,各子算法独立运行,因而可以方便地集成任何演化算法,但由于没有任何信息共享机制,因而存在分散投资,浪费计算资源的风险.PAP采用并行架构,私有种群,通过简单的个体迁移进行子算法之间的信息交互,迁移方式为:对每个算法,在剩余的$q-1$个算法的种群中找到最佳个体,代替自有种群中最差的个体;其后续工作EPM-PAP[12]在PAP的基础上加入了组成算法选择模块.该方法的优点是私有种群和基于个体迁移的信息共享,在保持子算法独立性的同时,达到了子算法的相互促进效果.此外,PPA[13]提出了算法组合的并行化实现方式,以节约计算时间,并应用于时间序列预测的神经网络的训练,提升预测精度.文献[14]提出一种并行算法组合,初始分配相同的适应度评估次数,运行过程中每个算法可以使用分配给自己的适应度评估次数向其他算法购买一些质量较好的解,以实现计算资源的自适应分配和解的交换.CSM[15]采用串行结构,共享种群,每一代使用多算法产生候选子代,根据候选点的密度值选择子代.

本文提出了一种基于信息共享的组合演化算法框架(evolutionary algorithm portfolios based on information sharing, EAP_IS),其基本思想是各子算法有自己的私有种群并独立演化,可以采用最好的运行参数,以最大限度地保留子算法的原有优势和优化过程的连续性;各子算法在各自独立演化的基础上通过外设的archive结构进行的信息交互.算法的独立演化可集成各种不同特性的算法,如GA、PSO、DE、ES、ABC以及一些局部搜索算法;算法间的信息交互可以帮助各子算法更快地找到更好的解,弥补各自搜索的不足.与EAP_IS相似的方法是PAP,它们均采用了私有种群和基于个体迁移的信息共享方法.与PAP不同的是,EAP_IS的个体迁移是基于外设archive的,因而可以更有效地控制共享流程并提高了共享效率;此外,并行框架也有利于实现信息的即时共享,提高演化效率,后续实验也证明了EAP_IS相对于PAP具有更好的演化性能.

1 组合演化算法框架(EAP_IS)

本文提出的组合演化算法框架主要有两个特点:①私有种群.每个子算法拥有自己的种群,各子算法只能操作自己的私有群体;②信息共享.各子算法间信息交互由外设的archive结构控制,并且信息获取是即时的.

采用私有种群的优点是:①可以最大程度保持各算法的优点和演化过程的连续性,避免复杂算法之间的互相影响;②有利于容纳更多现有算法,包括

那些具有自适应参数和记忆变量的算法，如 PSO、SaNSDE 等；③每个算法的种群规模可以根据需要设定，便于选择使子算法性能最佳的种群规模，以获得更好的解.

信息共享通过外设一个存储结构(archive)来完成，archive 中保存搜索历史中出现的质量较好的解，这些个体组成一个共享种群(shared pop，SP). SP 在搜索过程中即时更新，子算法 A_i($i=1,2,\cdots,N$)在运行到一个预先设定好的适应度评估次数(migrate FEs，MF)后，需要将自己的当前子种群共享给 archive，archive 通过一定的选择策略选择部分优胜解更新 SP.下一个运行的子算法 A_i+1 从 archive 中选择一定迁移数量(migrate number，MN)的个体替换原有私有种群中的个体，即时共享在尽可能保持各子算法优点的同时，实现了演化信息的即时共享，使各个演化算法协同合作，共同演化.

EAP_IS 框架的详细算法流程如算法 1.1 所示，首先设置一组子算法，子算法按照各自设定的参数进行初始化，形成 N 个私有种群；archive 中维持一个固定数量的共享种群 SP.每个迭代周期，依次执行各个子算法；在执行每个子算法时，先从 archive 中获取一定数量的个体取代当前种群中的个体，当子算法使用的适应度评估次数达到一定迁移间隔后，将当前群体传给 archive，archive 更新 SP；一个迭代周期结束时，检查算法终止条件，如果满足，算法停止，否则进入下一个迭代周期.

算法 1.1　EAP_IS 算法

初始化：

设定各子算法的参数，随机初始化生成各组成算法私有种群 Pop_i；

设定 archive 中共享种群大小(shared number，SN)，随机初始化生成共享种群 SP；

设置适应度评估次数计步器 FEs=0

迭代求解：

```
while (FEs <MaxFEs)
  for i=1:N(对每一个组成算法)
     从 archive 中选择 MN 个个体组成替换种群
     (Replace Pop，记为 RP)；
     RP 替换 Popi 中的 MN 个个体；
     子算法适应度评估计数器 Sub_FEs=0；
     while (Sub_FEs< MF)
       运行 Ai，更新 Popi；
       更新 Sub_FEs；
     End while
     FEs=FEs + Sub_FEs；
     子算法将 Popi 传递给 archive；
     archive 更新 SP；
  End for
End while
```

针对每一步的具体实施策略，本文设计了多种候选策略，并进行实验.

(Ⅰ)子算法传递至 archive 的共享个体 RP 的产生策略.①选择全部当前私有种群；②选择前 MN 个最优个体；③在当前种群中随机选择 MN 个个体.

(Ⅱ)Archive 接收各子算法个体后的更新策略.①与原 SP 混合后，根据适应度排序，淘汰最差的部分个体；②与原 SP 混合后，去掉重复个体后根据适应度排序，淘汰最差的部分个体；③与原 SP 混合后，为保持多样性，减少种群密集处的个体的适应度值，具体做法参考共享小生境[16]，然后根据处理后的共享适应度排序淘汰最差的部分个体.

(Ⅲ)MN 设定.①设为固定值；②按各子算法现有种群一定比例设定.共享个体选择策略：①选择最优的 MN 个；②随机选择 MN 个.

(4)私有种群中的替换策略.①替换现有种群中的最优个体；②替换现有种群中的最差个体；③替换现有种群中距离最近的个体.

经过多次实验并比较，本文选择出工作情况较好的策略如下：子算法共享全部个体，archive 接收 RP 后与原 SP 混合，去掉重复个体，淘汰最差的部分个体，MN 与各子算法私有种群大小成正比，设定替换率(replace rate，RR)，子算法接收 RP 后替换当前群体中距离最近的个体.各子算法独立执行，并且采用不同的种群规模，每个迁移间隔内，各个子算法获得相同的计算资源(即适应度评估次数).

2　实验框架

本文选用 5 个子算法进行组合实验研究，参数设置见表 1，算法选择参照文献[9].5 个算法分别是 CMA-ES[17]，SPSO[18]，SaDE[19]，ABC[5]，CoDE[20]. 5 个子算法组成两两组合示例共 10 种，三三组合示例共 10 种，包含 4 个子算法的示例 5 种以及五算法组合一个，共计 26 个.上述 5 个算法是本文选用的组成算法，其他使用本框架的研究者也可以自行选择任何合适的算法作为组成算法，本文提出的框架

可以很方便地融合各种不同类型的演化算法.在经典测试函数集 CEC2005[21] 上进行测试,CEC2005 包含 25 个测试函数,涵盖了多种问题类型,表 2 列出了 25 个函数的名称、性质和取值范围.测试问题的维度是 30 维.EAP_IS 和各子算法单独运行总适应度评估次数均为 $3.0 * 10^5$,在每个测试问题上独立运行 25 次.对组合算法和各子算法单独执行的结果进行比较和分析,验证组合算法是否能降低在不同类型问题上失败的风险,同时还对本文框架和文献[9]中提到的其他框架的结果进行对比,为确保比较的公平性,本文实验使用的子算法与参数均和文献[9]相同.

表 1 实验参数

Tab.1 Experimental parameters

参数名	数值
维度 D	30
最大适应度评估次数 MaxFEs	$3.0 * 10^5$
组成算法及种群大小	1) CMA-ES, NP=14 2)SPSO2011, NP=40 3) SaDE, NP=50 4) ABC, NP=62 5)CoDE, NP=30
archive 中共享种群大小	20
替换率 RR	0.1
迁移间隔 MF	200
运行次数(runs)	25

表 2 测试函数集(CEC2005[21])

Tab.2 Bench mark function set

性质		函数	范围
单峰函数		F_1: Shifted sphere function	[−100,100]
		F_2: Shifted schwefel's problem 1.2	[−100,100]
		F_3: Shifted rotated high conditioned elliptic function	[−100,100]
		F_4: Shifted schwefel's problem 1.2 with noise in fitness	[−100,100]
		F_5: Schwefel's problem 2.6 with global optimum on bounds	[−100,100]
多峰函数	基本函数	F_6: Shifted rosenbrock's function	[−100,100]
		F_7: Shifted rotated griewank's function without bounds	[0,600]
		F_8: Shifted rotated ackley's function with global optimum on bounds	[−32,32]
		F_9: Shifted rastrigin's function	[−5,5]
		F_{10}: Shifted rotated rastrigin's function	[−5,5]
		F_{11}: Shifted rotated weierstrass function	[−0.5,0.5]
		F_{12}: Schwefel's problem 2.13	[−π,π]
	扩展函数	F_{13}: Expanded extended griewank's plus rosenbrock's function (F_8F_2)	[−3,1]
		F_{14}: Shifted rotated expanded scaffer's F_6	[−100,100]
	混合组合函数	F_{15}: Hybrid composition function	[−5,5]
		F_{16}: Rotated hybrid composition function	[−5,5]
		F_{17}: Rotated hybrid composition function with noise in fitness	[−5,5]
		F_{18}: Rotated hybrid composition function	[−5,5]
		F_{19}: Rotated hybrid composition function with a narrow basin for the global optimum	[−5,5]
		F_{20}: Rotated hybrid composition function with the global optimum on the bounds	[−5,5]
		F_{21}: Rotated hybrid composition function	[−5,5]
		F_{22}: Rotated hybrid composition function with high condition number matrix	[−5,5]
		F_{23}: Non-continuous rotated hybrid composition function	[−5,5]
		F_{24}: Rotated hybrid composition function	[−5,5]
		F_{25}: Rotated hybrid composition function without bounds	[2,5]

3　实验结果

3.1　组合框架和子算法单独运行的结果与分析

为了验证 EAP_IS 能有效提升组成算法找到的解的质量，降低在大部分问题上的总体风险，对 5 个子算法组成的 26 种算法实例与 5 个子算法单独运行获得的结果进行统计，表 3 给出每个算法在 CEC2005 测试函数集上运行 25 次得到的最优解距离实际最优解误差的均值，并对得到的均值进行排名，每个测试函数上排名第一的算法均值加粗显示.

表 3　组合框架和子算法单独运行的实验结果

Tab。3　Experimental results of EAP_IS and sub-algorithms

algorithm	F_1		F_2		F_3		F_4		F_5	
	mean	rank	mean	rank	mean	rank	mean	rank	mean	rank
SPSO	4.27E−28	25	6.80E−24	18	2.15E+05	28	8.69E+01	27	1.21E+03	29
SaDE	**0.00E+00**	**1**	7.74E−21	19	3.55E+04	17	2.47E+00	24	2.81E+02	24
ABC	5.12E−16	31	2.54E+03	31	6.10E+06	31	2.12E+04	31	6.18E+03	31
CMA	3.53E−27	29	1.78E−26	15	1.25E−22	9	5.74E+03	30	5.21E+01	10
CoDE	2.02E−30	16	1.31E−17	20	6.61E+04	20	4.40E−03	19	2.77E+02	22
SPSO_SaDE	**0.00E+00**	**1**	2.37E−25	17	6.34E+04	19	3.51E−04	15	2.78E+02	23
SPSO_ABC	5.58E−28	28	1.97E−10	27	5.14E+05	30	1.04E+03	29	1.45E+03	30
SPSO_CMA	5.43E−28	27	3.13E−27	12	1.12E−22	6	1.73E−08	13	5.27E+01	11
SPSO_CoDE	**0.00E+00**	**1**	7.17E−17	21	1.61E+05	24	4.29E−04	16	2.24E+02	21
SaDE_ABC	**0.00E+00**	**1**	4.07E−08	29	1.63E+05	25	1.36E+02	28	4.07E+02	28
SaDE_CMA	2.02E−30	16	2.42E−27	11	7.57E−23	2	4.08E−27	6	2.46E+01	4
SaDE_CoDE	**0.00E+00**	**1**	2.52E−16	22	5.21E+04	18	2.54E−03	17	1.42E+02	16
ABC_CMA	3.84E−27	30	2.17E−26	16	1.57E−22	11	7.22E+01	26	1.70E+02	17
ABC_CoDE	**0.00E+00**	**1**	5.85E−08	30	2.09E+05	27	1.40E+01	25	3.84E+02	27
CMA_CoDE	8.08E−30	22	1.19E−27	8	**5.43E−23**	**1**	1.19E−27	2	2.20E+01	3
SPSO_SaDE_ABC	**0.00E+00**	**1**	7.71E−15	24	1.54E+05	23	1.88E−01	22	3.65E+02	26
SPSO_SaDE_CMA	**0.00E+00**	**1**	2.17E−27	10	1.23E−22	8	2.97E−27	4	**1.74E+01**	**1**
SPSO_SaDE_CoDE	**0.00E+00**	**1**	3.88E−16	23	1.01E+05	21	4.14E−03	18	1.78E+02	18
SPSO_ABC_CMA	5.24E−28	26	3.66E−27	14	1.44E−22	10	6.50E−05	14	7.40E+01	14
SPSO_ABC_CoDE	2.02E−30	16	8.26E−11	26	2.63E+05	29	1.26E−01	21	2.92E+02	25
SPSO_CMA_CoDE	2.02E−29	24	9.90E−28	4	9.48E−23	5	2.92E−27	3	3.42E+01	6
SaDE_ABC_CMA	2.02E−30	16	3.32E−27	13	1.15E−22	7	8.08E−27	10	2.56E+01	5
SaDE_ABC_CoDE	**0.00E+00**	**1**	1.62E−09	28	1.16E+05	22	3.25E−01	23	1.85E+02	19
SaDE_CMA_CoDE	**0.00E+00**	**1**	**7.49E−28**	1	9.44E−23	4	**9.21E−28**	**1**	2.16E+01	2
ABC_CMA_CoDE	2.02E−30	16	1.00E−27	5	7.98E−23	3	6.21E−27	8	1.14E+02	15
SPSO_SaDE_ABC_CMA	4.04E−30	21	1.90E−27	9	2.05E−22	12	6.69E−27	9	4.87E+01	9
SPSO_SaDE_ABC_CoDE	**0.00E+00**	**1**	6.28E−11	25	1.69E+05	26	5.52E−02	20	2.04E+02	20
SPSO_SaDE_CMA_CoDE	**0.00E+00**	**1**	1.08E−27	6	2.20E−22	13	3.00E−27	5	3.80E+01	7
SPSO_ABC_CMA_CoDE	1.62E−29	23	1.14E−27	7	3.04E−22	15	1.28E−26	12	4.67E+01	8

SaDE_ABC_CMA_CoDE	**0.00E+00**	**1**	8.04E−28	3	2.59E−22	14	4.62E−27	7	7.13E+01	13
SPSO_SaDE_ABC_CMA_CoDE	**0.00E+00**	**1**	8.00E−28	2	6.61E−10	16	1.05E−26	11	6.49E+01	12

algorithm	F_6		F_7		F_8		F_9		F_{10}	
	mean	rank	mean	rank	mean	rank	mean	rank	mean	rank
SPSO	4.08E+02	31	2.90E−02	30	2.04E+01	23	6.98E+01	31	8.20E+01	28
SaDE	1.62E+00	21	2.22E−02	27	2.08E+01	31	3.58E−01	19	5.00E+01	11
ABC	1.79E+00	24	1.19E−01	31	2.07E+01	29	1.64E−17	16	2.68E+02	31
CMA	6.38E−01	18	2.27E−03	6	2.00E+01	1	4.58E+01	30	4.59E+01	10
CoDE	4.78E−01	16	1.12E−02	18	2.03E+01	19	3.58E−01	19	4.38E+01	9
SPSO_SaDE	1.98E+00	26	1.85E−02	23	2.04E+01	22	1.83E+00	24	6.25E+01	20
SPSO_ABC	2.90E+00	27	2.06E−02	26	2.05E+01	27	4.62E−14	17	8.84E+01	30
SPSO_CMA	1.59E−01	10	1.77E−03	4	2.00E+01	1	4.09E+01	29	4.01E+01	4
SPSO_CoDE	1.74E+00	23	1.96E−02	25	2.00E+01	11	1.23E+00	23	6.33E+01	21
SaDE_ABC	6.54E−02	9	1.88E−02	24	2.08E+01	30	7.89E−33	4	7.69E+01	25
SaDE_CMA	**3.56E−25**	**1**	**1.48E−03**	**1**	2.05E+01	25	1.95E+00	25	**3.35E+01**	**1**
SaDE_CoDE	1.59E+00	20	1.62E−02	22	2.02E+01	18	**0.00E+00**	**1**	4.09E+01	8
ABC_CMA	1.82E−24	5	2.37E−03	7	2.00E+01	10	1.20E−29	12	5.77E+01	18
ABC_CoDE	1.00E+00	19	1.06E−02	17	2.03E+01	21	1.97E−33	3	7.80E+01	26
CMA_CoDE	1.59E−01	10	3.64E−03	13	2.00E+01	1	2.43E+00	26	4.02E+01	5
SPSO_SaDE_ABC	1.33E+01	28	1.44E−02	20	2.05E+01	26	3.89E−31	8	6.82E+01	22
SPSO_SaDE_CMA	1.59E−01	10	1.68E−03	3	2.00E+01	1	4.50E+00	28	4.09E+01	7
SPSO_SaDE_CoDE	1.98E+00	25	2.23E−02	28	2.02E+01	16	3.58E−01	19	7.04E+01	24
SPSO_ABC_CMA	9.66E−25	4	2.07E−03	5	2.00E+01	1	2.54E−30	9	5.66E+01	17
SPSO_ABC_CoDE	1.45E+01	29	1.53E−02	21	2.03E+01	20	1.84E−24	14	8.54E+01	29
SPSO_CMA_CoDE	3.19E−01	13	4.83E−03	16	**2.00E+01**	**1**	4.10E+00	27	3.89E+01	3
SaDE_ABC_CMA	4.51E−25	2	2.96E−03	10	2.01E+01	15	0.00E+00	1	5.16E+01	14
SaDE_ABC_CoDE	1.65E+00	22	1.39E−02	19	2.06E+01	28	1.89E−28	13	6.97E+01	23
SaDE_CMA_CoDE	4.78E−01	15	2.46E−03	8	2.01E+01	14	1.99E−01	18	3.82E+01	2
ABC_CMA_CoDE	6.76E−25	3	4.24E−03	14	2.00E+01	13	3.35E−32	5	5.48E+01	15
SPSO_SaDE_ABC_CMA	3.87E−02	8	3.55E−03	11	2.00E+01	1	4.49E−32	6	5.08E+01	13
SPSO_SaDE_ABC_CoDE	2.29E+01	30	2.32E−02	29	2.04E+01	24	3.39E−21	15	7.89E+01	27
SPSO_SaDE_CMA_CoDE	4.80E−01	17	2.66E−03	9	2.00E+01	1	7.96E−01	22	4.03E+01	6
SPSO_ABC_CMA_CoDE	2.04E−05	7	4.43E−03	15	2.00E+01	12	3.51E−31	7	5.54E+01	16
SaDE_ABC_CMA_CoDE	1.05E−05	6	3.64E−03	12	2.02E+01	17	2.93E−30	10	5.04E+01	12
SPSO_SaDE_ABC_CMA_CoDE	3.51E−01	14	1.68E−03	2	2.00E+01	9	2.62E−30	11	5.77E+01	19

algorithm	F_{11}		F_{12}		F_{13}		F_{14}		F_{15}	
	mean	rank	mean	rank	mean	rank	mean	rank	mean	rank
SPSO	2.52E+01	30	6.34E+03	29	4.497345	31	**11.25927**	**1**	4.33E+02	31
SaDE	1.78E+01	22	1.74E+03	14	1.562736	18	12.00633	11	3.54E+02	24
ABC	2.56E+01	31	7.35E+03	31	1.169613	7	12.81506	30	2.94E−04	12
CMA	**6.74E+00**	**1**	**5.66E+02**	**1**	3.229087	29	13.43923	31	3.25E+02	18
CoDE	1.47E+01	18	3.41E+03	25	1.865362	23	12.49381	25	3.80E+02	28
SPSO_SaDE	1.99E+01	27	1.90E+03	15	1.680181	19	11.55425	2	3.47E+02	22
SPSO_ABC	2.47E+01	29	7.10E+03	30	1.121392	5	11.70454	3	7.17E−06	11
SPSO_CMA	8.46E+00	13	1.73E+03	13	3.652192	30	11.92933	7	3.73E+02	26
SPSO_CoDE	1.64E+01	20	5.56E+03	28	2.353207	26	11.81859	4	3.74E+02	27
SaDE_ABC	1.92E+01	26	2.81E+03	22	1.153794	6	12.42578	23	1.89E−16	4
SaDE_CMA	8.38E+00	10	1.11E+03	7	1.739112	20	12.10228	12	2.84E+02	17
SaDE_CoDE	1.26E+01	17	1.12E+03	8	1.345137	17	12.20761	17	4.17E+02	30
ABC_CMA	9.04E+00	15	3.36E+03	24	1.074399	2	12.75928	29	1.89E−16	4
ABC_CoDE	1.67E+01	21	3.22E+03	23	**1.034789**	**1**	12.51808	26	5.75E−13	6
CMA_CoDE	8.23E+00	8	8.11E+02	2	2.872344	28	12.25569	19	3.40E+02	21
SPSO_SaDE_ABC	2.09E+01	28	3.54E+03	26	1.221003	12	11.8193	5	5.08E−09	9
SPSO_SaDE_CMA	8.13E+00	7	1.60E+03	12	1.904056	24	11.95532	9	3.25E+02	19
SPSO_SaDE_CoDE	1.56E+01	19	2.34E+03	18	1.740484	21	11.92133	6	4.08E+02	29
SPSO_ABC_CMA	9.22E+00	16	9.18E+02	3	1.192579	10	12.33192	21	**0.00E+00**	**1**
SPSO_ABC_CoDE	1.83E+01	24	4.37E+03	27	1.232325	15	11.94047	8	1.08E−09	7
SPSO_CMA_CoDE	7.97E+00	6	2.48E+03	19	2.673532	27	12.16079	14	3.32E+02	20
SaDE_ABC_CMA	7.14E+00	3	1.17E+03	9	1.084371	3	12.42267	22	0.00E+00	1
SaDE_ABC_CoDE	1.81E+01	23	2.53E+03	20	1.221979	13	12.44719	24	1.00E−07	10
SaDE_CMA_CoDE	8.35E+00	9	1.07E+03	5	1.816694	22	12.23856	18	3.47E+02	23
ABC_CMA_CoDE	8.43E+00	12	1.28E+03	10	1.102428	4	12.55397	28	3.59E+02	13
SPSO_SaDE_ABC_CMA	7.25E+00	4	2.19E+03	17	1.172896	8	12.15274	13	0.00E+00	1
SPSO_SaDE_ABC_CoDE	1.91E+01	25	2.07E+03	16	1.282116	16	11.95836	10	2.08E−12	8
SPSO_SaDE_CMA_CoDE	7.57E+00	5	1.57E+03	11	2.02569	25	12.16337	15	2.34E−05	25
SPSO_ABC_CMA_CoDE	8.43E+00	11	1.05E+03	4	1.210365	11	12.27681	20	2.85E−01	16
SaDE_ABC_CMA_CoDE	6.97E+00	2	1.08E+03	6	1.189963	9	12.5437	27	3.45E+02	14
SPSO_SaDE_ABC_CMA_CoDE	9.03E+00	14	2.66E+03	21	1.225836	14	12.20153	16	5.18E+00	15

algorithm	F_{16}		F_{17}		F_{18}		F_{19}		F_{20}	
	mean	rank	mean	rank	mean	rank	mean	rank	mean	rank
SPSO	1.43E+02	26	1.98E+02	29	9.02E+02	31	9.26E+02	31	9.19E+02	30
SaDE	1.02E+02	5	9.87E+01	5	8.85E+02	21	8.86E+02	17	8.77E+02	17

ABC	2.73E+02	31	3.27E+02	31	8.16E+02	18	9.10E+02	29	9.15E+02	29
CMA	2.67E+02	30	2.35E+02	30	9.05E+02	16	8.16E+02	10	8.16E+02	11
CoDE	1.13E+02	13	1.02E+02	6	9.16E+02	25	9.06E+02	25	9.05E+02	24
SPSO_SaDE	1.09E+02	10	1.50E+02	19	9.15E+02	30	9.12E+02	30	9.23E+02	31
SPSO_ABC	1.23E+02	18	1.77E+02	27	8.16E+02	29	9.07E+02	27	9.06E+02	27
SPSO_CMA	1.66E+02	29	1.67E+02	24	9.02E+02	10	8.16E+02	16	8.16E+02	16
SPSO_CoDE	1.04E+02	8	1.28E+02	13	8.84E+02	23	8.98E+02	21	9.07E+02	28
SaDE_ABC	9.70E+01	2	1.27E+02	12	8.13E+02	17	9.07E+02	28	8.89E+02	19
SaDE_CMA	1.64E+02	28	1.67E+02	25	8.97E+02	2	8.15E+02	7	8.12E+02	2
SaDE_CoDE	**6.70E+01**	**1**	**8.06E+01**	**1**	8.16E+02	19	8.89E+02	18	9.06E+02	25
ABC_CMA	1.29E+02	22	1.97E+02	28	9.05E+02	14	8.16E+02	9	**7.99E+02**	**1**
ABC_CoDE	1.15E+02	15	1.19E+02	9	8.15E+02	24	9.02E+02	22	9.06E+02	26
CMA_CoDE	1.43E+02	25	1.51E+02	20	9.13E+02	9	8.16E+02	12	8.16E+02	13
SPSO_SaDE_ABC	1.30E+02	23	1.61E+02	22	8.14E+02	28	8.97E+02	20	9.02E+02	22
SPSO_SaDE_CMA	1.32E+02	24	1.75E+02	26	8.98E+02	5	8.14E+02	5	8.15E+02	7
SPSO_SaDE_CoDE	1.04E+02	7	9.56E+01	3	8.16E+02	20	9.06E+02	26	9.02E+02	23
SPSO_ABC_CMA	1.08E+02	9	1.40E+02	16	9.07E+02	11	8.16E+02	11	8.15E+02	10
SPSO_ABC_CoDE	1.27E+02	20	1.17E+02	7	8.16E+02	26	9.02E+02	24	9.02E+02	21
SPSO_CMA_CoDE	1.48E+02	27	1.60E+02	21	8.14E+02	13	8.16E+02	15	8.16E+02	15
SaDE_ABC_CMA	9.91E+01	3	1.39E+02	15	9.02E+02	7	8.15E+02	6	8.13E+02	4
SaDE_ABC_CoDE	1.15E+02	14	9.44E+01	2	8.13E+02	22	9.02E+02	23	8.82E+02	18
SaDE_CMA_CoDE	1.25E+02	19	1.36E+02	14	8.16E+02	3	8.15E+02	8	8.14E+02	5
ABC_CMA_CoDE	1.02E+02	4	9.58E+01	4	8.15E+02	15	8.16E+02	14	8.16E+02	12
SPSO_SaDE_ABC_CMA	1.29E+02	21	1.49E+02	18	9.07E+02	8	**8.10E+02**	**1**	8.13E+02	3
SPSO_SaDE_ABC_CoDE	1.16E+02	16	1.25E+02	11	8.14E+02	27	8.94E+02	19	8.93E+02	20
SPSO_SaDE_CMA_CoDE	1.23E+02	17	1.24E+02	10	8.16E+02	6	8.14E+02	4	8.15E+02	8
SPSO_ABC_CMA_CoDE	1.12E+02	11	1.67E+02	23	8.12E+02	12	8.16E+02	13	8.16E+02	14
SaDE_ABC_CMA_CoDE	1.13E+02	12	1.18E+02	8	**8.13E+02**	**1**	8.13E+02	2	8.15E+02	9
SPSO_SaDE_ABC_CMA_CoDE	1.02E+02	6	1.44E+02	17	9.02E+02	4	8.14E+02	3	8.14E+02	6

algorithm	F_{21}		F_{22}		F_{23}		F_{24}		F_{25}	
	mean	rank	mean	rank	mean	rank	mean	rank	mean	rank
SPSO	6.59E+02	30	9.21E+02	27	7.14E+02	30	**2.00E+02**	**1**	2.16E+02	29
SaDE	5.38E+02	28	9.08E+02	25	5.62E+02	28	**2.00E+02**	**1**	2.14E+02	27
ABC	4.89E+02	2	1.02E+03	31	**5.33E+02**	**1**	3.14E+02	31	5.18E+02	31
CMA	8.62E+02	31	5.43E+02	16	8.70E+02	31	2.11E+02	28	2.48E+02	30
CoDE	5.00E+02	11	8.77E+02	18	5.34E+02	18	**2.00E+02**	**1**	2.11E+02	17
SPSO_SaDE	5.40E+02	29	9.14E+02	26	5.92E+02	29	**2.00E+02**	**1**	2.11E+02	21

SPSO_ABC	**4.83E+02**	**1**	9.38E+02	30	5.34E+02	25	**2.00E+02**	**1**	2.16E+02	28
SPSO_CMA	5.00E+02	11	5.18E+02	15	5.37E+02	26	2.11E+02	27	2.11E+02	20
SPSO_CoDE	5.12E+02	27	8.87E+02	21	5.34E+02	16	**2.00E+02**	**1**	2.11E+02	13
SaDE_ABC	4.93E+02	3	9.33E+02	29	5.33E+02	3	**2.00E+02**	**1**	2.13E+02	26
SaDE_CMA	5.00E+02	11	5.17E+02	13	5.34E+02	17	**2.00E+02**	**1**	2.10E+02	9
SaDE_CoDE	5.00E+02	11	8.73E+02	17	5.34E+02	10	**2.00E+02**	**1**	2.11E+02	15
ABC_CMA	4.97E+02	9	**5.03E+02**	**1**	5.33E+02	4	2.12E+02	30	2.12E+02	22
ABC_CoDE	4.96E+02	6	8.81E+02	19	5.34E+02	19	**2.00E+02**	**1**	2.12E+02	25
CMA_CoDE	5.00E+02	11	5.05E+02	6	5.34E+02	24	**2.00E+02**	**1**	2.10E+02	4
SPSO_SaDE_ABC	5.00E+02	11	9.22E+02	28	5.34E+02	11	**2.00E+02**	**1**	2.12E+02	23
SPSO_SaDE_CMA	5.00E+02	11	5.06E+02	8	5.34E+02	15	**2.00E+02**	**1**	2.10E+02	12
SPSO_SaDE_CoDE	5.00E+02	11	8.83E+02	20	5.50E+02	27	**2.00E+02**	**1**	2.11E+02	14
SPSO_ABC_CMA	5.00E+02	11	5.04E+02	2	5.34E+02	20	2.12E+02	29	2.12E+02	24
SPSO_ABC_CoDE	5.00E+02	11	8.89E+02	22	5.34E+02	23	**2.00E+02**	**1**	2.11E+02	18
SPSO_CMA_CoDE	5.00E+02	11	5.17E+02	14	5.34E+02	21	**2.00E+02**	**1**	2.10E+02	7
SaDE_ABC_CMA	4.96E+02	7	5.06E+02	11	5.34E+02	14	**2.00E+02**	**1**	2.10E+02	10
SaDE_ABC_CoDE	5.00E+02	11	8.91E+02	23	5.34E+02	12	**2.00E+02**	**1**	2.11E+02	19
SaDE_CMA_CoDE	5.00E+02	11	5.05E+02	5	5.34E+02	9	**2.00E+02**	**1**	2.10E+02	1
ABC_CMA_CoDE	4.97E+02	10	5.06E+02	10	5.34E+02	22	**2.00E+02**	**1**	2.10E+02	5
SPSO_SaDE_ABC_CMA	4.96E+02	8	5.07E+02	12	5.34E+02	8	**2.00E+02**	**1**	2.10E+02	11
SPSO_SaDE_ABC_CoDE	4.93E+02	5	8.96E+02	24	5.34E+02	5	**2.00E+02**	**1**	2.11E+02	16
SPSO_SaDE_CMA_CoDE	5.00E+02	11	5.05E+02	7	5.34E+02	7	**2.00E+02**	**1**	2.10E+02	3
SPSO_ABC_CMA_CoDE	5.00E+02	11	5.06E+02	9	5.34E+02	13	**2.00E+02**	**1**	2.10E+02	6
SaDE_ABC_CMA_CoDE	4.93E+02	4	5.04E+02	3	5.33E+02	2	**2.00E+02**	**1**	2.10E+02	2
SPSO_SaDE_ABC_CMA_CoDE	5.00E+02	11	5.04E+02	4	5.34E+02	6	**2.00E+02**	**1**	2.10E+02	8

为了更直观地比较各算法的性能,需要对上述实验结果再进行进一步的统计分析,比较的标准如下:

(Ⅰ)各算法在所有测试函数下的排名的均值,记为 average rank.

(Ⅱ)基于 Wilcoxon 符号秩检验[22]的 nWins[23],nWins 统计方法适用于对算法在每个函数上的实验结果进行全局比较分析.该方法通过使用置信区间为 0.05 的威尔科克森符号秩和检验(Wilcoxon signed rank)对算法进行两两比较.如果一个算法明显优于其他算法(即 $p<0.05$),获胜算法+1,失败算法−1,若 $p>0.05$,Z 则认为这两个算法没有明显的统计学上的优胜方.完成全部比较后得到累加的 nWins 值.nWins 值越大表明该算法相对于其他算法的优势越大.

首先我们统计了全部 25 个测试函数上的平均排名和 nWins;然后根据测试函数特性对算法进行分类统计,以分析组合算法在不同类型问题上的组合效率.测试函数特性类别如下:

单峰:F_1～F_5;

多峰:F_6～F_{25};

旋转:F_3,F_7,F_8,F_{10},F_{11},F_{14},F_{16}～F_{25};

最优解在边界:F_5,F_8,F_{20},F_{22},F_{23},F_{25}.

统计结果见表 4.平均排名前三的算法组合加粗显示.

表 4 算法框架平均排名和 nWins 统计

Tab.4 Average rank and nWins

algorithm	全部		单峰		多峰		旋转		最优值在边界	
	average rank	nWins	average rank	nWins	average rank	nWins	average rank	nWins	average rank	nWins
SPSO	26.24	−27	25.4	−11	26.45	−28	25.38	−27	27.67	−24
SaDE	18.68	−11	19	2	18.6	−10	18.94	−10	25.33	−14
ABC	25.24	−29	31	−30	23.8	−27	26.06	−29	25.33	−24
CMA	19.52	−14	22	−5	18.9	−15	18.00	−14	19.83	−6
CoDE	18	−11	18.4	−3	17.9	−10	16.56	−6	19.17	−8
SPSO_SaDE	19.88	−15	14.2	4	21.3	−18	21.44	−18	24.00	−21
SPSO_ABC	22.28	−25	27.8	−23	20.9	−17	22.13	−25	27.67	−23
SPSO_CMA	16	−2	13.8	−1	16.55	−10	14.19	−2	15.17	−3
SPSO_CoDE	18.32	−13	15.8	3	18.95	−12	17.31	−10	17.50	−6
SaDE_ABC	16.92	−12	22	1	15.65	−1	18.25	−10	22.50	−14
SaDE_CMA	10.96	16	8	7	11.7	8	10.31	15	12.00	14
SaDE_CoDE	14.2	1	15.8	3	13.8	8	13.88	5	17.00	−6
ABC_CMA	14.84	6	21	−8	13.3	4	14.13	7	10.67	13
ABC_CoDE	17.08	−11	19.6	1	16.45	−2	18.38	−13	22.50	−13
CMA_CoDE	12.2	16	9.4	7	12.9	10	11.50	15	8.33	17
SPSO_SaDE_ABC	18.48	−15	17.8	2	18.65	−14	19.38	−15	22.00	−13
SPSO_SaDE_CMA	10.24	17	**4.4**	**8**	11.7	7	9.00	16	7.83	17
SPSO_SaDE_CoDE	17.8	−11	17.6	3	17.85	−10	17.06	−10	19.67	−13
SPSO_ABC_CMA	12.12	15	14.6	−3	11.5	14	13.00	9	11.83	8
SPSO_ABC_CoDE	18.88	−15	21	−8	18.35	−10	18.50	−15	20.83	−9
SPSO_CMA_CoDE	13.64	8	10	6	14.55	2	12.56	12	10.17	15
SaDE_ABC_CMA	**8.68**	**23**	11.8	5	**7.9**	**22**	9.50	19	10.67	17
SaDE_ABC_CoDE	17.36	−8	18.8	1	17	−4	17.56	−9	20.50	−8
SaDE_CMA_CoDE	**8.88**	**19**	**2.4**	**13**	10.5	14	8.44	21	**5.83**	**20**
ABC_CMA_CoDE	10.6	15	10.2	4	10.7	16	12.06	15	11.67	11
SPSO_SaDE_ABC_CMA	9.28	19	11.8	2	**8.65**	**17**	9.00	19	7.33	17
SPSO_SaDE_ABC_CoDE	17.56	−8	19	3	17.2	−2	18.00	−10	18.17	−7
SPSO_SaDE_CMA_CoDE	9.44	17	**5.2**	**6**	10.5	14	**7.44**	**21**	**5.17**	**19**
SPSO_ABC_CMA_CoDE	12.04	15	13.8	1	11.6	14	12.63	8	11.00	13
SaDE_ABC_CMA_CoDE	**7.56**	**26**	6	5	**7.95**	**27**	**7.88**	**23**	**6.67**	**16**
SPSO_SaDE_ABC_CMA_CoDE	9.12	14	5.4	5	10.05	13	**8.00**	**18**	7.33	15

根据表 4 的统计结果来看，大多数情况下，无论是均值排名还是 nWins，EAP_IS 组合算法实例的总体性能优于其组成算法.多算法组合表现出更优越的性能和更小的风险，可以在大部分问题上取得

不错的结果.

从全部 25 个测试问题的排名上看,我们发现 CMA 单独执行时性能一般,但有 CMA 参与的组合均获得了不错的性能,综合排名前 10 的框架里均有 CMA 参与结合,说明 CMA 对其他几个子算法有良好的互补作用.CoDE 是 5 个子算法中综合性能最好的,但 CoDE 参与组合后并不会像 CMA 一样对其他组成算法有非常大的促进作用.我们也观察到仍有少数情况下组合算法的性能弱于其子算法单独执行,如 SPSO_SaDE 的性能虽优于 SPSO,但弱于其子算法 SaDE,这表示 SPSO 对 SaDE 没有协同效应.同时我们也观察到 SaDE_CMA,SaDE 和 CMA 两个算法性能均弱于 CoDE,但组合后的性能比 5 个子算法中性能最佳的 CoDE 更好,组合算法在没有利用 CoDE 中任何算子的情况下超过了 CoDE,这两个算法之间有很好的协同作用,证明一些相对较弱的算法组合后可以比现有的较强的算法具有更好的性能.

单峰问题相对简单,收敛速度快的算法会有更好的优势,我们注意到两种 DE 算法在单峰函数上有比较好的表现,组合算法中排名靠前的均有 DE 算法的参与,DE 算法本身快速收敛的能力使得组合算法可以很快找到最优解.

多峰问题比单峰问题复杂,有大量的局部极值,可以更明显地体现出混合框架的优势,各子算法单独执行时的排名都相对靠后.多峰问题上平均排名最好的是 SaDE_ABC_CMA,但综合考虑 average rank 和 nWins,表现最好的算法组合应为 SaDE_ABC_CMA_CoDE.我们看到 DE 算法在多峰问题上参与组合的表现不如在单峰问题上那么优越,本身性能稍弱的 CMA_ES、ABC 的协同作用开始发挥更重要的作用.

对于旋转和最优值在边界上的问题,CMA 体现出了非常明显的协同优势,有 CMA 参与的所有组合性能都优于没有 CMA 参与的算法组合,同时也优于 CMA 单独运行,说明如果选择到了合适的子算法,算法组合可以大大增加解决问题的成功率.

3.2　与其他组合框架对比

前文比较了混合框架和组成算法之间的性能,验证了本文提出的框架相对于单一子算法有更优越的性能和更小的风险.本节将 EAP_IS 与其他群体智能算法混合框架进行对比.本文选用文献[9]中总结的几种混合进化框架的实验结果作比较.用于对比的三个框架为 AMALGAM-GO、MulitiEA、PAP,这三种框架包含有共享种群/私有种群、串行结构/并行结构、有自适应/无自适应、合作机制/竞争机制等不同的策略,较为全面地包含了现有组合框架中的多种策略机制.

三个对比框架均使用了全部 5 个组成算法(CMA-ES, PSO, SaDE, ABC, CoDE),本文也将使用全部 5 个组成算法的 EAP_IS 得到的实验结果与之对比,记为 EAP_IS 1.根据表 4 的统计结果,组合 SaDE_ABC_CMA_CoDE 的综合性能最好,记为 EAP_IS 2.

为确保对比实验的公平性,本文选取的实验参数均与文献[9]中使用的参数保持一致.表 5 是 EAP_IS 组合框架和其他组合框架对比结果排名统计.其中,3 个对比算法的数据均来自文献[9].表格给出每个算法在各测试函数上运行 25 次得到的解距离实际最优解误差的均值(mean),并进行排名(r),最后统计了各算法在全部测试问题上排名的均值.

表 5　各种组合框架对比结果

Tab.5　Experimental results of EAP_IS and other algorithm portfolios

algorithm	F_1		F_2		F_3		F_4		F_5		F_6		F_7	
	mean	r	mean	r	mean	r	mean	r	mean	r	mean	r	mean	r
EAP_IS 2	0.00E+00	1	8.04E-28	2	2.59E-22	1	4.62E-27	1	7.13E+01	3	1.05E-05	2	3.64E-03	3
EAP_IS 1	0.00E+00	1	8.00E-28	1	6.61E-10	3	1.05E-26	2	6.49E+01	2	3.51E-01	5	1.68E-03	2
AMALGAM-SO[9]	8.38E-15	5	4.82E-15	5	9.47E-14	2	8.53E+01	5	9.72E-05	1	9.04E-15	1	3.94E-15	1
MultiEA[9]	0.00E+00	1	3.95E-24	4	8.82E+04	5	5.18E-02	3	1.01E+03	4	3.26E-01	3	6.90E-03	4
PAP[9]	3.89E-29	4	9.83E-26	3	4.06E+04	4	2.36E+00	4	1.33E+03	5	3.26E-01	3	8.56E-03	5

algorithm	F_8		F_9		F_{10}		F_{11}		F_{12}		F_{13}		F_{14}	
	mean	r	mean	r	mean	r	mean	r	mean	r	mean	r	mean	r
EAP_IS 2	2.02E+01	2	2.93E-30	3	5.04E+01	4	6.97E+00	2	1.08E+03	3	1.19E+00	2	1.25E+01	4
EAP_IS 1	2.00E+01	1	3.23E-30	4	5.77E+01	5	9.03E+00	3	2.66E+03	4	1.23E+00	3	1.22E+01	2
AMALGAM-SO[9]	2.10E+01	5	5.05E+00	5	7.93E+00	1	9.91E-01	1	3.86E+02	2	1.73E+00	4	1.13E+01	1
MultiEA[9]	2.06E+01	4	0.00E+00	1	4.29E+01	3	9.06E+00	4	9.06E+00	1	9.94E-01	1	1.25E+01	3
PAP[9]	2.03E+01	3	0.00E+00	1	3.70E+01	2	3.41E+01	5	1.38E+04	5	9.46E+00	5	1.33E+01	5

algorithm	F15		F16		F17		F18		F19		F20		F21	
	mean	r	mean	r	mean	r	mean	r	mean	r	mean	r	mean	r
EAP_IS 2	2.73E-01	1	1.13E+02	4	1.18E+02	2	8.12E+02	1	8.13E+02	1	8.15E+02	2	4.93E+02	1
EAP_IS 1	8.12E-01	2	1.02E+02	3	1.44E+02	3	8.13E+02	2	8.14E+02	2	8.14E+02	1	5.00E+02	4
AMALGAM-SO[9]	2.45E+02	4	2.58E+01	1	4.80E+01	1	9.04E+02	4	9.04E+02	4	9.04E+02	4	5.00E+02	3
MultiEA[9]	1.40E+01	3	7.66E+01	2	1.51E+02	4	8.52E+02	3	8.52E+02	3	8.52E+02	3	4.93E+02	2
PAP[9]	3.24E+02	5	1.46E+02	5	2.23E+02	5	9.07E+02	5	9.07E+02	5	9.07E+02	5	5.12E+02	5

algorithm	F_{22}		F_{23}		F_{24}		F_{25}		averageRank
	mean	r	mean	r	mean	r	mean	r	
EAP_IS 2	5.04E+02	1	5.33E+02	1	2.00E+02	1	2.10E+02	2	2.00
EAP_IS 1	5.04E+02	2	5.34E+02	5	2.00E+02	1	2.11E+02	3	2.64
AMALGAM-SO[9]	8.49E+02	4	5.34E+02	3	2.00E+02	1	2.09E+02	1	2.76
MultiEA[9]	5.77E+02	3	5.33E+02	2	2.00E+02	1	5.79E+02	4	2.84
PAP[9]	9.06E+02	5	5.34E+02	3	2.00E+02	1	1.63E+03	5	4.12

统计结果显示，本文提出的 EAP_IS 组合框架在没有加入子算法参与度动态调整的情况下已经具有了比其他组合框架更好的性能，均值排名第一，证明了该框架的有效性.

4 结论

本文提出一种基于信息共享的一般性组合演化算法框架 EAP_IS，该框架可以允许各种不同种类和数量的演化算法进行合作以提升算法综合性能，降低了算法处理不同种类问题的风险，节约了选择算法所需要的时间.本文提出的 EAP_IS 框架也还有一些不足和需要改进的地方：①适应度评估次数被均匀地分配给各子算法，后续可以根据算法现阶段的表现自适应调整分配适应度，进一步提升算法的性能.②组合框架中子算法之间的互补性也是影响框架性能的重要因素，后期需要进行算法之间差异性和互补性的相关研究.

参考文献(References)

[1] HOLLAND J H. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence [J]. Quarterly Review of Biology, 1992, 6(2): 126-137.

[2] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory[C]// International Symposium on MICRO Machine and Human Science. Nagoya: IEEE Press, 1995: 39-43.

[3] STORN R, PRICE K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces [J]. Journal of Global Optimization, 1997, 11(4): 341-359.

[4] LARRAÑAGA, P, LOZANO J A Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation [M]. Springer Science & Business Media, 2001.

[5] KARABOGA D, BASTURK B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm[J]. Journal of

Global Optimization, 2007, 39(3): 459-471.
[6] TALBI E G. A taxonomy of hybrid metaheuristics [J]. Journal of Heuristics, 2002, 8(5): 541-564.
[7] VRUGT J A, ROBINSON B A, HYMAN J M. Self-adaptive multimethod search for global optimization in real-parameter spaces [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 243-259.
[8] VRUGT J A, ROBINSON B A. Improved evolutionary optimization from genetically adaptive multimethod search[J]. Proceedings of the National Academy of Sciences, 2007, 104(3): 708-711.
[9] YUEN S Y, CHOW C K, ZHANG X. Which algorithm should I choose at any point of the search: An evolutionary portfolio approach[C] // Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation. Amsterdam: ACM Press, 2013: 567-574.
[10] MOLINA D, LOZANO M, HERRERA F. MA-SW-Chains: Memetic algorithm based on local search chains for large scale continuous global optimization [C]// Proceedings of the IEEE Congress on Evolutionary Computation. Barcelona: ACM Press, 2010: 1-8.
[11] PENG F, TANG K, CHEN G, et al. Population-based algorithm portfolios for numerical optimization [J]. IEEE Transactions on Evolutionary Computation, 2010, 14(5): 782-800.
[12] TANG K, PENG F, CHEN G, et al. Population-based algorithm portfolios with automated constituent algorithms selection[J]. Information Sciences, 2014, 27(1)9: 94-104.
[13] AKAY R, BASTURK A, KALINLI A, et al. Parallel population-based algorithm portfolios: An empirical study[J]. Neurocomputing, 2017, 247(C): 115-125.
[14] SOURAVLIAS D, PARSOPOULOS K E, ALBA E. Parallel algorithm portfolio with market trading-based time allocation[C]// Proceedings of the Operations Research. Springer, 2016: 567-574.
[15] GONG W, ZHOU A, CAI Z. A multioperator search strategy based on cheap surrogate models for evolutionary optimization[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(5): 746-758.
[16] Goldberg D E, Richardson J. Genetic algorithms with sharing for multimodal function optimization[C]// Proceedings of the 2nd International Conference on Genetic Algorithms. Cambridge, USA: Hillsdale, 1987: 41-49.
[17] HANSEN N. The CMA evolution strategy: A comparing review[A]// Study in Fuzziness & Soft Computing. Springe, 2006, 192: 75-102.
[18] ZAMBRANOBIGIARINI M, CLERC M, ROJAS R. Standard particle swarm optimisation 2011 at CEC-2013: A baseline for future PSO improvements[C]// IEEE Congress on Evolutionary Computation, 2013: 2337-2344.
[19] QIN A K, HUANG V L, SUGANTHAN P N. Differential evolution algorithm with strategy adaptation for global numerical optimization[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2):398-417.
[20] WANG Y, CAI Z, ZHANG Q. Differential evolution with composite trial vector generation strategies and control parameters [J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 55-66.
[21] SUGANTHAN P N, HANSEN N, LIANG J J, et al. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization [R]. KanGAL Report, NCL-TR-2005001, 2005.
[22] WILCOXON F. Individual comparisons by ranking methods[J]. Biometrics Bulletin, 1945, 1(6): 80-83.
[23] DE LA FUENTE A L T, SáNCHEZ J M P. A framework for hybrid dynamic evolutionary algorithms: Multiple offspring sampling (MOS)[D]. Universidad Politécnica de Madrid,2009.