

求解高维优化问题的改进正弦余弦算法*

徐松金¹, 龙文²

(1. 铜仁学院 大数据学院, 贵州 铜仁 554300; 2. 贵州财经大学 贵州省经济系统仿真重点实验室, 贵阳 550025)

摘要: 提出一种改进的正弦余弦算法(简记为 ISCA)。受粒子群优化(PSO)算法的启发,引入惯性权重以提高正弦余弦算法的收敛精度和加快收敛速度。此外,采取反向学习策略产生初始个体以提高种群的多样性和解的质量。采用八个高维基准测试函数进行仿真实验:在相同的最大适应度函数评价次数下,ISCA 总体性能上均优于基本 SCA 和 HGWO 算法;当维数较高($D=1\,000$)时,ISCA 所用计算量远小于 HDEO 算法。实验结果表明 ISCA 在收敛精度和收敛速度指标上均优于对比算法。

关键词: 正弦余弦算法; 高维优化问题; 反向学习; 惯性权重

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2018)09-2574-04

doi:10.3969/j.issn.1001-3695.2018.09.003

Improved sine cosine algorithm for solving high-dimensional optimization problems

Xu Songjin¹, Long Wen²

(1. School of Data Science, Tongren University, Tongren Guizhou 554300, China; 2. Guizhou Key Laboratory of Economics System Simulation, Guizhou University of Finance & Economics, Guiyang 550025, China)

Abstract: This paper proposed an improved sine cosine algorithm (ISCA) for solving high-dimensional function optimization problems. It inspired by particle swarm optimization (PSO) algorithm, introduced inertia weight to enhance the convergence precision and accelerate the convergence speed. In addition, in order to enhance the diversity of population and solution quality, when producing the initial individuals, it employed the opposite-based learning method. It conducted simulation experiments on the 8 benchmark high-dimensional functions. The compute consumption of ISCA was far less than HDEO in high dimension ($D=1\,000$), and its overall performance was much better than the basic SCA and HGWO algorithm in the same number of maximum fitness function evaluation. The experimental results demonstrate that the proposed ISCA has better performance in convergence precision and convergence speed.

Key words: sine cosine algorithm(SCA); high-dimensional problem; opposite-based learning(OBL); inertia weight

0 引言

群体智能优化算法是一类模拟自然界中群居生物相互作用工作机制而设计出的全局优化方法,在函数优化、工程应用等领域得到了广泛的应用^[1]。典型的群体智能优化算法有粒子群优化(particle swarm optimization, PSO)算法^[2]、人工蜂群(artificial bee colony, ABC)算法^[3]、蚁群优化(ant colony optimization, ACO)算法^[4]、灰狼优化(grey wolf optimization, GWO)算法^[5]、蝙蝠算法(bat algorithm, BA)^[6]、人工萤火虫(artificial firefly swarm, AFS)算法^[7]等。

正弦余弦算法(sine cosine algorithm, SCA)是由澳大利亚学者 Mirjalili^[8]于2016年提出的一种新型群体智能优化算法。SCA 参数设置少,容易实现,仅仅通过正弦余弦函数性质迭代达到寻优目的。虽然 SCA 已被证明在收敛精度和收敛速度方面均优于 PSO、遗传算法(genetic algorithm, GA)和 BA 等^[8],但其依然存在解精度低、易出现早熟收敛等缺点。然而对 SCA 在这些问题上的改进研究国内外仍然尚未展开。

针对基本 SCA 存在的缺点,本文提出一种改进的正弦余弦算法(记为 ISCA)用于求解复杂高维全局优化问题。为了提

高 SCA 的收敛精度和加快收敛速度,受 PSO 算法的启发,在位置更新方程中引入惯性权重;为了提高初始种群的多样性和初始解的质量,利用反向学习策略替代随机方法。八个高维标准测试函数的仿真结果表明,与基本 SCA 相比,ISCA 具有更好的寻优性能。

1 正弦余弦算法

在 SCA 中,假设种群规模为 N ,搜索空间为 d 维,个体 i 在第 d 维空间中的位置可表示为 $X_i = (x_i^1, x_i^2, \dots, x_i^d)$, $i=1, 2, \dots, N$ 。首先,在解空间中随机产生 N 个个体的位置;然后,计算群体中每个个体的适应度值,排序并记录当前最优个体及位置;在每一次迭代中,群体中个体均按式(1)更新位置。

$$X(t+1) = \begin{cases} X(t) + r_1 \times \sin r_2 \times |r_3 \times X^* - X(t)| & r_4 < 0.5 \\ X(t) + r_1 \times \cos r_2 \times |r_3 \times X^* - X(t)| & r_4 \geq 0.5 \end{cases} \quad (1)$$

其中: t 为当前迭代次数; X^* 为当前最优个体位置; $r_2 \in [0, 2\pi]$ 、 $r_3 \in [0, 2]$ 和 $r_4 \in [0, 1]$ 为三个随机参数; r_1 称为控制参数,随迭代次数增加从 a 线性减小到 0,即

$$r_1 = a - a \frac{t}{t_{\max}} \quad (2)$$

收稿日期: 2017-04-19; **修回日期:** 2017-06-19 **基金项目:** 国家自然科学基金资助项目(61463009, 61364003);贵州省科技厅、铜仁市科技局、铜仁学院联合课题(黔科合 LH 字[2015]7248 号);贵州省教育厅创新群体项目(黔教合 KY 字[2016]051)

作者简介: 徐松金(1972-),男,湖南隆回人,副教授,硕士,主要研究方向为智能计算、优化算法(trxyxsj_72@126.com);龙文(1977-),男,湖南隆回人,教授,博导,博士,主要研究方向为智能计算、数据挖掘及工程应用。

其中: $a > 0$ 为常数; t_{\max} 为最大迭代次数。

最后,循环迭代至 SCA 满足终止条件。SCA 伪代码如算法 1 所示。

算法 1 基本 SCA

```

初始化算法参数:种群规模  $N$ , 控制参数  $a$ , 最大迭代次数  $t_{\max}$ 
在解空间中随机初始化  $N$  个个体组成初始种群
计算每个个体的适应度值,并记录最优个体位置
while ( $t < t_{\max}$ )
    for  $i = 1$  to  $N$  do
        for  $j = 1$  to  $d$  do
            根据式(2)计算控制参数  $r_1$  的值
            if  $r_4 < 0.5$ 
                根据式(1)中正弦函数部分更新位置
            else
                根据式(1)中余弦函数部分更新位置
            end for
        end for
    end while

```

2 改进的正弦余弦算法

2.1 反向学习初始化

文献[9]指出,种群初始化在群体智能优化算法中显得尤为重要,原因在于它影响着算法的解精度和收敛速度。由于对问题的全局最优解没有任何先验知识的前提下,标准 SCA 一般采用随机方法生成初始种群。然而随机方法生成的初始群体不能保证能有效地提取解空间的有用信息,从而在一定程度上会影响算法的求解效率。

反向学习 (opposition-based learning, OBL)^[10] 是近年来智能计算领域出现的一种新技术,目前已在 PSO、DE 等群体智能优化算法中得到了成功的应用^[11,12]。因此,本文将反向学习策略嵌入到标准 SCA 中进行种群初始化,为改善算法性能奠定基础。

定义 1 反向点 (opposite point)^[10]。假设在 $[l, u]$ 上存在数 x , 则 x 的反向点定义为 $x' = l + u - x$ 。将反向点的定义扩展到 d 维空间,设 $p = (x_1, x_2, \dots, x_d)$ 为 d 维空间中的一个点,其中 $x_i \in [l_i, u_i], i = 1, 2, \dots, d$, 则其反向点为 $p' = (x'_1, x'_2, \dots, x'_d)$, 其中 $x'_i = l_i + u_i - x_i$ 。

根据上述定义,利用反向学习策略产生初始群体的具体步骤如算法 2 所示。

算法 2 基于反向学习的初始化方法

```

设置种群规模  $N$ 
for  $i = 1$  to  $N$  do
    for  $j = 1$  to  $d$  do
         $X_i^j = l_i + \text{rand}(0, 1) \times (u_i - l_i)$ 
    end for
    for  $i = 1$  to  $N$  do
        for  $j = 1$  to  $d$  do
             $OX_i^j = l_i + u_i - X_i^j$ 
        end for
        合并  $\{X(N) \cup OX(N)\}$ , 从中选取适应度值最好的  $N$  个个体作为初始种群
    end for

```

2.2 修改个体位置更新方程

从式(1)可以看出,在每次迭代中,产生新的候选个体位置仅与当前目标个体位置和全局最优个体位置有关,从而导致标准 SCA 具有局部搜索能力较强而全局搜索能力较弱的缺点,且容易陷入局部最优。因此,对个体位置更新式进行改进是 SCA 研究的一个热点问题。

Shi 等人^[13]在分析了 PSO 算法的不足后引入惯性权重 w , 对粒子的速度更新公式进行改进,使得 PSO 算法能快速收敛于全局最优解。为了提高标准 SCA 的收敛精度和加快收敛速度,受 PSO 算法启发,本文在个体位置更新方程式(1)中引入

惯性权重 w , 得到改进的个体位置更新公式如下:

$$X(t+1) = \begin{cases} wX(t) + r_1 \times \sin r_2 \times |r_3 \times X^* - X(t)| & r_4 < 0.5 \\ wX(t) + r_1 \times \cos r_2 \times |r_3 \times X^* - X(t)| & r_4 \geq 0.5 \end{cases} \quad (3)$$

其中: w 为惯性权重。

由文献[13]可知,较大的惯性权重 w 有利于全局搜索,较小的惯性权重 w 有利于局部搜索。因此,本文提出一种自适应变化的惯性权重策略,即惯性权重随迭代次数增加而线性减小。

$$w(t) = w_{\max} - (w_{\max} - w_{\min}) \times \frac{t}{t_{\max}} \quad (4)$$

其中: t 为当前迭代次数; t_{\max} 为最大迭代次数; w_{\max} 和 w_{\min} 分别为惯性权重的最大值和最小值。由式(4)可以清晰地看出,惯性权重 w 随迭代次数的增加而递减,在进化前期 w 选取较大的值利于全局搜索,在进化后期 w 选取较小的值利于局部精确搜索,从而提高算法的收敛精度和加快算法的收敛速度。

2.3 算法步骤

本文提出的 ISCA 流程如图 1 所示。

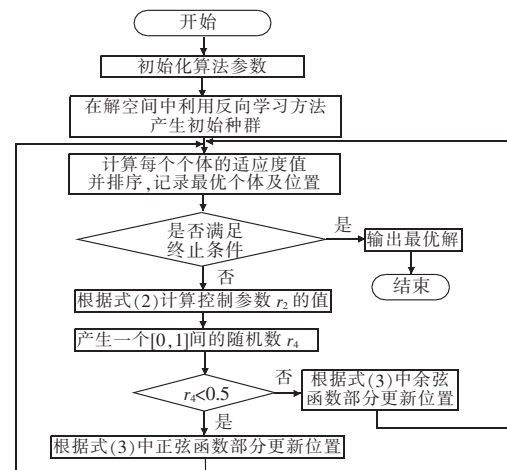


图 1 ISCA 流程

3 仿真实验

3.1 测试函数

为了验证本文提出的 ISCA 的寻优性能,从文献中选取八个高维标准函数进行测试,其具体表达式及变量范围如表 1 所示,八个函数的理论最优值均为 0。

表 1 测试函数

测试函数	变量范围
$f_1(x) = \sum_{i=1}^d x_i^2$	$[-100, 100]$
$f_2(x) = \sum_{i=1}^d x_i + \prod_{i=1}^d x_i $	$[-10, 10]$
$f_3(x) = \sum_{i=1}^d (\sum_{j=1}^d x_j)^2$	$[-100, 100]$
$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	$[-100, 100]$
$f_5(x) = \sum_{i=1}^d (\lfloor x_i + 0.5 \rfloor)^2$	$[-100, 100]$
$f_6(x) = \sum_{i=1}^d (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$
$f_7(x) = -20 \exp\left(-\frac{1}{5} \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]$
$f_8(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$

本文选取寻优结果精确度和寻优成功率两个性能指标来评价算法的性能。寻优结果精确度反映了算法所得结果与问题全局最优解的接近程度;寻优成功率是指多次实验总算法收敛到问题最优解的比例。在某次实验中,如果算法迭代所得的结果对应的精确度小于所设定的收敛精度,则认为算法在该次实验成功收敛到问题的全局最优解。

3.2 实验结果

采用本文所提出的 ISCA 对八个标准测试函数进行求解,并与基本 SCA 进行比较。八个函数的维数均设置为 $D=30$ 、100、500 和 1 000 维,八个高维测试函数的收敛精度均设置为 1×10^{-6} 。为了比较的公平性,ISCA 和 SCA 的种群规模均设置为 50,最大迭代次数设置为 1 000,即最大适应度函数计算次数为 50 000。另外,经过反复多次实验确定 ISCA 其他参数设置如下:惯性权重最大值 $w_{\max}=0.8$,惯性权重最小值 $w_{\min}=0.1$ 。ISCA 和 SCA 分别对八个测试函数单独运行 30 次,记录其最优精度、平均精度、标准差和寻优成功率,结果如表 2 所示。所有仿真实验均在 Intel Core Quad, CPU: Q8300、2 GB 内存、2.50 GHz 主频的计算机上实现,程序采用 MATLAB 7.0 语言实现。

表2 相同迭代次数下 SCA 和 ISCA 的收敛精度结果比较

函数	算法	最优值	平均值	标准差	成功率/%
f_1	SCA	2.75E-007	1.83E-003	1.26E-002	40
	ISCA	0	0	0	100
f_2	SCA	1.02E-009	3.85E-006	7.16E-006	80
	ISCA	0	0	0	100
f_3	SCA	7.48E+001	2.82E+003	2.71E+003	0
	ISCA	0	0	0	100
f_4	SCA	1.25E+000	1.01E+001	8.74E+000	0
	ISCA	0	0	0	100
f_5	SCA	0	0	0	100
	ISCA	0	0	0	100
f_6	SCA	3.11E-005	1.22E+001	1.19E+001	0
	ISCA	0	0	0	100
f_7	SCA	1.36E-004	1.07E+001	9.21E+000	0
	ISCA	8.88E-016	8.88E-016	0	100
f_8	SCA	3.12E-007	2.63E-001	2.59E-001	10
	ISCA	0	0	0	100

从表 2 中结果可知,ISCA 在八个测试函数上 30 次实验均能一致地收敛到全局最优解。尤其是除了函数 f_1 ,ISCA 30 次实验均能收敛到理论最优值 0。与基本 SCA 相比,除了测试函数 f_5 ,ISCA 在七个测试函数上获得了较好的最优值、平均值和标准差;对于函数 f_5 ,ISCA 和 SCA 得到了相似的寻优结果。另外,在寻优成功率方面,ISCA 在八个测试函数上的寻优成功率均为 100%,而基本 SCA 仅在函数 f_5 上的寻优成功率为 100%。

表 3 给出了基本 SCA 和 ISCA 在固定目标精度下的最小收敛和平均收敛代数比较。表中“—”表示算法达到最大迭代

次数(1 000 次)时没有收敛到目标精度值。从表 3 中结果可以看出,与基本 SCA 相比,ISCA 无论是最小收敛代数还是平均收敛代数均要少,这充分说明 ISCA 在收敛速度上具有明显的优势。

表3 两种算法在目标精度下的最小和平均收敛代数的比较($D=30$)

函数	目标精度	最小收敛代数		平均收敛代数	
		SCA	ISCA	SCA	ISCA
f_1	1×10^{-6}	—	199	—	234
f_2	1×10^{-6}	566	241	674	253
f_3	1×10^{-6}	—	399	—	409
f_4	1×10^{-6}	—	310	—	358
f_5	1×10^{-6}	483	201	533	238
f_6	1×10^{-6}	—	261	—	326
f_7	1×10^{-6}	—	252	—	275
f_8	1×10^{-6}	—	247	—	296

图 2 给出了 ISCA 与基本 SCA 对四个测试函数的寻优收敛曲线比较。从图 2 可以清晰地看出,与基本 SCA 相比,ISCA 在四个测试函数上均能获得较高的收敛精度和较快的收敛速度。

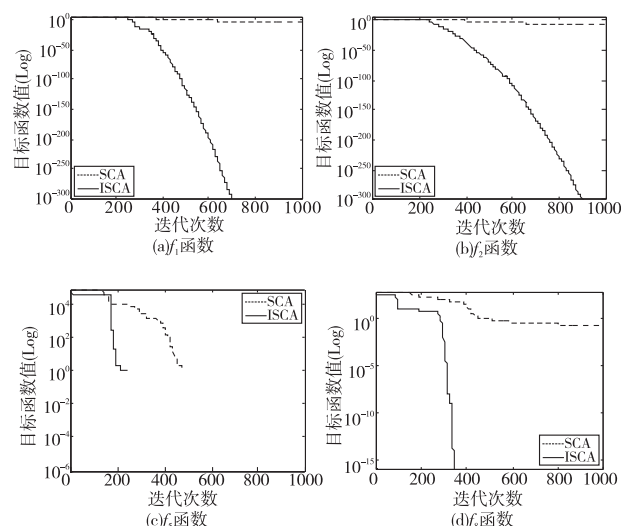


图2 ISCA和SCA对四个函数的收敛曲线

为了进一步验证 ISCA 解决复杂高维优化问题的能力,将八个测试函数的维数分别设置为 $D=100$ 、500 和 1 000 维。需要说明的是,利用 ISCA 在解决 $D=100$ 、500 和 1 000 维优化问题时,其参数设置与 $D=30$ 维相同。表 4 给出了 ISCA 对八个高维测试函数 30 次实验所得的最优值、平均值和标准差。

表4 ISCA 对八个高维函数的测试结果比较

函数	$D=100$ 维			$D=500$ 维			$D=1\,000$ 维		
	最优值	平均值	标准差	最优值	平均值	标准差	最优值	平均值	标准差
f_1	0	0	0	0	0	0	0	0	0
f_2	0	0	0	0	0	0	0	0	0
f_3	0	0	0	0	0	0	0	0	0
f_4	2.65E-312	1.33E-280	0	4.11E-224	2.32E-206	0	2.06E-198	4.28E-192	0
f_5	0	0	0	0	0	0	0	0	0
f_6	0	0	0	0	0	0	0	0	0
f_7	8.88E-016	8.88E-016	0	8.88E-016	8.88E-016	0	8.88E-016	8.88E-016	0
f_8	0	0	0	0	0	0	0	0	0

从表 4 中结果可知,与 $D=30$ 维相比,除了测试函数 f_4 ,ISCA 在其他七个高维测试函数($D=100$ 、 $D=500$ 和 $D=1\,000$ 维)上所获得的结果相同。也就是说,在不增加适应度函数计算次数的前提下,ISCA 在处理高维函数时也具有较好的寻优性能和较强的鲁棒性。对于函数 f_4 ,ISCA 的寻优性能稍有变化。从上述结果可以看出,ISCA 在求解高维优化问题时显示出较强的寻优能力。

图 3 给出了在相同参数设置下 ISCA 求解不同高维($D=$

100、 $D=500$ 和 $D=1\,000$ 维)优化问题时的寻优收敛曲线。从图 3 可以看出,虽然 ISCA 在不同高维问题上最终获得了相同的收敛精度,但其收敛速度稍有不同。

3.3 与其他群体智能优化算法的比较

为了进一步说明 ISCA 的有效性,将其与其他三种不同的智能优化算法,即混合灰狼优化(hybrid grey wolf optimization, HGWO)^[14]、正交反向混合差分进化(orthogonal opposition hybrid differential evolution, HDEO)^[15] 和 composite differential

evolution (CoDE)^[16]算法。HGW0 算法的参数设置为种群规模 $N=50$, 最大迭代次数为 1 000, 即最大适应度函数评价次数为 50 000 次。HDE00 算法的参数设置为种群规模 $N=100$, 缩放因子 $F=0.9$, 交叉概率 $CR=0.9$, 最大适应度函数评价次数为 10 000 000 次。CoDE 算法的参数设置为种群规模为 $N=30$, 最大适应度函数评价次数为 300 000 次。表 5 给出了 HGW0、HDE00、CoDE 算法和 ISCA 对六个高维函数 ($D=1\ 000$ 维) 的平均值 (mean) 和标准差 (Std) 比较。为了比较的可信度, 其他三种算法的结果直接来源于各自参考文献。表中黑体字表示比较算法中的最好结果。

从表 5 可知, 与 HGW0 和 CoDE 算法相比, ISCA 在六个函数上获得了较好的寻优性能; 与 HDE00 算法相比, 除了测试函数 f_7 , ISCA 取得了相似的寻优结果, 但 ISCA 的最大适应度函数评价次数仅为 HDE00 算法的二百分之一; 对于函数 f_7 , ISCA 获得了较好的平均值和标准差。从以上比较结果可知, 对于高维优化问题, ISCA 具有较强的竞争力。

表 5 ISCA 对八个高维函数的收敛精度结果比较

函数	HGW0 算法		HDE00 算法		CoDE 算法		ISCA	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
f_1	4.62E-010	2.73E-010	0	0	9.62E-015	1.03E-014	0	0
f_3	1.81E-005	1.00E-005	0	0	2.68E+004	4.09E+003	0	0
f_5	8.49E+000	2.75E+000	0	0	4.40E+000	1.05E+001	0	0
f_6	3.95E-007	2.52E-007	0	0	2.41E+003	1.63E+002	0	0
f_7	5.45E-007	7.08E-008	4.09E-015	1.08E-015	2.24E+000	1.79E-001	8.88E-016	0
f_8	2.59E-011	1.16E-011	0	0	3.87E-002	1.14E-001	0	0

4 结束语

正弦余弦算法是最近提出的一种新型群体智能优化算法。受粒子群算法的启发, 在个体位置更新公式中引入惯性权重, 利用反向学习策略产生初始种群, 提出一种改进的正弦余弦算法 (ISCA)。通过对八个标准高维测试函数进行数值实验, 与基本 SCA 相比, 本文提出的 ISCA 具有较高的寻优精度和较快的收敛速度。下一步研究方向将 ISCA 应用于多目标优化问题、约束优化问题和实际工程问题中。

参考文献:

- [1] Biswas S, Kundun S, Das S. Inducing niching behavior in differential evolution through local information sharing[J]. *IEEE Trans on Evolutionary Computation*, 2015, 19(2): 246-263.
- [2] 谭跃, 谭冠政, 邓曙光. 基于遗传交叉和多混沌策略改进的粒子群优化算法[J]. *计算机应用研究*, 2016, 33(12): 3643-3647.
- [3] Gao Weifeng, Liu Sanyang, Huang Lingling. A novel artificial bee colony algorithm based on modified search equation and orthogonal learning[J]. *IEEE Trans on Cybernetics*, 2013, 43(3): 1011-1024.
- [4] Yu Bin, Yang Zhongzhen, Yao Baozhen. An improved ant colony optimization for vehicle routing problem[J]. *European Journal of Operational Research*, 2009, 196(1): 171-176.
- [5] 王敏, 唐明珠. 一种新型非线性收敛因子的灰狼优化算法[J]. *计算机应用研究*, 2016, 33(12): 3648-3653.
- [6] He Xingshi, Ding Wenjing, Yang Xinshe. Bat algorithm based on simulated annealing and Gaussian perturbations[J]. *Neural Computing and Applications*, 2014, 25(2): 459-468.
- [7] 徐华丽, 苏守宝, 陈家俊, 等. 变尺度混沌光强吸收系数的萤火虫优化算法[J]. *计算机应用研究*, 2015, 32(2): 368-371.
- [8] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems[J]. *Knowledge-based Systems*, 2016, 96(3): 120-133.
- [9] Gao Weifeng, Liu Sanyang, Huang Lingling. A global best artificial bee colony algorithm for global optimization[J]. *Journal of Computational and Applied Mathematics*, 2012, 236(11): 2741-2753.

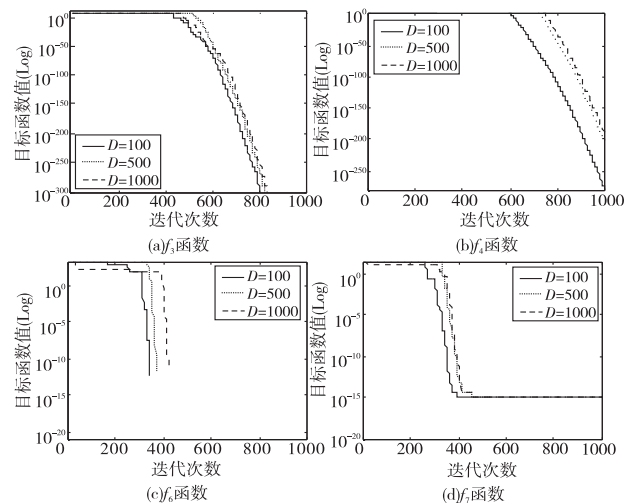


图 3 ISCA 和 SCA 对四个函数不同维数的收敛曲线

- [10] Tizhoosh H R. Opposition-based learning: a new scheme for machine intelligence[C]//Proc of International Conference on Computational Intelligence for Modelling, Control and Automation and IEEE International Conference on Intelligent Agents, Web Technologies and Internet Commerce. Washington DC: IEEE Computer Society, 2005: 695-701.
- [11] Rahnamayan S, Tizhoosh H R, Sakama M A. Opposition-based differential evolution[J]. *IEEE Trans on Evolutionary Computation*, 2008, 12(1): 64-79.
- [12] Wang Hui, Wu Zhijian, Rahamayan S, et al. Enhancing particle swarm optimization using generalized opposition-based learning[J]. *Information Sciences*, 2011, 181(20): 4699-4714.
- [13] Shi Yuhui, Eberhart R. Modified particle swarm optimizer[C]//Proc of IEEE International Conference on Evolutionary Computation. Piscataway, NJ: IEEE Press, 1998: 69-73.
- [14] 龙文, 蔡绍洪, 焦建军, 等. 求解高维优化问题的混合灰狼优化算法[J]. *控制与决策*, 2016, 31(11): 1991-1997.
- [15] 董小刚, 邓长寿, 谭毓澄, 等. 求解大规模优化问题的正交反向混合差分进化算法[J]. *计算机应用研究*, 2016, 33(6): 1656-1661.
- [16] Wang Yong, Cai Zixing, Zhang Qingfu. Differential evolution with composite trial vector generation strategies and control parameters[J]. *IEEE Trans on Evolutionary Computation*, 2011, 15(1): 55-66.

下期要目

- ◆ 深度神经网络的压缩研究
- ◆ 基于词义消歧的卷积神经网络文本分类模型
- ◆ 具有漏泄时滞的随机神经网络的均方指数稳定性
- ◆ 基于模糊聚类和改进混合蛙跳的协同过滤推荐
- ◆ 基于标签优化的协同过滤推荐算法
- ◆ 融合正态分布函数相似度的协同过滤算法
- ◆ 一种面向共享账号的个性化推荐算法
- ◆ 引入偏置的矩阵分解推荐算法研究
- ◆ 零售商品关联大数据稀疏网络的快速聚类算法
- ◆ 汉语语篇的连贯性标注困难度分析研究