

## 多点交叉学习组织进化算法\*

吕艳萍<sup>1,2</sup>, 李绍滋<sup>1+</sup>, 周昌乐<sup>1</sup>

<sup>1</sup>(厦门大学 智能科学与技术系, 福建 厦门 361005)

<sup>2</sup>(Department of Informatique, University of Sherbrooke, Canada)

### Multipoint Crossover Learning Organizational Evolutionary Algorithm

LÜ Yan-Ping<sup>1,2</sup>, LI Shao-Zi<sup>1+</sup>, ZHOU Chang-Le<sup>1</sup>

<sup>1</sup>(Department of Cognitive Science, Xiamen University, Xiamen 361005, China)

<sup>2</sup>(Department of Informatique, University of Sherbrooke, Canada)

+ Corresponding author: Phn: +86-592-2580080, Fax: +86-592-2580258, E-mail: szlig@xmu.edu.cn

Lü YP, Li SZ, Zhou CL. Multipoint crossover learning organizational evolutionary algorithm. *Journal of Software*, 2007,18(Suppl.):63-70. <http://www.jos.org.cn/1000-9825/18/s63.htm>

**Abstract:** The original organizational evolutionary algorithm (OEA) is often trapped in local optima when optimizing multimodal functions with high dimensions. In this paper, following an analysis of the main causes of the premature convergence, it proposes a novel algorithm, called the multipoint organizational evolutionary algorithm (mOEA). To discourage the premature convergence, a crossover strategy of multiple points is designed to achieve a better diversity of leader population. Inspired by the cognition and learning physics of social swarms, an improved annexing operator enables members in an organization to either partially climb around their leader or randomly mutate within the search range. The new annexing manipulation both enhances fitness values and preserves a good diversity of member population. Experiments on six complex optimization benchmark functions with 30 or 100 dimensions and very large numbers of local minima show that, comparing with the original OEA and CLPSO, mOEA effectively converges faster, results in better optima, is more robust.

**Key words:** evolutionary algorithm (EA); multimodal function; organization; multipoint crossover

**摘要:** 为了提高组织进化算法(organizational evolutionary algorithm,简称 OEA)在高维多模函数全局优化中陷入局部极值,分析了 OEA 算法早熟收敛的原因,提出了多点交叉学习组织进化算法(mOEA).设计了一个多个组织的领导交叉学习策略来提高组织领导种群多样性,避免早熟收敛;结合社会群体认知和学习的习性,改进 OEA 中的吞并算子,使得同一组织内的个体成员有的在其领导周围爬山运动,有的在搜索范围内随机变异,既提高成员群体的适应度值,又增强成员群体的多样性,不易陷入局部极值.与 OEA 算法和 CLPSO 算法相比,高维、复杂基准函数仿真优化结果表明,新算法收敛速度快,求解精度高,稳定性好.

**关键词:** 进化算法;多模函数;组织;多点交叉

---

\* Supported by the National Natural Science Foundation of China under Grant No.60373080 (国家自然科学基金); the Natural Science Foundation of Fujian Province of China under Grant No.A0310009 (福建省自然科学基金); the 985 Innovation Project on Information Technique of Xiamen University (2004-2007) of China (厦门大学 985 二期信息创新平台项目).

Received 2007-04-15; Accepted 2007-11-25

全局优化问题存在于各个科学研究和工程应用领域中,在工程方案优化、智能商务、信息检索、生物工程等许多领域都有优化算法的广泛应用.目前,优化问题研究的一个关键问题是对避免进入局部极值的有效全局优化算法的创新,特别是对于具有高维、大量局部极值的全局优化,该问题更有挑战性.而高维优化,甚至超高维优化问题已成为现代信息工程应用的一个主要特点,如机械化工厂模板的优化、超大规模集成电路布图等.已有的全局优化算法很少对 30 维或更高维的全局优化问题进行测试<sup>[1,2]</sup>,因此,求解高维问题的优化算法迫在眉睫.

近年来,刘静等人<sup>[3,4]</sup>在 2004 年提出了组织进化算法(organizational evolutionary algorithm,简称 OEA)这种新型的优化算法求解全局优化问题.与其他进化算法的运行机制区别在于,3 个进化算子不是直接作用于个体上,而是作用于组织上.OEA 这种特别的搜索机制已被应用在多个优化问题上,并被证明比 OGA/Q<sup>[1]</sup>,RY<sup>[5]</sup>,GA-MGA<sup>[6]</sup>三种进化算法优越.刘静等人提出的“组织”这个概念对于减少组织进化算法的计算复杂度、提高解的精度是可行的,然而,当求解高维多模优化问题时,OEA 非常容易陷入局部极值而导致早熟收敛.因此,在 OEA 算法的基础上,增强进化种群的群体多样性,提高高维多模函数的优化精度是非常必要的.

本文根据社会群体认知和学习的习性,提出了一种新的组织进化算法的变种,用来提高高维多模优化问题解的精度.本文第 1 节介绍了初始求解全局优化问题的组织进化算法 OEA.第 2 节详细描述本文改进的多点交叉组织进化算法 mOEA.第 3 节描述用于测试算法性能的基准函数,给出本文算法的参数设置和实验结果以及与 OEA 算法和 CLPSO 算法实验结果比较分析.最后的小结中给出相应的结论以及进一步的工作方向.

## 1 引言

### 1.1 问题定义

无约束最优化问题可描述为下面的  $D$  维最小化问题:

$$\text{Min } f(x), x = [x_1, x_2, \dots, x_D], \text{ s.t. } l \leq x \leq u,$$

这里,  $x$  是  $D$  维欧式空间中的一个变量,  $f(x)$  是变量  $x$  对应的目标函数, 向量  $l = [l_1, l_2, \dots, l_D]$  和  $u = [u_1, u_2, \dots, u_D]$  定义了待优化问题的可行解空间.

### 1.2 组织进化算法 OEA

文献[3]提出了组织进化算法 OEA,该算法首先初始化一个规模大的群体,再通过分裂算子、吞并算子和合作算子这 3 个作用于组织群体的进化算子来引导种群进化.大规模的初始化过程使该算法不易陷入局部极值,而 3 个组织进化算法的搜索机制使其即使也具有较快的收敛速率<sup>[3]</sup>.该组织进化算法已成功地应用于实际优化问题,如组合优化、多目标优化等<sup>[7-9]</sup>.

在组织进化算法中,个体用实数向量表示,组织为若干个个体的集合.这些个体称为组织的成员,适应度最大的成员称为领导成员.一个组织形成之后,若组织过于庞大就会被分成若干个小组织,若组织实力弱则会被吞并,也有可能某些环境下,组织会与其他组织相互协作增强彼此的实力.组织进化算法的吞并策略是一个组织的全体成员与另一个组织的领导启发式交叉或者启发式变异,形成新个体群体即新组织,因此在搜索空间内,第一个组织的成员随机地向另一个组织的领导随机靠近,其作用相当于对较好的解进行局部爬山操作.合作策略则通过两个组织领导成员的相互作用来增强彼此的适应度.

更多算法的思想见文献[3].

## 2 多点交叉组织进化算法

虽然组织进化算法具有一个较快的收敛率,但是对于高维或超高维多模、有大量局部极值的全局优化问题,组织进化算法仍然存在着早熟收敛问题.在组织进化算法的吞并策略里,一个组织的所有成员在另一组织的领导周围局部爬山操作,这些成员快速地受其吸引,即该组织群体多样性迅速降低,因此,在拥有大量局部极值的复杂搜索环境中,容易快速收敛于该领导成员.并且,在合作策略里的两个交叉操作算子局限作用于两个组织

的两个领导,搜索空间非常有限.从领导成员演绎出的个体成员势必受其误导而远离全局极值,所以当搜索环境复杂、具有大量局部极值时,组织进化算法容易陷入局部极值.

一个个体的适应度值依赖于  $D$  维空间的  $D$  个方向值,一个适应度值劣的个体必然在某些维上具有很差的变量值,从而影响到该个体的整体适应度值.为了充分利用个体在各个维数上的有益信息,Liang 在文献[10]中提出了综合学习策略来保持群体的多样性.这里,我们引入这种多父亲交叉的学习机制到组织进化算法的合作算子,保证各个组织的领导成员均能被潜在运用以提高被选中两个组织的领导成员的适应度值.

事实上,在社会活动中,在一个组织里,个体成员之间的认知和学习是多方面的,且需要一个时间过程,是有差异的.受其启发,分析了在一个组织小群体内,每个个体向其所在组织的领导学习到的有益信息是有差异、不一致的.在新算法中整改了合作算子,在一个组织内,一部分个体成员在每一维均向其领导学习,另一部分个体成员则随机产生变异,既提高了组织内群体多样性,又增强了组织内部分个体的适应度值.

## 2.1 多点交叉学习合作算子

首先随机选择两个父代组织,在组织的多个领导成员中运用多点交叉算子产生子代领导成员,子代的个体成员是通过父代的个体与子代的领导算术交叉或者离散交叉而得到的.最后,从现有组织群体中删除被选中的这两个父代组织,下一代组织群体中增加新产生的子代组织.

子代组织领导成员的每一维向选中的父代组织领导成员学习,其中选择机制采用竞标选择策略,主要是为了保证子代领导的每一维具备更好的适应度值,避免在某一维上适应度值过低影响整个个体的适应度值.选择过程是从所有的父代组织领导成员随机选择两个,比较它们的适应度值,让待产生的子代领导的某一维向适应度值高的父代领导成员学习.因此,新领导成员的各维来自于不同的、适应度高的领导成员,降低了浪费在无效搜索的时间,且即使某一个父代领导陷入局部极值,在其他领导成员的牵引下,也能有效地跳出局部极值点.

下面 Matlab 源代码描述了如何产生组织的子代个体成员和领导成员.

```
fi1=ceil(parentOrgaNum*rand(1,Dim));
fi2=ceil(parentOrgaNum*rand(1,Dim));
fi=(fparentLead(fi1)<fparentLead(fi2)).*fi1+(fparentLead(fi1)>= fparentLead(fi2)).*fi2;
for dimcnt=1:Dim
    childLead(dimcnt)=parentLead(fi(dimcnt),dimcnt);
end
aa=round(rand(1,Dim));
childmember(i)=aa.*parentmember(i)+(1-aa). *childLead;
```

这里,  $fi$  决定了子代的领导成员的每一维从哪个组织的父代领导取值,  $aa$  是一个随机向量,该向量的每个元素决定了一个父代个体成员和子代领导成员交叉运算的全值.

## 2.2 新的吞并算子

随机选定两个父代组织,进化成两个中间种群,随后将它们引入子代组织群体中.若  $U(0,1) < AS$ ,则中间群体的成员由吞并策略 1 决定,向其组织的领导成员靠近;否则,由吞并策略 2 决定在搜索范围内变异; $U(0,1)$  是产生的一随机数, $AS$  是预先指定的参数.最后,从现有组织群体中删除被选中的这两个父代组织,下一代组织群体中增加新产生的子代组织.

给定两个父代组织  $Org_1, Org_2$ ,它们的领导成员分别为  $Lead_1$  和  $Lead_2$ ,组织  $Org_1$  中的第  $i$  个成员和  $Org_2$  中的第  $j$  个成员分别表示为  $Mem_i, Mem_j$ ,它们依照吞并策略的公式(1)~(3)来进化.

$$Mem_{i,k} = \begin{cases} Lead_{1,k} + U_k(0,1) \cdot (Lead_{1,k} - Mem_{i,k}), & U(0,1) < AS \\ Swarm_k + U_k(0,1) \cdot Swarm_k, & \text{else} \end{cases} \quad (1)$$

$$Mem_{j,k} = \begin{cases} Lead_{2,k} + U_k(0,1) \cdot (Lead_{2,k} - Mem_{j,k}), & U(0,1) < AS \\ x_k + U_k(0,1) \cdot x_k, & \text{else} \end{cases} \quad (2)$$

$$Mem_{i,k}, Mem_{j,k} = \begin{cases} \overline{x_k}, & Mem_{i,k}, Mem_{j,k} > \overline{x_k} \\ \underline{x_k}, & Mem_{i,k}, Mem_{j,k} < \underline{x_k} \\ \overline{Mem_{i,k}}, \overline{Mem_{j,k}}, & \text{else} \end{cases} \quad (3)$$

其中,  $\overline{x_k}$  和  $\underline{x_k}$  是搜索空间的上下限,  $\overline{Swarm_k}$  和  $\underline{Swarm_k}$  是中间种群变量值的上下限,  $U_k(0,1)$  是为每一维产生的随机数.

### 2.3 mOEA与OEA的区别

mOEA 算法和 OEA 算法<sup>[4]</sup>的区别在于以下几点:

1) 在合作算子中,新的领导成员能够从所有父代组织的领导成员学习,不限于被选中两个组织的领导成员,并且,新的领导成员不同的维可向不同的领导成员学习,因此,新的领导成员通过所有父代领导的共同协作,当陷入局部最优时能够容易跳出局部极值,从而更有效地引导其相应的个体成员搜索.

2) mOEA 中的吞并算子与 OEA 算法不同的是,不同的策略被应用到被选中的两个组织.为了提高合并后新组织的群体多样性,避免早熟收敛问题,一些个体成员在其领导成员周围局部爬山操作,而另一些成员则以一定的概率在搜索范围内变异或在中间群体范围内变异;同时,该新策略增强了个体成员的适应度值,改善了算法局部搜索能力.

领导群体与个体成员群体独立进化,替代这两个群体的频繁相互作用.一个领导成员仅当其个体成员的适应度值比它更优则被取代,而个体成员仅受父代领导成员和它们自身适应度的影响,因此容易分析群体行为.

## 3 算法实验验证

### 3.1 基准函数

为了评价本文提出的多点交叉学习组织进化算法(mOEA)对高维多模函数优化时的求解精度,引入了 1 个复杂的单模基准函数以及 5 个多模基准函数进行分析,将该算法与组织进化算法(OEA)和综合学习微粒群算法(CLPSO)进行了对比.所有的基准函数在 30 维和 100 维的情况分别做了测试,特别是 100 维的函数优化,是目前的一个研究难点.高维测试函数的特点以及公式描述如下:

1) Rosenbrock's 函数

$$f_1 = \sum_{i=1}^D [100 \cdot (x_{i+1} - x_i^2)^2 + (x_i - 1)^2].$$

Rosenbrock 函数是一个经典复杂优化问题,可被当成多模优化函数,它的全局最优点位于一个平滑、狭长的抛物线形山谷内.由于函数仅为优化算法提供了少量信息,使算法很难辨别搜索方向,找到全局最小点的机会微乎其微,该函数通常用于评价优化算法的执行效率.在下一节的实验测试过程中,我们也发现优化算法一般只能获得该函数的局部极值.

2) Ackley's 函数

$$f_2 = 20 + e - 20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left[ \frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right].$$

3) Griewank's 函数

$$f_3 = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^n \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1.$$

4) Weierstrass's 函数

$$f_4 = \sum_{i=1}^D \left( \sum_{k=0}^{k_{\max}} [a^k \cdot \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{\max}} [a^k \cdot \cos(2\pi b^k \cdot 0.5)], a=0.5, b=3, k_{\max}=20.$$

## 5) Rastrigin's 函数

$$f_5 = \sum_{i=1}^D [x_i^2 - 10 \cdot \cos(2\pi x_i) + 10].$$

## 6) Schwefel's 函数

$$f_6 = 418.9829 \cdot D - \sum_{i=1}^D x_i \cdot \sin\left(|x_i|^{1/2}\right).$$

本文引入 5 个多模基准函数进行测试. Ackley's 函数在狭长的全局极值点周围拥有很多的局部极值,它是 6 个基准测试函数中最容易得到最优解的,主要用以测试算法的收敛率. Griewank's 函数由于其各维上的变量是密切相关的,互相作用,因此很难取得最优解. Weierstrass's 是一个处处连续又只在有限的几个点可微分的函数. Rastrigin's 函数是一个复杂的多模问题,具有大量的局部极值点,主要用于检验算法的群体多样性. Schwefel's 函数由于其全局极值点周围的所有局部极值点远离它,因此搜索难以抵达全局极值点,主要用于检验算法的种群多样性.

表 1 列出了各个基准测试函数的搜索范围、全局极值点以及全局极值.

**Table 1** The properties of six benchmark test functions

**表 1** 6 个基准测试函数的基本属性

Benchmark function	Search range	Global optimum point	Global extreme value
Rosenbrock $f_1$	[-2.048, 2.048]	[1, 1, ..., 1]	0
Ackley $f_2$	[-32.768, 32.768]	[0, 0, ..., 0]	0
Griewank $f_3$	[-600, 600]	[0, 0, ..., 0]	0
Weierstrass $f_4$	[-0.5, 0.5]	[0, 0, ..., 0]	0
Rastrigin $f_5$	[-5.12, 5.12]	[0, 0, ..., 0]	0
Schwefel $f_6$	[-500, 500]	[420.96, ..., 420.96]	0

### 3.2 参数设置

为了验证多点交叉学习组织进化算法对基准测试函数的性能,该算法分别对规模为 30 维和 100 维的基准函数进行测试,并将实验结果与组织进化算法和综合学习的微粒群算法进行比较.在对 30 维基准函数测试时,最大的适应度评估次数为 100 000,解的求解精度设为  $\varepsilon=10^{-10}$ ;在对 100 维基准函数测试时,最大的适应度评估次数为 300 000,解的求解精度设为  $\varepsilon=10^{-5}$ .每种算法在每个基准函数上均运行 30 次,记录得到的每 30 个测试结果,并且计算它们的均值和方差.

除了上面提及的共同参数外,还需要为 OEA 设置 5 个参数,为 CLPSO 设置 7 个参数. OEA 的 5 个参数为种群规模  $n_o$ ,组织的最大规模  $\max_{os}$ ,决定选择吞并算子或合作算子的调节概率  $p_r$ ,吞并算子和合作算子的两个调制策略概率 AS 和 CS,设置与文献[3]OEA 算法一致,分别为 150, 20, 0.5, 0.8, 0.6. CLPSO 的其余 7 个参数为种群规模、最大迭代次数、学习向量、惯性权值向量、群体认知学习的最大迭代次数,其设置与文献[10]一致,分别为 40, 3000, [1.49445, 1.49445], 0.95~0.4, 5. 而 mOEA 算法由于多点交叉学习的引入不再需要参数 CS,所以, mOEA 的其余 4 个参数设置为 150, 5, 0.5, 0.6. 为了提高一个组织内个体成员的多样性, mOEA 算法的参数  $\max_{os}$  设置比 OEA 算法来得小.

### 3.3 实验比较

本节实现了 mOEA 和 OEA 这两种算法, CLPSO 算法程序来自于文献[10],对第 3.1 节中的 6 个基准函数进行测试,这 6 个函数被设置 30 维和 100 维分别进行测试.为了统计评价 mOEA 算法的实验结果与其他两种算法实验结果的区别,每种算法在每个测试函数连续运行 30 次,并且记录了以下变量值:1) 每次测试的函数评估次数, 2) 30 次连续测试得到的最优/最差函数值, 3) 30 个函数值的均值和方差.

30 维基准函数的测试结果如图 1 所示.

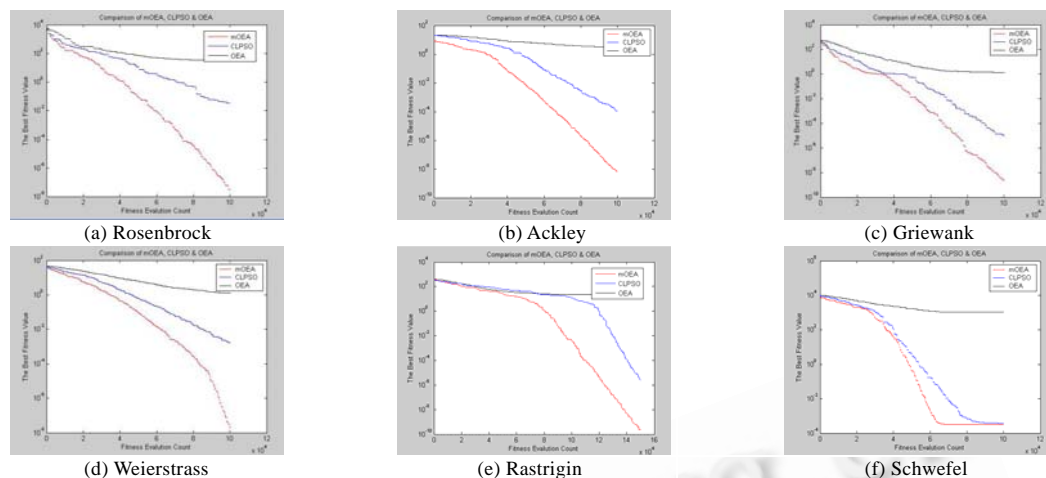


Fig.1 mOEA vs. CLPSO and OEA on 30D benchmark functions

图1 mOEA与CLPSO,OEA在30维基准函数上的比较

在每个30维基准函数上,每种算法被连续运行30次,图1是mOEA,CLPSO和OEA在一次运行中的最优适应度值变化曲线图,从该图可以清晰地观察到最优适应度值在进化过程中的变化情况。

从图1的6个子图可以观察到,mOEA对30维基准函数优化测试结果均比CLPSO和OEA更佳.在最大迭代次数内,它总是能取得最佳的求解精度,具有良好的全局收敛能力.在每个测试中,容易观察到,OEA算法在迭代初期运行良好,但迭代后期容易收敛于某一局部极值点,且没有能力改进最佳适应度值,对于多模函数优化,该缺陷尤其明显.CLPSO在进化过程中,总能取得不同的最佳适应度值,说明其微粒群体具有种群多样性,不容易陷入某一局部极值点,然而,函数优化的收敛速度非常缓慢.图1的所有子图表明,mOEA算法在函数优化问题上不会发生类似的现象,该算法在整个进化过程中最佳适应度值总是不断变化,得到改善,并且它较快地收敛于全局极值.但对于Schwefel函数优化,mOEA虽然比其他两种算法更早达到某一局部极值点,但这3种优化算法均无法到达该函数的全局极值点,主要由于该函数的局部极值点远离全局极值点,无法为全局优化提供有效的搜索信息。

**Table 2** The best/worst, mean and variance results achieved on the six benchmark functions with 30 dimensions using mOEA vs. CLPSO and OEA after running 30 times

**表2** 在每个30维基准函数上,mOEA与CLPSO和OEA连续运行30次得到的最佳/最差适应度值/适应度值均值和方差的比较

Algorithm	mOEA		LPSO		OEA	
	Best/Worst	Mean/Variance	Best/Worst	Mean/Variance	Best/Worst	Mean/Variance
Rosenbrock	9.8635e-006	6.4325e-006	0.018 4	0.178 4	29.456 1	31.541 2
	2.8229e-006	3.5237e-006	0.683 0	0.283 6	33.722 1	2.134 6
Ackley	5.1624e-009	9.0384e-009	1.3437e-004	1.1120e-003	1.586 3	3.978 7
	1.1997e-008	3.5084e-009	2.9584e-003	1.6000e-003	9.522 1	3.774 1
Griewank	2.4345e-010	3.5592e-009	3.7990e-006	7.3576e-006	1.255 4	4.238 3
	9.2168e-009	4.9238e-009	1.3840e-005	5.6229e-006	9.654 7	3.954 9
Weierstrass	2.3974e-008	3.6257e-008	0.001 1	0.001 6	2.116 7	2.716 6
	6.9683e-008	1.2741e-008	0.002 0	3.2201e-004	3.200 1	0.452 2
Rastrigin	2.3781e-010	5.1804e-010	3.5246e-009	7.8547e-009	14.508 9	29.061 6
	1.1492e-009	3.6231e-010	1.1807e-008	3.1959e-009	39.584 6	8.843 9
Schwefel	3.8209e-004	3.8266e-004	3.8183e-004	3.8183e-004	961.194 3	1.2242e+003
	3.8343e-004	5.0379e-007	3.8183e-004	0	1.3515e+003	177.655 9

表2的数据进一步表明了,在最大迭代次数内,mOEA对于上述6个基准函数的优化均取得比CLPSO和OEA更好的求解精度.实验结果表明,上述复杂多模基准函数测试中,OEA算法无法找到全局最优解.也可以看出,新算法的求解精度明显优于OEA以及PSO的典型变种CLPSO,且得到解的适应度均值为全局极值或近似

全局极值,得到结果适应度值的方差也相对小.CLPSO 也能达到近似全局极值,但是适应度值均值不理想,且由于综合学习策略的使用,方差也较大.

100 维测试结果如下:

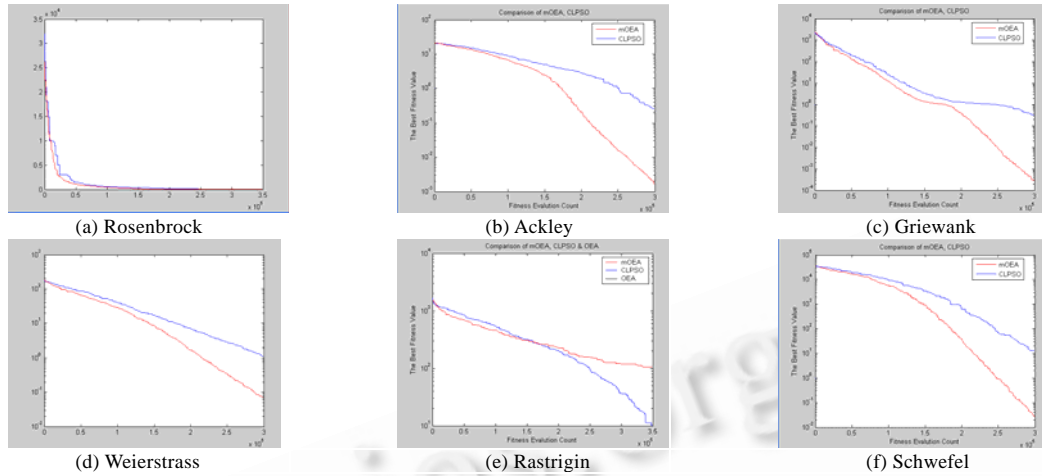


Fig.2 mOEA vs. CLPSO on 100 D benchmark functions

图 2 mOEA 与 CLPSO 在 100 维基准函数上的比较

图 2 表明了 mOEA 和 CLPSO 在 6 个 100 维基准函数优化时的性能比较.同样地,两种算法在每个测试函数上分别连续运行 30 次,图 2 为每种算法在一次运行中最佳适应度值被记录且被画于图上.由于 OEA 算法易失去群体多样性而陷入局部极值点,因此,OEA 算法就不被考虑在 100 维基准函数的测试范围内.

上面最佳适应度变化曲线图,将 mOEA 和 CLPSO 算法性能进行了比较.CLPSO 已被证明能够优越求解高维多模函数优化问题,将 mOEA 与 CLPSO 算法进行比较具有代表性意义.不同于 30 维函数优化,100 维函数优化非常复杂,因此导致两种算法得到的实验结果比 30 维的更差.所幸的是,这两种算法均有能力提高种群多样性的策略来克服早熟收敛问题.可以看出,在最大迭代次数内,除了 Rastrigin's 函数外,mOEA 在其他函数的测试比 CLPSO 取得了更优的函数精度值.

Table 3 The best/worst, mean and variance results achieved on the six benchmark functions with 100 dimensions using mOEA vs. CLPSO and OEA after running 30 times

表 3 在每个 100 维基准函数上,mOEA 与 CLPSO 连续运行 30 次得到的最佳/最差适应度值/适应度值均值和方差的比较

Algorithm Function	mOEA		CLPSO	
	Best/Worst	Mean/Variance	Best/Worst	Mean/Variance
Rosenbrock	29.4103/36.9807	33.2235/3.1011	29.1403/36.9807	33.1448/3.2109
Ackley	1.7718e-003/2.0189e-003	1.8829e-003/1.0743e-004	0.2474/0.2920	0.2615/0.0205
Griewank	1.8541e-00/2.7546e-004	2.2384e-004/3.8382e-005	0.3178/0.3564	0.3311/0.0181
Weierstrass	0.0695/0.0727	0.0714/0.0015	1.1416/1.6571	1.2900/0.2475
Rastrigin	70.2137/120.5096	98.7208/20.9695	6.1080/42.8249	24.6829/9.2951
Schwefel	0.0151/0.0448	0.0303/0.0128	12.5568/13.1160	12.7374/0.2590

表 3 的数据表明了 mOEA 和 CLPSO 的实验结果.除了 Rastrigin'函数外,mOEA 在 100 维的 Ackley, Griewank,Weierstrass 和 Schwefel 函数优化中均能找到近似全局最优解,但是该算法需要耗费比 30 维函数优化更多的函数评估次数才能达到近全局极值.对于这 4 个函数和 Rosenbrock's 函数的优化,mOEA 也能得到比 CLPSO 更好的平均适应度值和更小的解方差,说明 mOEA 算法更加稳定,可以得到更佳的解质量.

#### 4 小 结

本文借鉴社会群体认知的习性,分析了组织进化算法中个体之间互相学习的规律,给出了一种求解数值优

化问题的改进 OEA 算法——多点交叉学习组织进化算法。mOEA 算法中,合作算子的多点交叉学习策略有助于提高领导群体的多样性,避免算法早熟收敛;同一组织内,吞并算子的各维交叉和变异使得一部分个体提高适应度值,一部分随机变异,既提高了组织内个体群体的多样性,又改善了个体的质量。

**致谢** 在此,我们感谢 Suganthan 教授向我们提供了 CLPSO 的源码。

#### References:

- [1] Leung YW, Wang YP. An orthogonal genetic algorithm with quantization for global numerical optimization. *IEEE Trans. on Evolutionary Computation*, 2001,5(1):41–53.
- [2] Yao X, Liu Y. Fast evolution strategies. In: *Proc. of the 6th Conf. on Evolutionary Programming*. LNCS 1213, 1997. 151–161.
- [3] Liu J, Zhong WC, Liu F, Jiao LC. An organizational evolutionary algorithm for constrained and unconstrained optimization problems. *Chinese Journal of Computers*, 2004,27(2):157–167 (in Chinese with English abstract).
- [4] Liu J, Zhong WC, Liu F, Jiao LC. Numerical optimization using organizational evolutionary algorithm. In: *Proc. of the 5th Conf. on Computational Intelligence and Multimedia Applications (ICCIMA 2003)*. IEEE Press, 2003. 284–289.
- [5] Runarsson TP, Yao X. Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. on Evolutionary Computation*, 2000,4(3):284–294.
- [6] Kazarlis SA, Papadakis SE, Theoharis JB, Petridis V. Microgenetic algorithms as generalized hill-climbing operators for GA optimization. *IEEE Trans. on Evolutionary Computation*, 2001,5(3):204–217.
- [7] Liu J, Zhong WC, Liu F, Jiao LC. Classification based on organizational coevolutionary algorithm. *Chinese Journal of Computers*, 2003,26(4):446–453 (in Chinese with English abstract).
- [8] Zhong WC, Liu J, Liu F, Jiao LC. Combinatorial optimization using multi-Agent evolutionary algorithm. *Chinese Journal of Computers*, 2004,27(10):1341–1353 (in Chinese with English abstract).
- [9] Liu J, Zhong WC, Liu F, Jiao LC. An organizational evolutionary algorithm for SAT problem. *Chinese Journal of Computers*, 2004,27(10):1423–1428 (in Chinese with English abstract).
- [10] Liang JJ, Qin AK, Suganthan PN. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. on Evolutionary Computation*, 2006,10(3):281–295.
- [11] Lü YP, Li SZ, Jiang QS, Guo WZ. Particle swarm optimization based on information diffusion and clone selection. In: *Proc. of the 6th Conf. on Simulated Evolution and Learning (Seal 2006)*. LNCS 4247, 2006. 521–528.

#### 附中文参考文献:

- [3] 刘静,钟伟才,刘芳,焦李成.组织进化数值优化算法. *计算机学报*,2004,27(2):157–167.
- [7] 刘静,钟伟才,刘芳,焦李成.组织协同进化分类算法. *计算机学报*,2003,26(4):446–453.
- [8] 钟伟才,刘静,刘芳,焦李成.组合优化多智能体进化算法. *计算机学报*,2004,27(10):1341–1353.
- [9] 刘静,钟伟才,刘芳,焦李成.组织进化算法求解 SAT 问题. *计算机学报*,2004,27(10):1423–1428.



吕艳萍(1978—),女,福建永春人,博士生,主要研究领域为智能计算,群集/进化/模糊优化,多目标优化。



周昌乐(1959—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为人工智能及其应用技术,艺术认知与计算。



李绍滋(1963—),男,教授,博士生导师,CCF 高级会员,主要研究领域为人工智能与自然语言处理,智能多媒体信息检索,智能中医信息处理,网络多媒体及 CSCW 技术,软件体系结构与中间件技术。