

文章编号: 2095-4980(2016)02-0276-06

## 合作式粒子群优化算法

杜清福

(上海晨越计算机科技有限公司, 上海 201100)

**摘要:** 针对粒子群优化(PSO)算法随着维数增加而导致的收敛速度慢, 容易陷入局部最优的问题, 提出了一种合作式粒子群(CPSO)算法。通过多粒子群不同的组态向量合作, 显著改善了标准算法的早熟问题。利用标准测试函数对CPSO算法、协同进化遗传算法(CCGA)、遗传算法(GA)、PSO算法进行比较测试, 结果表明, CPSO算法在多个基准优化问题方面显示了较佳性能。

**关键词:** 收敛行为; 合作式算法; 合作种群; 粒子群优化算法

**中图分类号:** TN911.7

**文献标识码:** A

**doi:** 10.11805/TKYDA201602.0276

## Cooperative Particle Swarm Optimization algorithm

DU Qingfu

(Shanghai Chen Yue Computer Technology Co., Ltd. Shanghai 201100, China)

**Abstract:** Particle Swarm Optimization(PSO) algorithm converges more slowly as the dimension increases, which easily cause the local optimum. A Cooperative Particle Swarm Optimization(CPSO) algorithm is presented, by using the way of cooperation to improve the premature convergence problem of standard algorithm. CPSO, Cooperative Coevolutionary Genetic Algorithm(CCGA), Genetic Algorithm(GA), PSO algorithm are compared with the test of standard function. The results indicate that CPSO algorithm shows better performance in a number of benchmark optimization problems, compared with the traditional algorithm.

**Key words:** convergence behavior; cooperative algorithm; cooperative population; Particle Swarm Optimization algorithm

大多数随机优化算法(包括PSO算法和GA)的执行能力随着搜索空间维数的增大而变差。一个基本的随机全局搜索算法, 产生一个均匀分布的样本覆盖整个搜索空间, 当产生一个落在最优区域的解决方案时, 该算法停止运算, 围绕着全局最优的小体积空间搜索。产生最优区域内的样品的概率是最优区域体积除以搜索空间的体积, 随着搜索空间维数的增大, 这个概率呈指数衰减<sup>[1-3]</sup>。低维拓扑结构很难找到高维问题的全局最优解。解决此问题的一种方法是把搜索空间划分成低维子空间, 分别运用最优化算法, 确保搜索到空间的每一个可能的区域。遗传算法是进化算法的一部分, 它以一定的方法优化种群类问题, 采用多种重组, 选择和变异算子, 在每次迭代过程中产生新的解决方案。由于它们的随机特性, 将分区搜索空间向量分割成更小的向量, 这些较小的搜索空间由一个单独的GA搜索, 每个GA代表较小的子空间。这种分解导致基本遗传算法性能明显改善<sup>[4]</sup>, 但并没有深入分析分区可能导致伪极小值的引入, 同时没有意识到合作共同进化遗传算法参数之间的依赖。针对粒子群优化(PSO)算法随着维数增加而导致的收敛速度慢, 容易陷入局部最优的问题, 提出了一种合作式粒子群(CPSO)算法, 即由2个算法模型CPSO-S<sub>k</sub>和CPSO-H<sub>k</sub>相互合作协同组成。

### 1 PSO算法及改进

#### 1.1 PSO算法

PSO算法由Kennedy和Eberhart提出, 是一种随机优化技术, 可以比作一群飞鸟的行为或一群人的社会行为, 被用来解决一系列优化问题, 包括神经网络训练和函数最小化。以 $s$ 代表种群的大小, 每个个体在 $1 \leq i \leq s$ 这个范

围内变化,当前搜索空间的位置为 $x_i$ ,当前的速度为 $v_i$ ,在搜索空间中的个体极值为 $y_i$ ,在每次迭代中,种群中的每个粒子按照式(1)和式(2)更新<sup>[5-7]</sup>。假设 $f$ 是最小化函数,种群是由 $n$ 个粒子群组成的, $r_1 \sim U(0,1)$ , $r_2 \sim U(0,1)$ 是2个变化范围为(0,1)的随机序列。

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1r_{1,i}(t)[y_{i,j}(t) - x_{i,j}(t)] + c_2r_{2,i}(t)[\hat{y}_{i,j}(t) - x_{i,j}(t)] \quad (1)$$

式中: $v_{i,j}$ 是第 $i$ 个粒子的第 $j$ 维空间, $j \in 1,2,\dots,n$ ;  $c_1, c_2$ 分别为粒子群1,2的学习因子。新的粒子位置由式(2)得到。

$$X_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

每个粒子的个体最优位置采用如下公式更新:

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if } f(X_i(t+1)) \geq f(y_i(t)) \\ x_i(t) & \text{if } f(X_i(t+1)) < f(y_i(t)) \end{cases} \quad (3)$$

$\hat{y}$ 定义为:

$$\hat{y}(t+1) = \arg \min_{y_i} f(y_i(t+1)), \quad 1 \leq i \leq s \quad (4)$$

## 1.2 改进粒子群算法

标准PSO算法在搜索空间被困在一个非最优位置。因此介绍一种改进的算法CPSO-H<sub>k</sub>,结合CPSO-S<sub>k</sub>算法与PSO算法,并保留2种算法的最佳性能,以解决早熟问题。

### CPSO-H<sub>k</sub>算法

鉴于PSO算法逃离伪极小值的能力,CPSO-S<sub>k</sub>算法在某些功能具有更快的收敛性,理想情况下可以利用这2种算法的特性。可以试图用CPSO-S<sub>k</sub>构造一个算法,但切换到PSO算法的CPSO-S<sub>k</sub>算法已经成为局部最优。这就需要考虑何时切换算法,实际上很难实现。

另一种方法是交错2个算法,CPSO-S<sub>k</sub>算法先迭代执行,随后执行一个迭代PSO算法。可以在每次迭代结束时交换2种算法目前所发现的最优解信息,然后将该最优解信息形成复合式算法。

一种简单的机制可以实现这种信息交换,具体地说,在一种迭代的CPSO-S<sub>k</sub>算法中,相关向量用于覆盖PSO算法中一个随机选择的粒子,随后执行一个迭代的Q群算法,产生一个新的全局最好的粒子向量。然后将这个向量分成子向量,覆盖随机选择的粒子P<sub>j</sub>群。

虽然粒子在覆盖信息交换过程中是随机选择的,但该算法不会覆盖种群全局最佳位置,因为这有可能产生对群体性能不利的影 响。实验研究也表明,信息交换过多使用此机制可能阻碍算法的进展<sup>[8-9]</sup>。使用统一随机分布选择一个粒子群(针对替换),极有可能是一个 $s$ 粒子群被所有 $2s$ 粒子覆盖,在信息交换过程中,只有全局最优的粒子得到保护。如果P<sub>j</sub>群落后于Q群,意味着P<sub>j</sub>群可以仅在几个迭代中覆盖Q种群中的劣质位置。另一方面,Q群将以同样的速度覆盖粒子P<sub>j</sub>群,所以整个算法永远保存最佳解决方案。由于信息交换频繁,粒子的多样性会显著减少,通过限制粒子的数量,可以积极参与信息交流,防止成群意外,保持种群的多样性。因此,正常运行的种群是没有被干扰的。

## 2 实验条件

为比较不同算法,必须选择一个公平的时间测量。因为它们处理的向量较小,分割和混合复合式算法开销较低。如果使用处理时间作为时间测量,将造成不公平。由于算法不同,内部循环迭代的数量不能用时间衡量。因此用数量(FEs)作为时间函数,来作为评估方式。这里介绍的所有函数值0是全局最小点,选择以下函数进行测试。

Ackley的函数(多通道):

$$f_1(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$$

广义Griewank函数(多通道):

$$f_2(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

表1列出的参数用于实验。“domain”列中的值为初始随机粒子的大小,“threshold”列中的值为某些测试停止准则。

对于大部分的函数,尤其当它们之间存在交互变量 $f_1$ 和 $f_2$ ,使用简单的方法,如松弛法,难以解决。因为它

们是预定的，且变量之间具有相关性，使得CPSO算法更难处理，所以这些函数被测试前，在旋转坐标下采用所罗门的算法<sup>[10-12]</sup>，使每个个体在运行一个新的旋转之前经过一个特定的旋转计算为无偏差的。

## 2.1 PSO结构

所有实验运行 $2 \times 10^5$ 次或到误差低于停止阈值时结束，根据实验类型执行。选择迭代次数对应为 $10^4$ 次迭代的普通PSO算法(20个粒子)。所有实验均运行50次，报告的结果来自50个运行计算的平均值。每个类型的群的实验重复使用10,15和20粒子群。对以下类型算法进行测试：

PSO：“原型”种群使用 $c_1=1.49, c_2=1.49, w=0.72, v_{\max}$ 是限制域。

CPSO-S：一个最大的“分离”种群使用 $c_1=1.49, c_2=1.49, w$ 随时间线性递减， $v_{\max}$ 是限制域。

CPSO-S<sub>6</sub>：“分离”种群使用 $c_1=1.49, c_2=1.49, w$ 线性递减。这种种群类型和分离CPSO的不同之处在于搜索空间向量对CPSO-S<sub>6</sub>分离仅6个部分(每部分由5个部分组成)，代替了30个部分。

CPSO-H：一个混合种群，组成了一个最大化的分离种群，这2个组件使用的值 $c_1=1.49, c_2=1.49, w$ 线性递减，且 $v_{\max}$ 是限制域。

$c_1, c_2, w$ 参数值根据其他文献<sup>[13]</sup>的经验选择，并提供良好的性能。

## 2.2 GA配置

实验重复使用GA，结果使用合作遗传算法，2个GA算法标记如下。

GA：指定标准遗传算法。

CCGA：合作遗传算法，空间搜索向量最大限度地分开，这样每个组件属于自己的群体。选取30个种群进行算法测试。

2种遗传算法的参数如下：

染色体类型：二进制编码；

染色体长度：48位/函数变量；

交叉概率：0.6；

交叉策略：2点；

突变概率： $1/(48 \times 30)$ 假设30变量/函数；

适应度缩放：窗口长度为5；

繁殖策略：适应度是与精英策略成比例变化；

人口规模：100。

注意，CCGA考虑函数的每个参数的人口，对应分裂CPSO。选择48位/变量的PSO算法和GA算法之间的比较更加公平。

## 3 结果分析

### 3.1 固定迭代结果

收集了所有的方法固定 $2 \times 10^5$ 次的函数测试运行评估值，见表2、表3。第2列为每个s群的粒子数或GA的人口规模。第3和第4列为运行 $2 \times 10^5$ 次后的平均误差和95%置信区间函数值，分别为非旋转和旋转版本，所使用的所有函数最小值为0。

表1 试验函数

parameters	n	domain	threshold
$f_1$	30	30	5.00
$f_2$	30	600	0.10

表2 Ackley( $f_1$ )函数运行 $2 \times 10^5$ 次后函数值

Table2 Ackley( $f_1$ ) after  $2 \times 10^5$  function evaluation

algorithm	s	mean(unrotated)	mean(rotated)
PSO	10	$7.33 \pm 6.23 \times 10^{-1}$	$7.54 \pm 5.82 \times 10^{-1}$
	15	$4.92 \pm 5.81 \times 10^{-1}$	$5.09 \pm 5.11 \times 10^{-1}$
	20	$3.57 \pm 4.58 \times 10^{-1}$	$3.42 \pm 3.74 \times 10^{-1}$
CPSO-S	10	$2.90 \times 10^{-14} \pm 1.60 \times 10^{-15}$	$1.73 \times 10^1 \pm 1.45$
	15	$3.01 \times 10^{-14} \pm 1.42 \times 10^{-15}$	$1.81 \times 10^1 \pm 1.09$
	20	$3.05 \times 10^{-14} \pm 1.84 \times 10^{-15}$	$1.85 \times 10^1 \pm 7.76 \times 10^{-1}$
CPSO-H	10	$2.78 \times 10^{-14} \pm 1.71 \times 10^{-15}$	$1.43 \times 10^1 \pm 1.57$
	15	$2.92 \times 10^{-14} \pm 1.67 \times 10^{-15}$	$1.43 \times 10^1 \pm 1.48$
	20	$2.98 \times 10^{-14} \pm 1.56 \times 10^{-15}$	$1.60 \times 10^1 \pm 1.42$
CPSO-S <sub>6</sub>	10	$1.12 \times 10^{-6} \pm 1.12 \times 10^{-1}$	$7.98 \times 10^{-1} \pm 1.06 \times 10^2$
	15	$1.11 \times 10^{-5} \pm 4.35 \times 10^{-6}$	$1.14 \times 10^1 \pm 1.26$
	20	$5.42 \times 10^{-5} \pm 1.66 \times 10^{-5}$	$1.54 \pm 1.46$
CPSO-H <sub>6</sub>	10	$9.42 \times 10^{-11} \pm 7.58 \times 10^{-11}$	$8.23 \times 10^{-1} \pm 1.04$
	15	$9.57 \times 10^{-12} \pm 7.96 \times 10^{-12}$	$8.12 \times 10^{-1} \pm 1.05$
	20	$2.73 \times 10^{-12} \pm 2.03 \times 10^{-12}$	$1.51 \times 10^{-12} \pm 6.83 \times 10^{-13}$
GA	100	$1.38 \times 10^1 \pm 4.04 \times 10^{-1}$	$1.27 \times 10^1 \pm 1.55$
CCGA	100	$9.51 \times 10^{-2} \pm 3.39 \times 10^{-2}$	$1.57 \times 10^1 \pm 1.87$

表3 Griewank( $f_2$ )函数运行 $2 \times 10^5$ 次后函数值

Table3 Griewank( $f_2$ ) after  $2 \times 10^5$  function evaluation

algorithm	s	mean(unrotated)	mean(rotated)
PSO	10	$9.65 \times 10^{-1} \pm 7.58 \times 10^{-1}$	$3.45 \times 10^{-1} \pm 1.64 \times 10^{-1}$
	15	$2.62 \times 10^{-1} \pm 1.61 \times 10^{-1}$	$1.17 \times 10^{-1} \pm 4.62 \times 10^{-2}$
	20	$6.51 \times 10^{-2} \pm 2.17 \times 10^{-2}$	$9.64 \times 10^{-2} \pm 4.95 \times 10^{-2}$
CPSO-S	10	$2.79 \times 10^{-2} \pm 8.36 \times 10^{-3}$	$5.10 \times 10^{-2} \pm 9.77 \times 10^{-3}$
	15	$2.21 \times 10^{-2} \pm 6.28 \times 10^{-3}$	$5.77 \times 10^{-2} \pm 1.26 \times 10^{-2}$
	20	$2.25 \times 10^{-2} \pm 6.10 \times 10^{-3}$	$6.11 \times 10^{-2} \pm 1.17 \times 10^{-2}$
CPSO-H	10	$2.45 \times 10^{-2} \pm 5.38 \times 10^{-3}$	$5.19 \times 10^{-2} \pm 1.34 \times 10^{-2}$
	15	$2.38 \times 10^{-2} \pm 9.81 \times 10^{-3}$	$5.40 \times 10^{-2} \pm 1.51 \times 10^{-2}$
	20	$1.86 \times 10^{-2} \pm 5.46 \times 10^{-3}$	$4.42 \times 10^{-2} \pm 1.08 \times 10^{-2}$
CPSO-S <sub>6</sub>	10	$7.29 \times 10^{-2} \pm 1.49 \times 10^{-2}$	$6.41 \times 10^{-2} \pm 1.18 \times 10^{-2}$
	15	$6.90 \times 10^{-2} \pm 1.56 \times 10^{-2}$	$7.40 \times 10^{-2} \pm 1.38 \times 10^{-2}$
	20	$8.95 \times 10^{-2} \pm 1.68 \times 10^{-2}$	$5.51 \times 10^{-2} \pm 1.38 \times 10^{-2}$
CPSO-H <sub>6</sub>	10	$6.75 \times 10^{-2} \pm 1.40 \times 10^{-2}$	$4.67 \times 10^{-2} \pm 1.32 \times 10^{-2}$
	15	$5.54 \times 10^{-2} \pm 1.27 \times 10^{-2}$	$3.86 \times 10^{-2} \pm 1.05 \times 10^{-2}$
	20	$5.24 \times 10^{-2} \pm 1.19 \times 10^{-2}$	$4.06 \times 10^{-2} \pm 1.03 \times 10^{-2}$
GA	100	$5.94 \times 10^1 \pm 6.92$	$4.98 \times 10^1 \pm 8.06$
CCGA	100	$2.20 \times 10^{-1} \pm 6.57 \times 10^{-2}$	$1.93 \times 10^{-1} \pm 4.82 \times 10^{-2}$

Ackley函数是一个多峰函数，定位在规则的网格。在旋转情况下，粒子群优化算法陷入局部最小值。从图1可以看出，无论是否旋转，CPSO-H<sub>6</sub>算法能够继续提高其算法解决问题的成效。在旋转情况下，CPSO-S和CPSO-H算法，可以对函数进行优化处理。如果函数是非旋转的，这些“凹陷”是不相关的，每个维度都可以单独搜索，旋转后不再凹陷形成网格与协调轴。这导致搜索难度加大，而CPSO-H<sub>6</sub>算法能够解决这个问题，搜索最优值。

表3表明，在所有Griewank函数实验中，CPSO算法的性能优于PSO算法，图2显示了同样的趋势，不过所有的算法，在第1个10<sup>5</sup>函数评估之后趋于停滞不前。

迭代结果表明，改进PSO算法的表现比GA好。在80%的测试案例中，CPSO比PSO算法收集信息更好，特别是CPSO-S<sub>6</sub>算法可以提高PSO旋转多通道性问题。

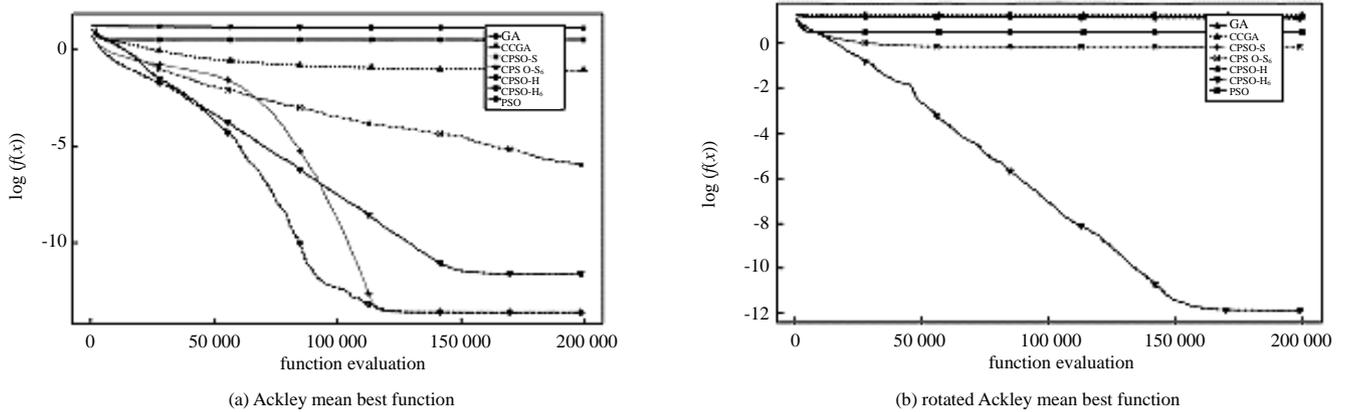


Fig.1 Ackley ( $f_1$ ) mean best function  
图 1 Ackley( $f_1$ )平均值最佳功能

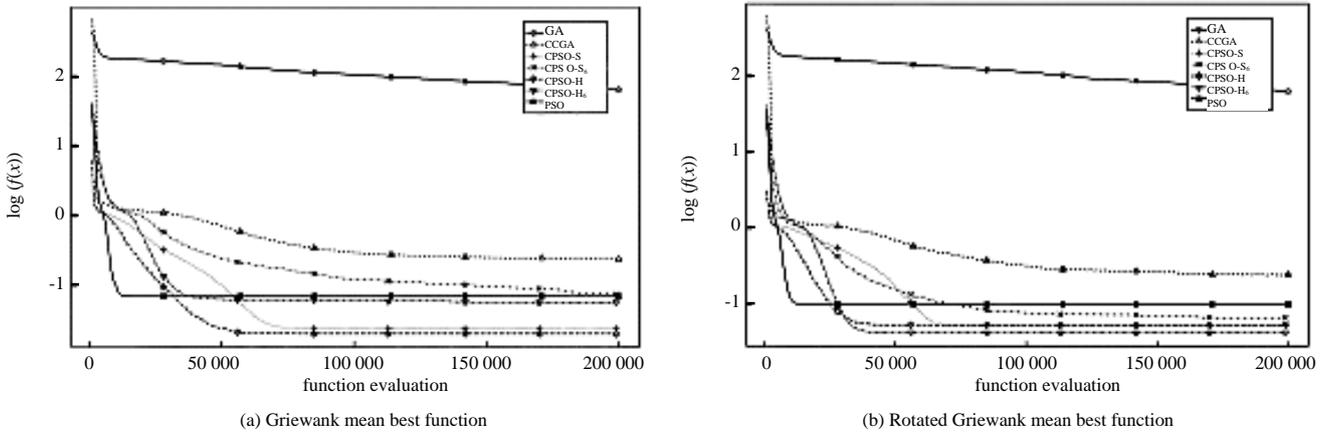


Fig.2 Griewank ( $f_2$ ) mean best function  
图 2 Griewank( $f_2$ ) 平均值最佳功能

### 3.2 鲁棒性

使用鲁棒性和收敛速度作为标准比较各种算法。“鲁棒性”是指算法到达最优函数值迭代次数的多少，在此根据达到指定的阈值时得到的函数值来评估算法的“鲁棒性”，研究参数变化对性能的影响。

表 4 和表 5 中：“success”列的数量(50)表示在低于指定阈值  $2 \times 10^5$  次运行，成功达到最优函数值的次数，“Fn Evals”列表示达到最优函数值最终所需的算法运行的平均值数量。但没有置信区间或标准差函数评估达到阈值所需的数量，原因是该算法达到阈值成功的次数已提供了关于结果的可变性，这意味着一个健壮的算法通常会有一个小标准偏差。

从表 4 中看出，在处理 Ackley 函数时 PSO 算法有一些困难，在旋转函数中 CPOS-S 和 CPOS-H 算法几乎完全失败，但 CPSO-S<sub>6</sub>算法和 CPSO-H<sub>6</sub>算法成功地解决了旋转问题。这表明了合作算法的一个重要的性质：对于不相关的函数，CPSO-S 和 CPSO-H 算法具有速度优势，而在多通道高度相关函数中较差。CPSO-S<sub>K</sub> 和 CPSO-H<sub>K</sub> 算法与 CPSO-S 和 CPSO-H 算法相比，可能收敛较慢，但在很多情况下，比 PSO 算法具有更强的鲁棒性。

从表5可以看到，Griewank函数是难以处理的。在非旋转时，只有CPSO-S和CPSO-H算法始终达到阈值；在

旋转时,算法无法达到最优,相比之下合作算法比PSO算法和GA效果要好。

对于整体的鲁棒性而言,合作算法最好。CPSO-S,CPSO-H算法鲁棒性欠佳,CCGA紧随CPSO-S,CPSO-H算法之后。处理这一系列问题,PSO算法和GA不可靠。这些结果表明,算法在收敛速度和鲁棒性之间有一个权衡。

表4 Ackley( $f_1$ )鲁棒性分析Table4 Ackley( $f_1$ ) robustness analysis

algorithm	s	unrotated		rotated	
		succeeded	Fn Evals	succeeded	Fn Evals
PSO	10	11	2 099	6	1 988
	15	32	3 019	32	3 385
	20	37	2 986	41	3 200
CPSO-S	10	50	935	5	6 240
	15	50	1 053	2	11 644
	20	50	1 227	2	43 314
CPSO-H	10	50	1 068	4	24 420
	15	50	1 154	2	5 836
	20	50	1 245	2	2 401
CPSO-S <sub>6</sub>	10	50	3 264	50	6 670
	15	50	4 136	47	4 533
	20	50	4 994	46	5 686
CPSO-H <sub>6</sub>	10	50	3 105	49	3 494
	15	50	3 924	46	5 355
	20	50	4 947	49	5 657
GA	100	50	100	50	100
CCGA	100	50	100	50	100

表5 Griewank( $f_2$ )鲁棒性分析Table5 Griewank( $f_2$ ) robustness analysis

algorithm	s	unrotated		rotated	
		succeeded	Fn Evals	succeeded	Fn Evals
PSO	10	19	17 521	18	24 081
	15	30	80 666	35	9 095
	20	34	8 405	34	8 620
CPSO-S	10	50	46 963	45	55 532
	15	50	47 174	40	59 911
	20	50	46 679	42	59 389
CPSO-H	10	47	20 170	40	24 374
	15	49	24 183	46	30 257
	20	50	27 121	43	35 715
CPSO-S <sub>6</sub>	10	40	85 580	44	64 311
	15	33	98 075	40	72 844
	20	34	105 770	40	77 259
CPSO-H <sub>6</sub>	10	40	24 445	44	19 478
	15	44	21 063	39	21 282
	20	40	28 577	43	28 099
GA	100	0	N/A	0	N/A
CCGA	100	26	134 056	5	128 545

#### 4 结论

提出了一种合作式方法的粒子群优化框架,促使搜索性能显著改善,尤其是提高了算法的质量和鲁棒性。合作算法增加种群的多样性,并改进了多峰问题的鲁棒性。

与传统的算法相比,合作方式的执行随着问题的维数增加越来越好,这种效应的一种可能的解释是:PSO算法(像大多数其他随机搜索算法一样)在低维搜索空间执行更好,由于指数增加,搜索空间的体积随着维数增加,而粒子的数量少,算法的效率低;大的种群往往有大量的粒子,这时候PSO算法不能较好地解决问题,尤其是在之后的迭代中,粒子数量的增加与体积的增加不匹配,而复合算法将大的搜索空间分解成几个较小的空间,这些亚种群的收敛速度得到提高,它们的子空间的收敛速度明显快于PSO算法在原始维搜索空间中的收敛速度。

#### 参考文献:

- [1] JONG K A D. An analysis of the behavior of a class of genetic adaptive systems[D]. Ann Arbor,USA:University of Michigan, 1975.
- [2] 郭文忠,陈国龙. 离散粒子群优化算法及其应用[M]. 北京:清华大学出版社, 2012. (GUO Wenzhong,CHEN Guolong. Discrete Particle Swarm Optimization Algorithm and Its Application[M]. Beijing:Tsinghua University Press, 2012.)
- [3] 秦昱,蒋平. 基于分层分区协同进化算法的电力系统无功优化[J]. 江苏电机工程, 2008,27(5):6-9. (QIN Yi,JIANG Ping. Power system reactive power optimization based on hierarchical partition cooperative evolutionary algorithm[J]. Jiangsu Electrical Engineering, 2008,27(5):6-9.)
- [4] POTTER M A. The design and analysis of a computational model of cooperative coevolution[J]. Journal of Molecular Liquids, 1997,12(4):11-13.
- [5] FRANS V D B. An analysis of particle swarm optimizers[D]. Pretoria,South Africa:University of Pretoria, 2001.
- [6] 纪震,廖惠连,吴青华. 粒子群算法及应用[M]. 北京:科学出版社, 2009. (JI Zhen,LIAO Huilian,WU Qinghua. Particle Swarm Optimization Algorithm and Its Application[M]. Beijing:Science Press, 2009.)
- [7] 江铭炎,袁东风. 人工鱼群算法及其应用[M]. 北京:科学出版社, 2012. (JIANG Mingyan,YUAN Dongfeng. Artificial Fish Swarm Algorithm and Its Application[M]. Beijing:Science Press, 2012.)
- [8] GREFENSTETTE J J. Deception Considered Harmful[M]. San Mateo,CA:Morgan Kaufmann, 1992.
- [9] 孙俊,方伟,吴小俊,等. 量子行为粒子群优化:原理及其应用[M]. 北京:清华大学出版社, 2011. (SUN Jun,FANG Wei, WU Xiaojun,et al. Quantum Behaved Particle Swarm Optimization:the Principle and Application[M]. Beijing:Tsinghua University Press, 2011.)

- [10] SUN Jun,XU Wenbo,FENG Bin. A global search strategy of quantum-behaved particle swarm optimization[C]// Proc of IEEE Conference on Cybernetics and Intelligent Systems. [S.l.]:IEEE, 2004:111-116.
- [11] 刘波. 粒子群优化算法及其工程应用[M]. 北京:电子工业出版社, 2010. (LIU Bo. Application of Particle Swarm Optimization Algorithm and Its Engineering[M]. Beijing:Publishing House of Electronics Industry, 2010.)
- [12] PHAM M T,ZHANG D,CHANG S K. Multi-guider and cross-searching approach in multi-objective particle swarm optimization for electromagnetic problems[J]. IEEE Trans. Magnetics, 2012,48(2):539-542.
- [13] 李丽,牛奔. 粒子群优化算法[M]. 北京:冶金工业出版社, 2009. (LI Li,NIU Ben. Particle Swarm Optimization Algorithm[M]. Beijing:Metallurgical Industry Press, 2009.)

#### 作者简介:



杜清福(1987-), 男, 河北省张家口市人, 在读硕士研究生, 主要研究方向为电力系统及其自动化. email:dqfgyt@163.com.

(上接第275页)

- [7] 曹广华,赵子滨,张晓川,等. 功能模块分离的无线传感器网络节点设计[J]. 太赫兹科学与电子信息学报, 2012,10(4):441-445. (CAO Guanghua,ZHAO Zibin,ZHANG Xiaochuan,et al. Design of function-modules separated node in wireless sensor network[J]. Journal of Terahertz Science and Electronic Information Technology, 2012,10(4):441-445.)
- [8] 王梓,王瑞辉,左海军,等. 湖南 5 种园林地被植物蒸腾耗水特征初探[J]. 北京林业大学学报, 2009,31(6):139-144. (WANG Zi,WANG Ruihui,ZUO Haijun,et al. Transpiration water-consumption characteristics of five kinds of landscape ground cover plants in Hunan,southern China[J]. Journal of Beijing Forestry University, 2009,31(6):139-144.)
- [9] 陈少勇,郭凯忠,董安祥. 黄土高原土壤湿度变化规律研究[J]. 高原气象, 2008,27(3):530-537. (CHEN Shaoyong,GUO Kaizhong,DONG Anxiang. Research of variety rule of soil humidity in loess plateau of China[J]. Plateau Meteorology, 2008, 27(3):530-537.)

#### 作者简介:



陈 君(1983-), 男, 四川省绵阳市人, 硕士, 工程师, 主要研究方向为无线电通信与信息处理. email:chenhyu66@163.com.

周达夫(1986-), 男, 湖北省武穴市人, 学士, 经济师, 主要研究方向为电气工程、自动化.

王 飞(1989-), 女, 河南省漯河市人, 硕士, 经济师, 主要研究方向为项目管理.

张 骏(1972-), 男, 成都市人, 硕士, 讲师, 主要研究方向为计算机网络和物联网.

汪文勇(1967-), 男, 成都市人, 博士, 教授, 博士生导师, 主要研究方向为计算机网络、下一代互联网.