

10-14-2020

Natural Computing Method Based on LLE Dimension Reduction

Luyao Zhang

College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China;

Weidong Ji

College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China;

Cheng Hao

College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Natural Computing Method Based on LLE Dimension Reduction

Abstract

Abstract: In the natural computing method, the appearance of high-dimensional problem can make some existing optimization algorithms avoid falling into local optimum, but it makes the performance of the algorithm worse and the running time longer. On the basis of traditional natural calculation method, a *natural calculation method based on LLE(Local Linear Embedding) algorithm is proposed, which analyzes the value of neighbor particle k and dimension d , and makes the algorithm get better optimization effect after dimension reduction.* In the process, a small bias s is added to the data after dimension reduction to increase the diversity of the population. The strategy is applied to PSO and GA respectively, and its performance is verified by using classical test function and four mainstream algorithms for dimension optimization. The experimental results show that the improved algorithm has obvious improvement in solving accuracy and convergence speed.

Keywords

high dimension, natural calculation method, LLE, dimension reduction

Recommended Citation

Zhang Luyao, Ji Weidong, Cheng Hao. Natural Computing Method Based on LLE Dimension Reduction[J]. Journal of System Simulation, 2020, 32(10): 1943-1955.

基于 LLE 降维思想的自然计算方法

张潞瑶, 季伟东, 程昊

(哈尔滨师范大学计算机科学与信息工程学院, 黑龙江 哈尔滨 150025)

摘要: 在自然计算方法中, 高维问题的出现使现有一些优化算法虽然能够避免算法陷入局部最优, 但是使得算法的性能变差、运行时间变长。在传统自然计算方法的基础上, 提出基于 LLE (Local Linear Embedding) 算法的自然计算方法, 通过对算法中邻居粒子 k 和维数 d 的取值进行分析, 降维后使算法得到较好的寻优效果。在此过程中, 将降维后的数据增加一个小偏置 s 来增加种群的多样性。将该策略分别应用于粒子群算法和遗传算法中, 采用经典测试函数以及主流针对维数进行优化的 4 个算法来验证其性能。实验结果表明, 改进的算法在求解精度和收敛速度上均有明显的提升。

关键词: 高维; 自然计算方法; LLE; 降维

中图分类号: TP301

文献标识码: A

文章编号: 1004-731X (2020) 10-1943-13

DOI: 10.16182/j.issn1004731x.joss.20-FZ0328

Natural Computing Method Based on LLE Dimension Reduction

Zhang Luyao, Ji Weidong, Cheng Hao

(College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China)

Abstract: In the natural computing method, the appearance of high-dimensional problem can make some existing optimization algorithms avoid falling into local optimum, but it makes the performance of the algorithm worse and the running time longer. On the basis of traditional natural calculation method, a natural calculation method based on LLE (Local Linear Embedding) algorithm is proposed, which analyzes the value of neighbor particle k and dimension d , and makes the algorithm get better optimization effect after dimension reduction. In the process, a small bias s is added to the data after dimension reduction to increase the diversity of the population. The strategy is applied to PSO and GA respectively, and its performance is verified by using classical test function and four mainstream algorithms for dimension optimization. The experimental results show that the improved algorithm has obvious improvement in solving accuracy and convergence speed.

Keywords: high dimension; natural calculation method; LLE; dimension reduction

引言

自然计算是指受自然界中生物体的启发, 综合自然的计算手段和自然材料, 抽取自然界中相应



收稿日期: 2020-03-26 修回日期: 2020-06-08;
基金项目: 国家自然科学基金(31971015), 哈尔滨市科技局科技创新人才研究专项(2017RAQXJ050);
作者简介: 张潞瑶(1995-), 女, 山西, 硕士生, 研究方向为群体智能; 季伟东(1978-), 黑龙江哈尔滨, 博士, 教授, 研究方向为大数据、群体智能。

的计算模型, 模拟或仿真实现发生在自然界中, 易作为计算的动态解释过程^[1-2]。当运用计算过程的角度分析复杂多变的自然现象时, 可以让人们对于自然界以及计算的本质内容有一个更深刻的理解^[3]。可以说, 自然计算是既人工智能以及云计算之后异军突起的一种独特的计算思维模式^[4]。自然计算研究的领域主要包括人工神经网络、进化算法、群体智能、人工免疫系统、分形几何、人工生

<http://www.china-simulation.com>

• 1943 •

命、DNA 计算和量子计算等^[5]。自然计算领域取得了蓬勃的发展,如利用蚁群算法解决离散优化问题^[6]、通过蚁群个体间的信息传递,运用反馈机制和分布式协作来寻找最优路径,解决 TSP 难题^[7]、粒子群优化(Particle Swarm Optimization, PSO)算法在多目标、动态、约束和组合优化问题中的广泛应用^[8]、以及利用遗传算法(Genetic Algorithm, GA)来解决高维、非线性、不确定性等问题^[9]。

虽然自然计算方法应用的领域众多且模拟的生物机理不尽相同,但是它们都存在一个共同点,即都是基于种群的优化方式^[10]。在种群的优化过程中,个体维数对于算法的优化能力和计算成本存在较大的影响。较高的维数能够减少种群陷入局部最优值的概率,但运行时间长,效率低。相反,较低的维数在提高运行效率的同时,导致种群的多样性降低,容易陷入局部最优值,易早熟收敛。所以,针对不同的问题,如何选择一种恰当的改变维数的方法来平衡算法的有效性和提高运行效率,成为自然计算方法应用中亟需解决的问题。

针对以上问题,已出现了许多关于高维数优化的解决方法。如求解高维复杂连续优化问题的粒子群算法研究,引入分组变异的思想 and 反向学习策略优化高维函数以获取更好性能^[11]。针对种群选择压力随维度增加问题,提出基于动态邻居维度学习的多目标粒子群算法,采用每一维上学习对象不固定的交流方式增加种群多样性^[12]。为保护粒子潜在的优良信息,提出维度学习策略,通过粒子个体最优位置的每一维度分别向群体最优位置对应的维度学习,为粒子构造学习范例^[13]。基于传统优化算法在求解高维非线性优化算法中的问题,提出改进的人工蜂群优化算法^[14],利用高斯分布和自适应差分算法对高维空间进行优化。为解决高维空间中的优化算法和非线性边界约束算法的优化问题,提出综合模式搜索法、最速下降法和转轴法^[15]。针对求解大规模问题,在动态多种群粒子群优化(Dynamic Multi-Swarm-Particle Swarm Optimizer, DMS-PSO)算法^[16]基础上,使用一种有效的分组策

略,采取将高维问题随机重组为低维问题,实现维数和种群双分组,提出协同进化动态粒子群优化(Dynamic Multi-Swarm-Cooperative Coevolution, DMS-CC)算法^[17]。同时,Cheng 等也基于大规模问题,提出社会学习粒子群优化(Social Learning Particle Swarm Optimization, SLPSO)算法和竞争学习(Competitive Swarm Optimizer, CSO)算法^[18-19],SLPSO 采用了与维数相关的参数控制方法,减轻了参数设置的负担问题;CSO 采用一种竞争机制,能够有效解决关于 5 000 维的问题。这些算法在一定程度上平衡了算法的效率和有效性,并且采用维数分组的方式可以解决决策变量多且变量之间相互关联的问题,但是,个体维数大部分都是针对维数本身或者某一维度进行的改进,如将高维随机分组变为低维问题进行求解,某一维向最优位置的维度学习,实现过程依赖于算法的具体步骤,适用性较弱,且效率提升并不明显。本文提出将 LLE 降维思想应用于自然计算领域,该思想与算法采取何种操作无关,因而基于种群优化的自然计算方法具有较好的普适性。同时,把该策略应用到 2 种不同的自然计算方法中,并与目前主流的对维数进行优化的算法进行对比,采用经典测试函数进行实验论证,验证了本文所提的降维思想具有普适性和有效性。

1 LLE 算法降维分析

LLE 是通过局部的线性关系相结合来表示全局非线性结构的一种非线性降维方法^[20-21]。LLE 算法实现降维过程如下:

(1) LLE 算法首先假设在较小局部范围内是线性的。假设 $\mathbf{X}=\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ 是在 R^D 空间中的 N 个样本的数据集。这里, $\mathbf{x}_i \in R^D (i=1, 2, \dots, N)$, D 是数据集的维度。在每个数据点 \mathbf{x}_i 处,使用 \mathbf{x}_i 所选择的 k 个邻居来表示局部线性组成情况,采用欧式距离来表示 \mathbf{x}_i 的邻域。

(2) 由每个 \mathbf{x}_i 的 k 个邻居组成的 \mathbf{x}_i 近似值造成的重构误差,用来计算局部重建权重矩阵,最小化重构误差。优化权重通过式(1)所示的优化方法

来完成。

$$\arg \min_{w_{ij}} = \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{j=1}^k w_{ij} \mathbf{x}_{ij} \right\|_2^2 \quad (1)$$

式中: w_{ij} 为 \mathbf{x}_i 与其第 j 个近邻点 \mathbf{x}_{ij} 间的权值, 且要满足条件 $\sum_{j=1}^k w_{ij} = 1$ 和 $w_{ij} = 0, j \notin \{1, 2, \dots, k\}$ 来计算样本点的局部重建权值矩阵 \mathbf{W} 。

(3) 计算出保证最佳效果的由局部重建的权值矩阵解释局部几何过程中的低维嵌入。

通过权重矩阵 $\mathbf{W} = \{w_i = w_{i1}, w_{i2}, \dots, w_{iN}\}^T$ 完成重构, LLE 算法把 $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ 映射到 $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$, 这里的 \mathbf{Y} 是一个低维的数据空间, 根据式(2), \mathbf{Y} 中保持了高维中的固有属性。

$$\arg \min_{\mathbf{y}_i} = \sum_{i=1}^N \left\| \mathbf{y}_i - \sum_{j=1}^k w_{ij} \mathbf{y}_{ij} \right\|_2^2 \quad (2)$$

式中: \mathbf{y}_i 为 \mathbf{x}_i 的 d 维输出向量 ($d < D$); $\mathbf{y}_{ij} (j=1, 2, \dots, k)$ 为 \mathbf{y}_i 的 k 个近邻点。且要满足 $\sum_{i=1}^N \mathbf{y}_i = 0$ 和 $\frac{1}{N} \sum_{i=1}^N \mathbf{y}_i \mathbf{y}_i^T = \mathbf{I}$, 其中 \mathbf{I} 是 $d \times d$ 的单位矩阵。LLE 算法通过这些过程完成优化, 就可以把 \mathbf{Y} 构造到一个低维数据空间中。

2 基于 LLE 算法降维思想在自然计算领域的应用

2.1 LLE 算法在粒子群算法和遗传算法中的应用

大部分随机算法随着维数增加而使得算法的性能变差, 导致运行时间较长、效率低。因此, 一种合适的降维方法对于算法的优化显得尤为重要。常用的降维方法有主成分分析^[22]、线性判别分析法^[23]、核函数主成分分析法^[24]、局部线性嵌入^[25]、拉普拉斯特征映射^[26]等。其中, LLE 算法是一种崭新的非线性降维方法, 已经被广泛的应用于各种高维数据处理的领域中, 如数据分类、识别, 图像处理, 数据压缩等^[27]。LLE 算法是完全不同于主成分分析法, 可以充分保持数据的非线性结构。所以将 LLE 降维思想应用于自然计算

领域, 如粒子群算法、遗传算法中, 可以解决在高维中粒子收敛速度慢、运行时间长的缺陷。

2.1.1 LLE 算法的种群降维思想

采用 LLE 算法, 进行大小为 N 行 D 列 (N 为粒子数, D 为维数) 的种群降维思想如下:

(1) 输入种群的规模 N , 随机初始化种群位置, 生成一个 $N \times D$ 的矩阵。确定种群中每个粒子 \mathbf{x}_i 与其相邻的 k 个粒子的大小和降维后的特征维数 d ($d < D$)。

(2) 在生成的种群中, 根据粒子和粒子间的欧式距离大小, 确定每一个粒子 \mathbf{x}_i 的 k 个邻居粒子 $\mathbf{x}_{ij} (j=1, 2, \dots, k)$ 。

(3) 求出每个粒子 $\mathbf{x}_i (i=1, 2, \dots, N)$ 关于其 k 个邻居粒子的协方差矩阵 \mathbf{C} , 具体如公式(3)所示:

$$\mathbf{C}_{jm}^{xi} = (\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_m) \quad (3)$$

式中: $\mathbf{x}_i (i=1, 2, \dots, N)$ 为 $1 \times D$ 维的一个位置向量; $\mathbf{x}_j (j=1, 2, \dots, k)$ 和 $\mathbf{x}_m (m=1, 2, \dots, k)$ 为 \mathbf{x}_i 的相邻粒子的 $1 \times D$ 维的位置向量; \mathbf{C}_{jm}^{xi} 代表粒子 \mathbf{x}_i 与相邻的第 j 个粒子、第 m 个粒子的协方差矩阵, 其中 $j \in [1, k], m \in [1, k], j \neq m$ 。

(4) 求解关于粒子 \mathbf{x}_i 的局部线性预测权值矩阵如公式(4)所示:

$$\mathbf{w}_j^{xi} = \frac{\sum_{m=1}^k (\mathbf{C}_{im}^{xi})^{-1}}{\sum_{p=1}^k \sum_{q=1}^k (\mathbf{C}_{pq}^{xi})^{-1}} \quad (4)$$

式中: \mathbf{C}_{pq}^{xi} 为粒子 \mathbf{x}_i 与相邻的第 p 个和第 q 个粒子的协方差矩阵。

(5) 权重系数矩阵 \mathbf{W} 是通过由所有粒子的权重系数组成的向量构成, 计算矩阵 \mathbf{Z} , 如式(5)所示:

$$\mathbf{Z} = (\mathbf{I} - \mathbf{W})(\mathbf{I} - \mathbf{W})^T \quad (5)$$

式中: \mathbf{I} 为 $N \times N$ 的单位矩阵

(6) 计算矩阵 \mathbf{Z} 的特征向量, 并提取前 $d+1$ 个最小特征向量, 即 $D_1 < D_2 < \dots < D_{d+1}$ 。

(7) 由 $D_2 \sim D_{d+1}$ 所组成的特征向量即为种群 N 降维后得到 d ($d < D$) 维的位置数据 \mathbf{A} 。其中 $\mathbf{A} = (D_2, D_3, \dots, D_{d+1})$ 。

(8) 降维后得到的数据为新的种群规模, 将新生成的种群增加一个小偏置 s 以增强种群的多样性, 避免过早陷入局部最优值。

(9) 利用得到的新的种群规模应用于种群的自然计算进化方法之中。

2.1.2 降维思想在自然计算方法中的实现策略

本文将基于 LLE 算法的种群降维思想应用于自然计算领域的粒子群算法和遗传算法中, 算法的实现策略如下:

粒子群算法:

step 1: 初始化种群规模 N , 维数 D 等参数, 随机生成粒子的位置 x_i 和速度 v_i 组成第一代种群 pop ;

step 2: 将生成的种群 pop 采用 LLE 算法的降维思想后, 组成新的种群 $newpop$;

step 3: 对于新种群 $newpop$ 的各维数添加小偏置 s , 更新 $newpop$;

step 4: 计算新种群中所有个体的适应度值 $fitness$, 令 $fitnessbest \leftarrow \min(fitness)$, 求出每个粒子的个体最优值和整个种群的全局最优值;

step 5: 根据标准粒子群的速度、位移公式更新粒子的位置 x_i 和速度 v_i ;

step 6: 若达到算法结束条件, 则返回最优解, 否则, 转 step 4。

遗传算法:

step 1: 初始化种群规模 N , 交叉和变异概率 pc 、 pm , 维数 D 等参数, 随机产生第一代种群 pop ;

step 2: 将生成的种群 pop 采用 LLE 算法的降维思想后, 组成新的种群 $newpop$;

step 3: 对于新种群 $newpop$ 的各维数添加小偏置 s , 更新 $newpop$;

step 4: 评估新种群中个体的适应度值 $fitness$;

step 5: 根据选择、交叉、变异操作产生下一代种群;

step 6: 若达到算法结束条件, 则返回最优解, 否则, 转 step 4。

2.2 LLE 算法使用中有关两个参数的取值分析

在 LLE 算法中, 近邻点 k 的取值采用人为设定方式大大影响了算法的效率和性能, 使得降维的效果不理想, 制约了算法的适用效果^[28]。当近邻点 k 发生改变时, 拓扑矩阵 W 也随之发生改变。在实际应用过程中, 为了避免算法造成运行时间过长、消耗过多的计算资源等缺陷, k 的取值范围不宜过大, 可以适当取小些, 若对数据集有一定的先验认知, 可以简单的将数据集划分成 m 块, 令 $k=N/m$ 即可^[27]。在本文中, 种群规模 N 的取值为 50, 设 m 的值为 5, 故 k 的取值设置为 10。

由于高维空间具有稀疏性的特点, 对于固定的样本容量 N , d 的取值应在一个合理的范围内。考虑数据在 R^D 中的分布, 若将 R^D 的每一维一分为二, 则整个 R^D 的空间被分为 2^D 个区域, 数据集至少在每一个区域中存在一个点。据此, 容易得到在固定样本容量 N 的条件下, 维数估计 d 应满足约束条件 $1 \leq d \leq d_s = \lceil \log_2^N \rceil$ ^[27]。本文中的种群规模 N 为 50, 在约束条件下, d 的取值最大为 6。所以在实验中, 分别取 $d=6$ 和 $d=25$ 进行实验的对比。

3 实验结果与分析

实验仿真平台为 Windows10, Matlab2018a。本实验把基于 LLE 降维思想策略应用到自然计算领域中的 GA 算法^[29]和 PSO 算法^[30]中, 得到 LLEPSO 算法和 LLEGA 算法, 并与 GA、PSO 算法和 4 个维数优化算法做对比。

本实验采用 12 个典型的测试函数: $F_1 \sim F_{12}$ (如表 1 所示)。其中 $F_1 \sim F_4$, F_{11} , F_{12} 是高维单峰函数, 仅存在一个全局最优值; $F_5 \sim F_{10}$ 是高维多峰函数, 这 6 个经典测试函数含有多个局部最优值, 较难找出全局最优值, 在寻优过程很容易陷入局部最优值。其中 F_8 的函数不能少于 4 维, 且全局最小值和维数有关。多峰函数较单峰函数相对复杂, 增大寻优难度。

表 1 经典测试函数
Tab. 1 Classical test function

函数名称	测试函数	维数	取值范围
Sphere F_1	$F_1 = \sum_{i=1}^D x_i^2$	1 000	$[-100,100]^n$
Rosenbrock F_2	$F_2 = \sum_{i=1}^{D-1} [100 * (x_i + 1 - x_i^2)^2 + (x_i - 1)^2]$	1 000	$[-5,10]^n$
dixon-price F_3	$F_3 = (x_1 - 1)^2 + \sum_{i=1}^D i * (2x_i^2 - x_{i-1})^2$	1 000	$[-10,10]^n$
sum of different powers F_4	$F_4 = \sum_{i=1}^D x_i ^{i+1}$	1 000	$[-1,1]^n$
Ackley F_5	$F_5 = -a * \exp\left(-b \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(cx_i)\right) + a + \exp(1);$ $a = 20, b = 0.2, c = 2\pi$	1 000	$[-32.768, 32.768]^n$
Rastrigin F_6	$F_6 = \sum_{i=1}^D (x_i^2 - 10 * \cos(2 * \pi * x_i) + 10)$	1 000	$[-5.12, 5.12]^n$
Griewank F_7	$F_7 = \sum_{i=1}^D \frac{x_i^2}{4000} \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	1 000	$[-600, 600]^n$
Powell F_8	$F_8 = \sum_{i=1}^{D/4} (x_i * (4i - 3) + 10 * x_i * (4i - 2))^2 + 5 \sum_{i=1}^D (x_i * (4i - 1) - x_i * (4i))^2 + \sum_{i=1}^D (x_i * (4i - 2) - 2 * x_i * (4i - 1))^4 + 10 \sum_{i=1}^D (x_i * (4i - 3) - x_i * (4i))^4$	1 000	$[0, \pi]^n$
Levy F_9	$F_9 = \sin^2(\pi \omega_1) + \sum_{i=1}^{D-1} (\omega_i - 1)^2 [1 + 10 \sin^2(\pi \omega_i + 1)] + (\omega_D - 1)^2 [1 + \sin^2(2\pi \omega_D)]; \omega_i = 1 + \frac{\omega_i - 1}{4}$	1 000	$[-10, 10]^n$
Schwefel F_{10}	$F_{10} = 418.9829D - \sum_{i=1}^D x_i \sin(\sqrt{ x_i })$	1 000	$[-500, 500]^n$
sum squares F_{11}	$F_{11} = \sum_{i=1}^D ix_i^2$	1 000	$[-10, 10]^n$
Zakharov F_{12}	$F_{12} = \sum_{i=1}^D x_i^2 + (0.5 \sum_{i=1}^D ix_i^2)^2 + (0.5 \sum_{i=1}^D ix_i^2)^4$	1 000	$[-5, 10]^n$

3.1 LLE 算法降维思想与标准算法的实验结果分析

3.1.1 参数设置

在实验中, GA 算法和 LLEGA 算法的参数设置为: 交叉概率 $P_c=0.70$, 变异概率 $P_m=0.05$, 进化迭代次数 $FEs=1\ 000$, 种群规模 $N=50$, 维数 $D=1\ 000$ 。PSO 算法和 LLEPSO 算法的参数设置为: 学习因子 $c_1=c_2=2$, 惯性权重 $\omega_{max}=0.9$, $\omega_{min}=0.4$, 进化迭代次数 $FEs=1\ 000$, 种群规模 $N=50$, 维数 $D=1\ 000$ 。各算法对表 1 中的 12 个测试函数分别执行 20 次。应用的 LLE 算法, 对于邻居粒子 k 和降

维维数 d 的选择, 文中 k 取值为 10, 维数 d 分别取值 6 和 25 进行实验的对比。越界粒子本文采用通用的越界处理办法, 将其重新设置为边界值, 如表 1 所示。

3.1.2 实验结果分析

图 1~2 是 12 个测试函数 $F_1 \sim F_{12}$ 分别在降维后维数 $d=25$ 和 $d=6$ 维以及降维前时的收敛图。横坐标代表进化次数, 纵坐标代表适应度值。表 2 记录了 PSO 和 GA 降维前 $D=1\ 000$ 和应用降维策略后, LLEPSO 和 LLEGA 维数 $d=25$ 的降维后数据对比图。

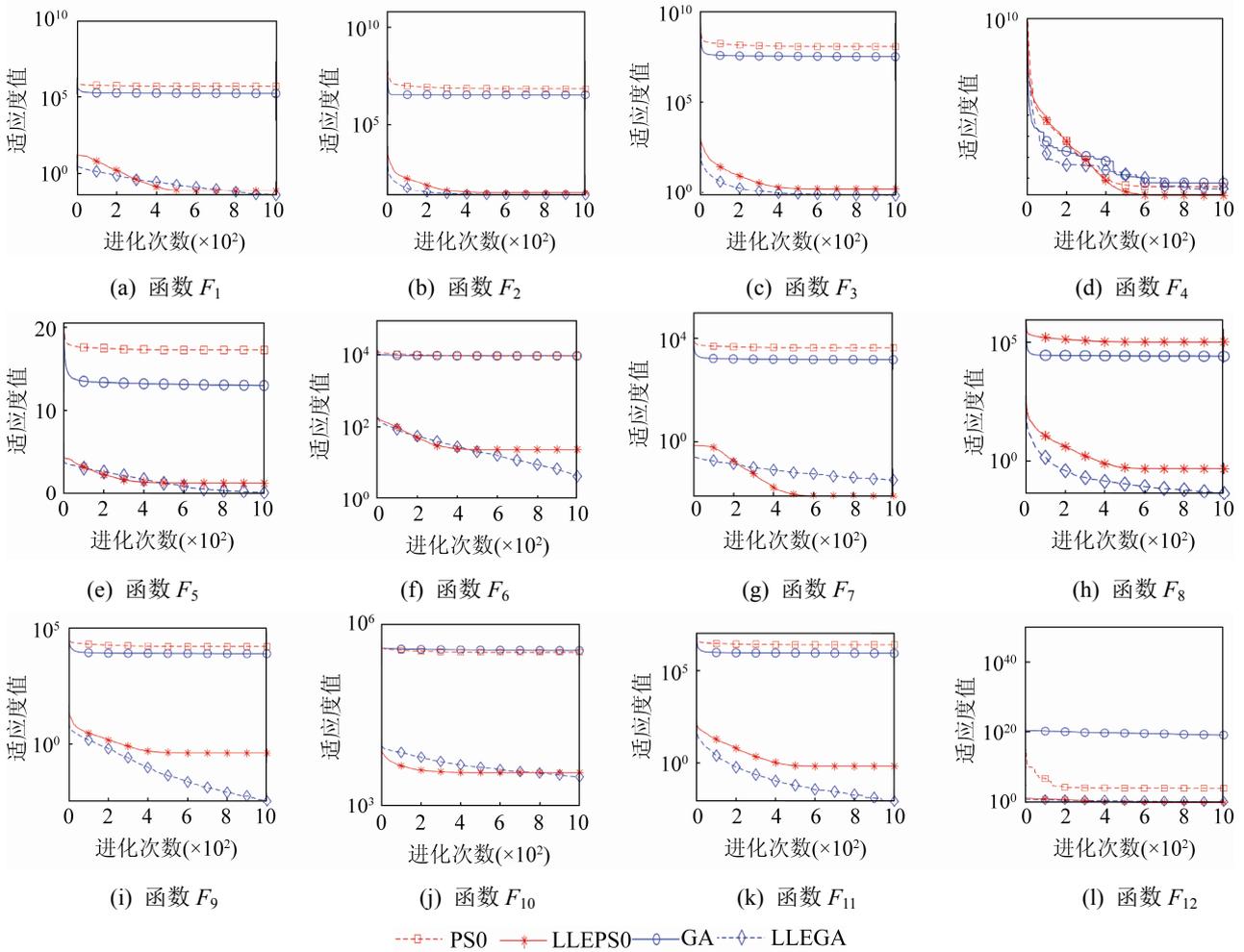
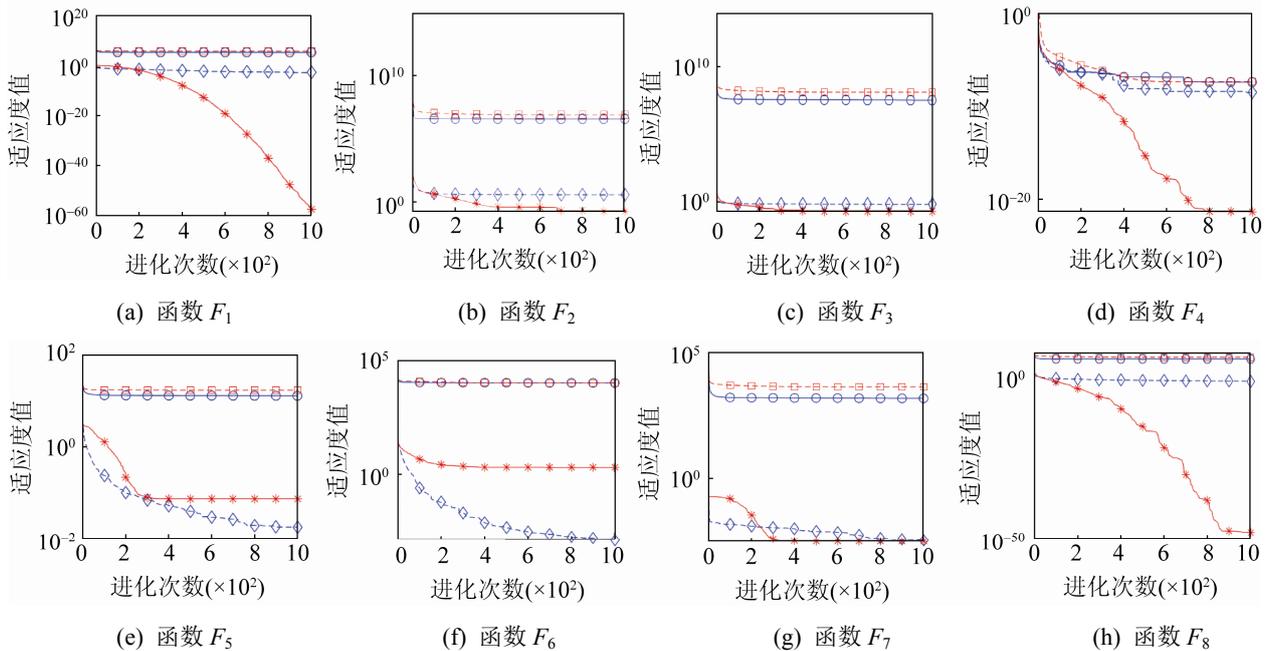


图1 $d=25$ 时, 经典测试函数对算法优化的收敛图

Fig. 1 Convergence graph of classical test functions optimization algorithm, $d=25$



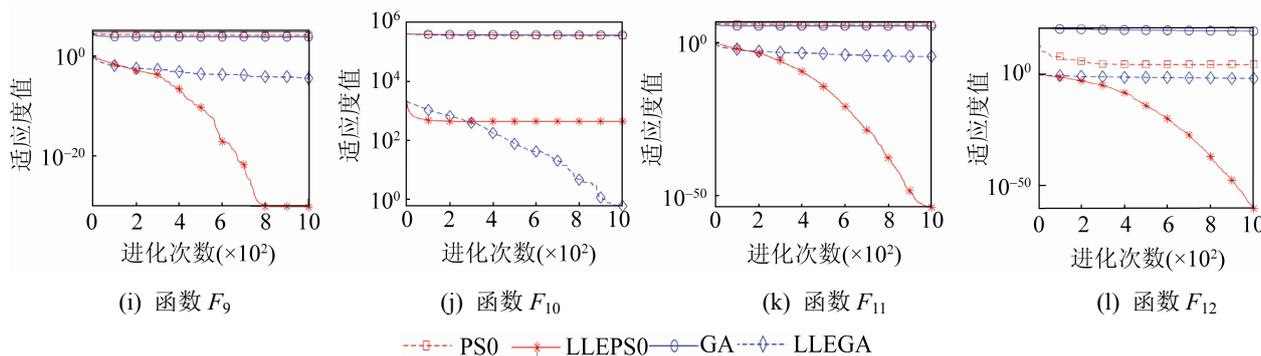


图 2 $d=6$ 时, 经典测试函数对算法优化的收敛图
Fig. 2 Convergence graph of classical test functions optimization algorithm, $d=6$

表 2 降维前后测试函数对算法的优化结果
Tab. 2 Results of algorithm by test function before and after dimension reduction

函数	平均结果及标准方差			
	PSO	LLEPSO	GA	LLEGA
F_1	4.58e+05±4.60e+04	0.071 9±0.054 1	1.69e+05±5.75e+03	0.027 3±0.012 4
F_2	5.82e+06±1.20e+06	25.843 3±2.201 7	3.61e+06±2.53e+05	22.845 8±0.583 8
F_3	1.26e+08±2.79e+07	1.727 9±0.824 1	3.32e+07±2.36e+06	0.678 1±0.007 4
F_4	9.45e-08±1.70e-07	1.549 4e-08±3.552 9e-08	4.51e-08±1.42e-07	1.981 8e-08±6.008 7e-08
F_5	17.147 2±0.448 3	1.028 4±0.370 5	12.9958±0.1261	0.087 7±0.037 3
F_6	9.87e+03±297.25	25.771 4±10.902 7	9.49e+03±156.02	3.024 0±2.966 2
F_7	4.29e+03±372.46	0.006 4±0.001 8	1.49e+03±53.06	0.023 1±0.011 0
F_8	9.87e+04±1.21e+04	0.448 8±0.374 1	2.37e+04±1.23e+03	0.051 0±0.023 3
F_9	1.74e+04±1.49e+03	0.282 7±0.300 1	7.81e+03±302.89	0.002 9±0.001 6
F_{10}	3.46e+05±6.84e+03	3.353 7e+03±743.178 0	3.71e+05±3.21e+03	3.221 6e+03±458.557 7
F_{11}	2.20e+06±2.90e+05	0.721 5±0.442 6	7.73e+05±3.39e+04	0.007 4±0.005 5
F_{12}	1.24e+04±3.85e+03	0.867 6±0.333 2	2.93e+19±2.68e+19	1.526 4±0.692 1

从图 1~2 中可以清楚地看到, 使用 LLE 降维策略的算法比未使用该策略具有更强的全局搜索能力。但在 F_2 、 F_{12} 的测试函数中, LLEPSO 和 LLEGA 有一定的差别。在不同维数的比较中, 维数较低的对于维数较高的算法相对有更好的全局搜索能力。

根据收敛图的曲线走势, 当 $d=25$ 时, LLEPSO 在函数 F_1 , F_3 , F_4 , F_5 , F_6 , F_7 , F_8 , F_{11} 中大概进化 500 次之后, 曲线达到水平状态, 在函数 F_2 , F_9 和 F_{10} 中大概进化 400 次之后曲线保持水平, 适应度值不再有下降的趋势, 说明此时种群陷入局部最优值, 这是由于 PSO 算法具有更强的局部搜索能力导致的; 而标准 PSO 算法在进化 100 次之后陷入局部最优值, 且一直保持水平状态, 说明在应用

了降维策略之后确实使算法具有了更强的全局搜索能力。当 $d=6$ 时, 所用维数满足文献的约束条件, LLEPSO 算法在函数 F_1 , F_{11} , F_{12} 这 3 个测试函数进化 1 000 次之后仍具有下降的趋势, 说明该维数下算法没有被局部最优值所牵绊, 在函数 F_4 , F_8 , F_9 中, 搜索前期具有很好的全局搜索能力, 避免种群陷入局部最优。但在函数 F_{10} 中, LLEPSO 虽然比 PSO 的全局搜索能力强, 收敛效果并没有特别明显, 这是由于测试函数本身的性质决定的, F_{10} 是一个较复杂的函数且有存在许多局部极小值, 这样就导致算法极易早熟。这两种维数的 LLEPSO 算法较其它标准算法相比能够接近全局最优, 尤其是在测试函数 F_1 , F_4 , F_5 , F_7 , F_9 和 F_{11} 上, 存在误差允许的范围, 该函数值已经找到全局最优值。

维数相对较低的算法几乎优于维数较高的算法。

而 LLEGA 算法中, 当 $d=25$ 时, 几乎在每个测试函数中进化 1 000 次之后仍具有继续下降的趋势, 说明算法没有陷入局部最优值, 但是 GA 算法大概在进化 100 次之后陷入局部最优, 这也更好地证明了改进后的算法具有更强的全局搜索能力。当 $d=6$ 时, 在函数 $F_1, F_2, F_3, F_8, F_{12}$ 中下降趋势不明显, 劣于维数为 25 时的 LLEGA 算法, 但其余函数下降趋势特别明显, 总体算法的性能好于标准遗传算法。测试函数 F_4 中 LLEGA 和 GA 曲线几乎重合, 这是由于与 F_4 的曲线平滑度有关, 使算法极易找到全局最优解。

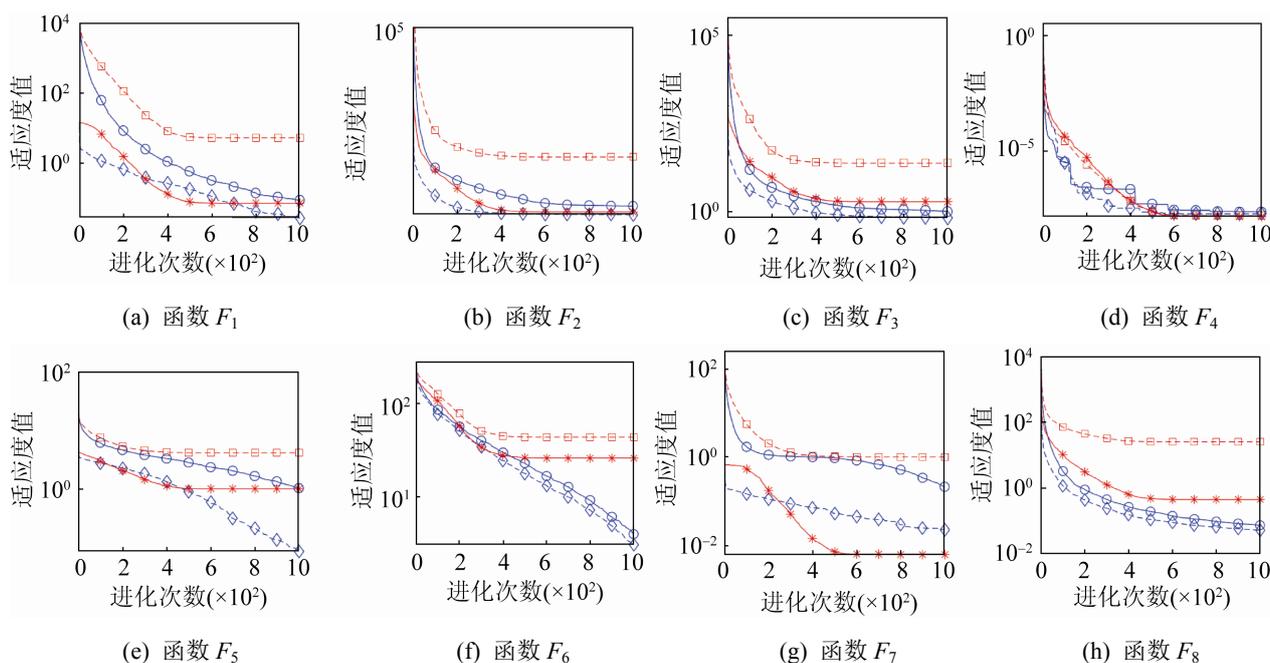
图 3 表示采用 LLE 降维策略将维数 $D=1\ 000$ 降为维数 $d=25$ 时与标准粒子群算法、标准遗传算法维数 D 设置为 25 时, 进行了对比。在优化的这 12 个测试函数时, LLEPSO 算法和 LLEGA 算法较其他标准算法 $d=25$ 时相比都能够接近全局最优, 且优势明显。根据收敛曲线走势, 当 4 个算法维数设置相同时, 基于降维策略的算法都优于基本粒子群和遗传算法, 并且有更好的全局搜索能力。尤其是在遗传算法中, 迭代进行 1 000 次后, 明显还存在继续下降的趋势, 以便能够继

续迭代寻求最优解。同时, 在误差允许存在的范围内, 函数 $F_1, F_2, F_4, F_5, F_7, F_{12}$, 该函数值已经接近全局最优。

表 3 列出了算法维数相同时平均结果及标准差。与 GA 和 PSO 相比, LLEPSO 和 LLEGA 的标准方差低于其他标准算法, 只有在测试函数 F_{10} 上标准方差劣于标准算法, 这是由于 F_{10} 是一个较复杂的函数。若 PSO 算法和 GA 算法处于高维空间中, 则 LLEPSO 和 LLEGA 算法就存在更明显的优势。总之, LLEPSO 算法和 LLEGA 算法比标准粒子群和遗传算法有较低的标准方差, 证明应用该降维策略的算法具有更好的稳定性。

3.1.3 算法中增加小偏置 s 的实验结果分析

利用降维思想后得到的数据增加一个小偏置 s 来增加种群多样性, 为验证 s 在算法中的作用, 本文将对 LLEPSO 算法以及添加小偏置 s 的 LLEPSO 算法进行对比。在此考虑篇幅问题, 只选用测试函数 F_4, F_7, F_8 来验证算法性能, 实验结果收敛效果如图 4 所示。在 3 个测试函数中, LLEPSO+ s 的寻优能力相对优于 LLEPSO, 说明小偏置 s 的添加有助于算法增加种群多样性, 提升寻优能力。



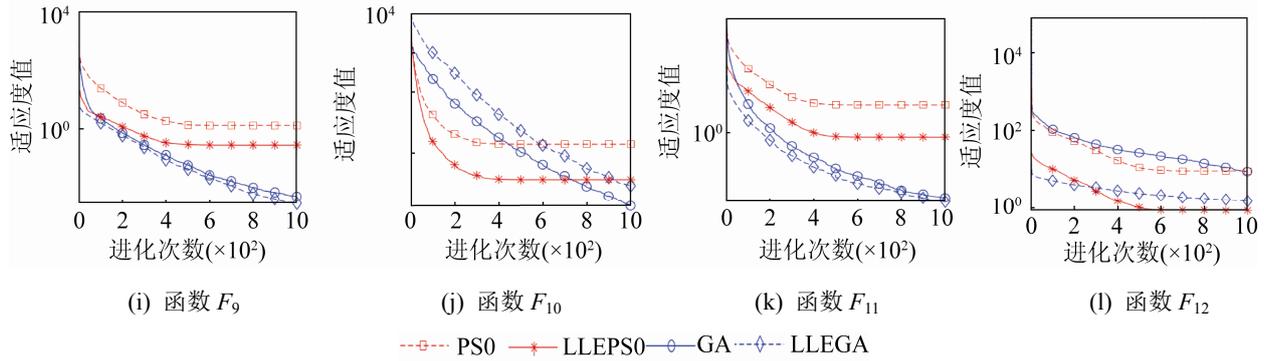


图 3 经典测试函数对维数相同时进行优化的收敛图

Fig. 3 Convergence graph for optimization of classical test functions with same dimension

表 3 12 个经典测试函数对算法的优化结果

Tab. 3 Optimization results of 12 classical test functions on algorithm

函数	平均结果及标准方差			
	PSO	LLEPSO	GA	LLEGA
F_1	5.305 0±3.250 8	0.071 9±0.054 1	0.088 1±0.065 3	0.027 3±0.012 4
F_2	304.378 7±752.023 6	25.843 3±2.201 7	33.742 5±22.275 3	22.845 8±0.583 8
F_3	10.197 0±18.743 6	1.727 9±0.824 1	0.903 7±0.726 2	0.678 1±0.007 4
F_4	1.691 2e-08±2.900 1e-08	1.549 4e-08±3.552 9e-08	2.444 3e-08±5.252 9e-08	1.981 8e-08±6.008 7e-08
F_5	4.264 6±0.885 1	1.028 4±0.370 5	1.056 2±0.767 5	0.087 7±0.037 3
F_6	43.289 8±14.645 4	25.771 4±10.902 7	3.918 3±3.244 5	3.024 0±2.966 2
F_7	1.032 1±0.074 5	0.006 4±0.001 8	0.217 7±0.090 8	0.023 1±0.011 0
F_8	26.169 4±47.045 9	0.448 8±0.374 1	0.073 4±0.023 3	0.051 0±0.023 3
F_9	1.325 2±0.726 8	0.282 7±0.300 1	0.004 9±0.003 0	0.002 9±0.001 6
F_{10}	4.239 2e+03±669.113 7	3.353 7e+03±743.178 0	2.816 8e+03±346.375 3	3.221 6e+03±458.557 7
F_{11}	7.124 3±21.701 1	0.721 5±0.442 6	0.009 6±0.005 8	0.007 4±0.005 5
F_{12}	9.172 3±13.808 5	0.867 6±0.333 2	8.872 8±5.294 8	1.526 4±0.692 1

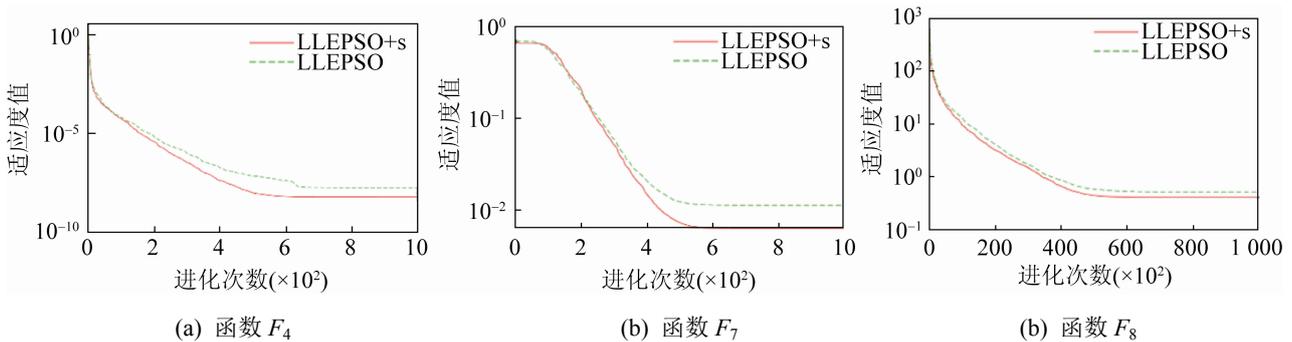


图 4 偏置 s 的算法收敛图

Fig. 4 Convergence graph for bias s

3.2 LLEPSO, LLEGA 和其他应用维数策略算法的实验结果

DMS-CC 算法^[17]、CSO 算法^[19]、SLPSO 算法^[18]以及 DMS-PSO 算法^[16], 不仅在维数的设置

和应用方面与本文有很大的相似性, 而且均为针对求解大规模高维复杂问题而提出的算法, DMS-CC 算法是在 DMS-PSO 算法的基础上进行的改进, 将高维问题随机重组为低维问题对算法

进行优化, SLPSO 算法则采用了与维数相对应的控制方法, CSO 算法提出的竞争机制可以有效的适用于高维空间, 实验表明 CSO 算法可以有效解决 5000 维的问题, 而文中提出的降维策略也是针对高维问题做出的改进。将本文所提出的基于 LLE 降维思想的策略与文献[16-19]中的算法对比, 测试函数的维数选取均为 1 000 维, 其余参数规模的选取与文献[17]中均保持一致, 降维策略中选取维数 d 的取值是 25。为了与文献中测试保持一致, 因此, 采用 F_1, F_4, F_5, F_6, F_7 和 F_{10} 这 6 个测试函数, 其中“-”表示文献中没有此函数数据, 结果如表 4 所示。从表 4 中可以直观地看到, LLEPSO 和 LLEGA 在 F_4, F_5, F_6, F_7 和 F_{10} 这 5 个测试函数中的寻优结果优于其他 4 个算法。利用 LLE 算法的降维思想, 可以使高维相对复杂的算法转换为求解低维问题的算法, 在通过降维后, 同时加入一个小偏置 s , 来增加种群多样性。这样就使得种群在不丧失多样性的前提下降低了维数, 避免了种群陷入局部最优, 提升了算法的效率。而其他 4 个算法没有改变种群的维数, 只是针对大规模高维复杂问题中维数本身做了相应的改进, 其种群内部的冗余信息极易使搜索陷入局部最优。在测试函数 F_1 中, 利用降维策略的算

法在寻优能力上劣于其它算法, 这是因为测试函数本身的性质所决的, 函数内部之间相关性较小, 导致同等程度下, 寻优能力强的算法使结果更优。但综合各算法的对比结果, 表明本文提出的降维思想在自然计算中的应用具有一定的有效性。

在 CSO 算法^[19]中, 实验表明该算法可以有效解决 5000 维问题, 于是将本文提出的 LLE 降维思想在自然计算领域的应用与 CSO 算法进行对比, 初始时, 种群规模 N 设置为 50, 维数 $D=5\ 000$, 降维后维数 d 的取值为 25, 为了与文献中保持一致, 选取 F_1, F_2, F_5, F_6 和 F_7 这 5 个测试函数, 实验结果表明, LLEPSO 在函数 F_2, F_6 这两个测试函数中优于 CSO, 函数 F_1, F_5, F_7 劣于 CSO 算法, 结果如表 5 所示, 造成这种结果可能是由于种群规模过小造成的, 而 CSO 算法中种群规模 N 的取值为 1 500, 所以在 F_1, F_5, F_7 这 3 个函数中, 种群规模由初始的 50 设置为 $N=1\ 500$ 进行对比, LLEPSO 算法收敛图如图 5 所示, 算法对比结果如表 6 所示。由表中数据可知, 在函数 F_1, F_5, F_7 中, 种群规模提升后, 函数收敛效果好于 CSO 算法, 说明利用降维策略可以使复杂的高维特性转换为低维问题求解, 在提高效率的同时, 也有更好的求解精度。

表 4 各算法的最优值实验对比结果

Tab. 4 Optimal value experimental comparison results of each algorithm

算法名称	F_1	F_4	F_5	F_6	F_7	F_{10}
DMS-CC	13.2	5.88e+05	1.60e+14	2.30e+06	2.20e+04	2.00e+11
SLPSO	4.78e-14	3.82e+06	1.90e+14	5.15e+09	9.42e+06	-
DMS-PSO	1.66e+01	1.14e+08	4.36e+07	1.68e+09	9.23e+04	4.51e+10
CSO	2.43e-09	3.71e+06	4.61e+14	3.25e+09	9.68e+06	2.13e+11
LLEPSO	0.012 3	9.451 1e-11	0.230 4	15.485 0	8.824 4e-04	2.676 8e+03
LLEGA	0.013 1	2.249 1e-12	0.050 1	0.227 8	0.011 8	2.395 1e+03

表 5 N 为 50 时, 算法最优值对比结果Tab. 5 $N=50$, algorithm optimal value comparison results

算法名称	F_1	F_2	F_5	F_6	F_7
CSO	1.43e-19	7.30e+03	6.86e-12	7.80e+03	4.44e-16
LLEPSO	0.0188	19.34	0.2607	13.08	0.0021

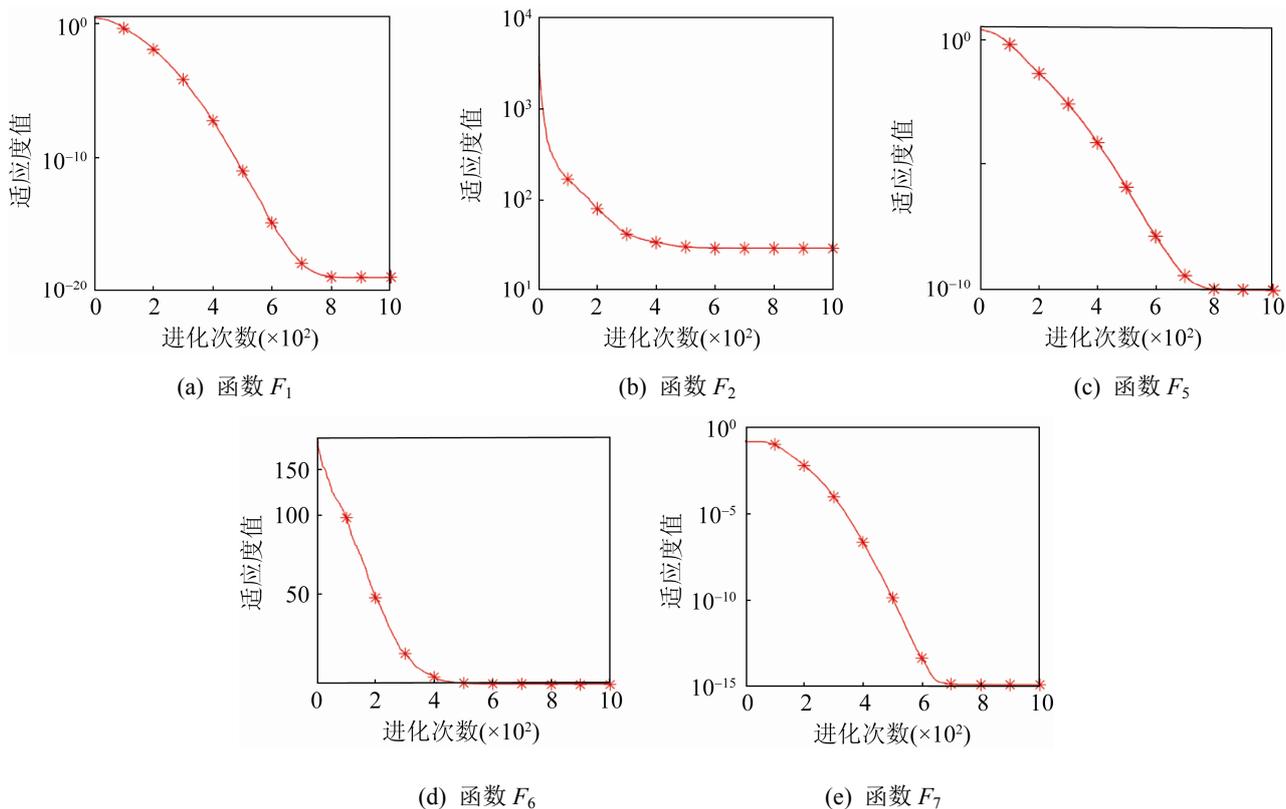


图 5 LLEPSO 算法效果收敛图
Fig. 5 LLEPSO algorithm effect convergence diagram

表 6 维数为 5 000 时, 算法最优值对比结果
Tab. 6 $D=5\ 000$, algorithm optimal value comparison results

算法名称	F_1	F_2	F_5	F_6	F_7
CSO	1.43e-19	7.30e+03	6.86e-12	7.80e+03	4.44e-16
LLEPSO	2.86e-22	19.34	2.80e-12	13.08	2.22e-16

4 结论

本文在标准算法的基础上, 针对高维测试函数的复杂性会使算法寻优能力减弱, 运行时间较长这些问题, 利用局部线性嵌入的思想, 提出了基于 LLE 降维思想的自然计算方法, 并给出了具体的降维操作步骤。同时, 在本算法对种群增加了一个小偏置 s , 来起到增加种群多样性的作用。通过 12 个测试函数证明, 应用该策略的算法与没有应用该策略的算法相比具有更快的收敛速度和更好的求解精度, 说明该策略是一种可以有效改变维数的方法来平衡算法的有效性和提升算法效率。但是, 本策略的不足之处在于关于维数 d 和邻居粒子

k 的选择缺乏一个行之有效的方法, 可能会使算法达不到最理想的效果。在后续研究中, 可以对维数 d 和邻居粒子 k 的选择进行探讨, 当取何值时, 可以使算法达到最好的寻优效果从而更好地增加种群多样性和全局性, 让算法有更好的性能。

参考文献:

[1] Martín-Vide C, Vega-Rodríguez M A. Theory and practice of natural computing: fifth edition[J]. Soft Computing - A Fusion of Foundations, Methodologies and Applications (S1432-7643), 2019, 23(5): 1421.
 [2] 张葛祥, 潘林强. 自然计算的新分支——膜计算[J]. 计算机学报, 2010, 33(2): 208-214.
 Zhang Gexiang, Pan Linqiang. A new branch of natural computing -- membrane computing[J]. Journal of

- Computer Science, 2010, 33(2): 208-214.
- [3] 康琦, 安静, 汪镭, 等. 自然计算的研究综述[J]. 电子学报, 2012, 40(3): 548-558.
Kang Qi, An Jing, Wang Lei, et al. Research summary of natural computation[J]. Journal of Electronic, 2012, 40(3): 548-558.
- [4] Rabinovich Z L. Natural Thinking Mechanisms and Computer Intelligence[J]. Cybernetics & Systems Analysis (S1060-0396), 2003, 39(5): 695-700.
- [5] Castro L N D. Fundamentals of natural computing: an overview[J]. Physics of Life Reviews (S1571-0645), 2007, 4(1): 1-36.
- [6] Bonabeau E, Théraulaz G. Swarm smarts[J]. Scientific American(S0036-8733), 2000, 282(3): 72-79.
- [7] 段海滨. 蚁群算法原理及其应用[M]. 北京: 科学出版社, 2005.
Duan Haibin. Principle and application of ant colony algorithm[M]. Beijing: Science Press, 2005.
- [8] Zhang L, Zhou C, Liu X, et al. Solving multi objective optimization problems using particle swarm optimization[C]. The 2003 Congress on Evolutionary Computation, 2003. Canbella, ACT, Australia: IEEE, 2003: 2400-2405.
- [9] Yuan Q, Qian F, Du W. A hybrid genetic algorithm with the Baldwin effect[J]. Information Sciences (S0020-0255), 2010, 180(5): 640-652.
- [10] 王蓉芳, 焦李成, 刘芳, 等. 自适应动态控制种群规模的自然计算方法[J]. 软件学报, 2012, 23(7): 1760-1772.
Wang Rongfang, Jiao Licheng, Liu Fang, et al. Nature computation with self-adaptive dynamic control strategy of population size[J]. Journal of Software, 2012, 23(7): 1760-1772.
- [11] 潘允敬. 求解高维复杂连续优化问题的粒子群算法研究[J]. 江西理工大学学报, 2018, 39(3): 102-106.
Pan Yunjing. Particle swarm optimization for complex continuous optimization problems[J]. Journal of Jiangxi University of Science and Technology, 2018, 39(3): 102-106.
- [12] 肖闪丽, 王宇嘉, 聂善坤. 动态邻居维度学习的多目标粒子群算法[J]. 计算机工程与应用, 2017, 53(20): 31-37, 60.
Xiao Shanli, Wang Yujia, Nie Shankun. Multi-objective particle swarm optimization for dynamic neighbor dimension learning[J]. Computer Engineering and Application, 2017, 53(20): 31-37, 60.
- [13] 徐桂萍. 基于维度学习策略的粒子群算法的研究与应用[D]. 长春: 吉林大学, 2019.
Xu Guiping. Research and application of particle swarm optimization based on dimension learning strategy[D]. Changchun: Jilin University, 2019.
- [14] 拓守恒. 一种基于人工蜂群的高维非线性优化算法[J]. 微电子学与计算机, 2012, 29(7): 42-46.
Tuo Shouheng. A New High-Dimensional Nonlinear Optimization Algorithm Based on Artificial Bee Colony[J]. Microelectronics & Computer, 2012, 29(7): 42-46.
- [15] 全亚民, 刘大勇, 邹良剑. 非线性边界和等式约束条件下的高维函数优化算法研究[J]. 科研信息化技术与应用, 2013, 4(5): 10-17.
Quan Yamin, Liu Dayong, Zou Liangjian. Optimization algorithm of high-dimensional functions under nonlinear boundary and equality constraints[J]. Information Technology and Application in Scientific Research, 2013, 4(5): 10-17.
- [16] Sabar N R, Jemal A, Yearwood J, et al. Heterogeneous cooperative co-evolution memetic differential evolution algorithm for big data optimization problems[J]. IEEE Transactions on Evolutionary Computation (S1089-778X), 2017, 21(2): 315-327.
- [17] 梁静, 刘睿, 于坤杰, 等. 求解大规模问题协同进化动态粒子群优化算法[J]. 软件学报, 2018, 29(9): 2595-2605.
Liang Jing, Liu Rui, Yu Kunjie, et al. Dynamic multi-swarm particle swarm optimization with cooperative coevolution for large scale global optimization[J]. Journal of Software, 2018, 29(9): 2595-2605.
- [18] Cheng R, Jin Y C. A social learning particle swarm optimization algorithm for scalable optimization[J]. Information Sciences (S0020-0255), 2015, 291(6): 43-60.
- [19] Cheng R, Jin Y C. A competitive swarm optimizer for large scale optimization[J]. IEEE Transactions on Cybernetics (S2168-2267), 2014, 45(2): 191-204.
- [20] Ridder D, Kouropteva O, Okun O. Supervised locally linear embedding[C]. Artificial Neural Networks and Neural Information Processing. Istanbul, Turkey: Springer, 2003: 333-341.
- [21] 马瑞, 王家威, 宋亦旭, 等. 基于局部线性嵌入(LLE)非线性降维的多流形学习[J]. 清华大学学报(自然科学版), 2008, 48(4): 582-585.
Ma Rui, Wang Jiawen, Song Yixu, et al. Multi-manifold learning using locally linear embedding(LLE) nonlinear dimensionality reduction[J]. Journal of Tsinghua University (Science and Technology), 2008, 48(4):

- 582-585.
- [22] 刘胜蓝, 闫德勤. 一种新的全局嵌入降维算法[J]. 自动化学报, 2011, 37(7): 828-835.
Liu Shenglan, Yan Deqin. A new global embedding algorithm[J]. Acta Automatica Sinica, 2011, 37(7): 828-835.
- [23] Sahni J, Vidyarthi D. A Cost-Effective Deadline-Constrained Dynamic Scheduling Algorithm for Scientific Workflows in a Cloud Environment[J]. IEEE Transactions on Cloud Computing (S2168-7161), 2015, 34(2): 1-10.
- [24] Sun T, Xiao C, Xu X, et al. An Improved Budget-Deadline Constrained Workflow Scheduling Algorithm on Heterogeneous Resources[C]. IEEE International Conference on Cyber Security & Cloud Computing. New York: IEEE, 2017: 23-29.
- [25] Rowe S T, Lawrence K S. Nonlinear dimensionality reduction by locally linear embedding[J]. Science (S0036-8075), 2000, 290: 2323-2326.
- [26] Scholkopf B, Smola A, Müller K R. Nonlinear component analysis as a kernel eigenvalue problem[J]. Neural Computation (S0899-7667), 1996, 10(5): 1299-1319.
- [27] 刘建. 高维数据的本征维数估计方法研究[D]. 北京: 国防科学技术大学, 2005.
Liu Jian. Study on Eigen dimension estimation method of high-dimensional data[D]. Beijing: National University of Defense Technology, 2005.
- [28] 李博, 杨丹, 雷明, 等. 基于近邻消息传递的自适应局部线性嵌入[J]. 光电子·激光, 2010, 21(5): 772-778.
Li Bo, Yang Dan, Lei Ming, et al. Adaptive local linear embedding based on neighbor messaging[J]. Photoelectron Laser, 2010, 21(5): 772-778.
- [29] 孟凡超, 初佃辉, 李克秋, 等. 基于混合遗传模拟退火算法的 SaaS 构件优化放置[J]. 软件学报, 2016, 27(4): 916-932.
Meng Fanchao, Chu Dianhui, Li Keqiu, et al. Solving SaaS components optimization placement problem with hybrid genetic and simulated annealing algorithm[J]. Journal of Software, 2016, 27(4): 916-932.
- [30] James K, Russell E. Particle swarm optimization[C]. Proceedings of the IEEE International Conference on Neural Networks. Piscataway: IEEE Service Center, 1995: 1942-1948.