

基于维度缺失检测与恢复的协同进化算法

陈昊^{1,2}, 陈园¹, 黎明^{1,2}, 李军华^{1,2}, 张聪炫²

(1. 南昌航空大学信息工程学院, 江西 南昌 330063;

2. 南昌航空大学无损检测技术教育部重点实验室, 江西 南昌 330063)

摘要: 大规模全局优化(LSGO)问题的搜索空间会随维数增加成指数倍增长, 使用进化算法时极易出现维度缺失、进化停滞现象, 检测维度缺失并跳出停滞状态是解决 LSGO 问题的关键所在. 针对 LSGO 问题维度缺失检测复杂度高的问题, 结合协同进化算法, 将高维问题分解成多个低维子问题; 在进化过程中, 利用主成分分析对所有子问题进行维度缺失检测, 在缺失维度方向上进行拓展以达到维度恢复效果, 能有效跳出停滞状态; 进而提出一种基于维度缺失检测与恢复的协同进化算法. 对 CEC2013 的 LSGO 基准函数中不完全可分测试函数仿真实验结果表明, 提出的算法在提高收敛精度和较好维持多样性的同时, 具有较低的复杂度.

关键词: 大规模全局优化问题; 维度缺失; 协同进化算法; 多样性

中图分类号: TP18 文献标识码: A 文章编号: 1000-5781(2021)05-0590-12

doi: 10.13383/j.cnki.jse.2021.05.002

Co-evolutionary algorithm based on dimension monitoring and recovery

Chen Hao^{1,2}, Chen Yuan¹, Li Ming^{1,2}, Li Junhua^{1,2}, Zhang Congxuan²

(1. School of Information Engineering, Nanchang Hangkong University, Nanchang 330063, China;

2. Key Laboratory of Nondestructive Testing, Ministry of Education, Nanchang Hangkong University, Nanchang 330063, China)

Abstract: The search space of large-scale global optimization (LSGO) problem will increase exponentially with dimension. It is very prone to appear dimension losing and evolutionary stagnation. How to effectively detect lost dimensions and break stagnation is the key to solving LSGO problem. To solve the high complexity of LSGO problem dimension monitoring, high-dimensional problems are decomposed into low-dimensional sub-problems based on co-evolution algorithm. In the process of evolution, principal component analysis is used for all sub-problems to detect the lost dimensions, and expanding in the direction of the lost dimension can achieve the dimension recovery, which can effectively break the stagnant state. Then a co-evolution algorithm based on dimension monitoring and recovery is proposed. The experiment is based on the partially additively separable test function in the LSGO benchmark function of CEC2013, and the results show that the proposed algorithm has lower complexity while improving convergence accuracy and better maintaining diversity.

Key words: large-scale global optimization problem; dimension monitoring; co-evolution algorithm; diversity

收稿日期: 2018-11-22; 修订日期: 2019-09-13.

基金项目: 国家自然科学基金资助项目(61961030; 61866025; 61866026); 江西省优势科技创新团队资助项目(2018BCB24-008); 江西省自然科学基金资助项目(20202BAB204036; 20181BAB202025).

1 引言

在实际的工程应用中, 高维、超高维优化问题普遍存在, 决策变量超过 100 维的此类问题被称为大规模优化问题^[1]. 例如资源调度, 交通网络规划等约束弧路径问题通常具有上千维的决策变量^[2]; 生物计算中饱和和系统优化问题的维数约为 $2N(N+1)$, 分量数 $N=50$ 时, 问题规模为 5 100 维.

大规模全局优化(LSGO)问题可以被定义为

$$\text{Min}_{\mathbf{x} \in X} f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$$

其中 $X \subseteq R^n$ 为 n 维的决策空间; $n \geq 100$; $\mathbf{x} = (x_1, x_2, \dots, x_n) \in R^n$ 为决策变量; $f: R^n \rightarrow R$ 表示 n 元函数. LSGO 问题中不同决策变量之间通常存在复杂的耦合关系, 属于不可分问题(non-separable problems)^[2], 如何处理该类复杂大规模优化问题是当前进化计算研究领域的研究热点. 目前, 解决 LSGO 问题的算法主要分为两大类, 非分解算法和协同进化算法, 前者未对高维决策变量分解, 通过改进算法中进化算子以提高算法的性能, 后者通过分解 LSGO 问题, 并对分解后的低维子问题分别进行优化.

非分解算法^[3]通过在进化过程中针对收敛速度慢, 变异策略失效, 初始化种群差等问题进行一系列改进的方法, 以显著增强其在探索期间处理 LSGO 问题的能力. Tizhoosh 等^[4]提出一种基于反方向学习的算法 OBL, 主要思想是对一个可行解, 同时计算并评估其反向解, 从中选择较优的解作为下一代个体, 反向解是指基于搜索空间中心对称的可行解, 可以加快收敛速度. Chu 等^[5]发现在高维空间内, 易出现“种群退化”现象, 因此提出维度缺失的概念, 发现未被样本群体覆盖维度, 并在这些维度两侧生成新的个体代替种群最差个体达到维度恢复效果. Cheng 等^[6]提出一种竞争学习的 PSO 算法, 随机从当前种群中抽取两个个体比较适应度值, 失败者向胜利者学习, 胜利者并不学习直接进入下一次循环. Mahdavi 等^[7]发现中心点附近的点接近未知解的概率要高于其它点, 并提出基于中心的正态分布采样, 中央黄金区域和混合随机中心正态分布采样等有效初始化方法.

协同进化算法^[8]通过设计不同的分解策略将 LSGO 问题分解成多个低维子问题, 对子问题分别进行优化, 进而通过合并获得问题的完整解. Potter 等^[9]最早提出静态分组, 将 n 维问题分为 n 个子问题, 或将 n 维问题分为 2 个子问题. 但是, 静态分组不能达到分组的目的, Yang 等^[10]提出一种随机动态分组, 将 n 维问题随机分为 m 个 s 维的子问题, 这种设计可以增加相互关联变量分在一起的概率. 在此基础上, Yao 等^[11]提出将 s 设计为一个集合, 若当前的 s 导致进化停滞, 则在集合中重新随机选择一个 s 值. 由于随机动态分组依旧不能完全区分决策变量, 若能通过在优化过程之前或期间获得问题的特征经验来了解识别变量间的相关性, 根据相关性对问题分组可以增加分组的正确性. Omidvar 等^[12]提出差分分组的分组方法, 对所有的决策变量两两进行相关性计算, 根据获得的相关性进行分组. Mohammad 等^[13]发现大规模优化问题的维度间具有不平衡性, 认为平均计算资源不合理, 提出一种基于变量效应的多层优化框架算法 MOFBVE, 根据灵敏度分析获得每个维度对输出的影响, 并根据获得的影响对维数进行聚类, 计算每类的贡献, 优先优化贡献大的类, 能够合理利用有限资源.

由于进化算法在解决 LSGO 问题时易出现“维度缺失、种群退化”现象, 降低算法的搜索能力. 维度缺失检测与恢复策略可以增加种群多样性, 能够有效跳出“种群退化”现象. 本文针对 LSGO 问题维度缺失检测与恢复复杂度高, 耗时长等问题, 利用协同进化算法框架, 选择差分分组方式, 将高维问题分解成多个低维子问题, 以达到降低复杂度的效果; 对所有子问题进行维度缺失检测, 选择一定数量个体对所有缺失维度进行维度恢复, 并替换种群中适应度较差个体.

2 维度缺失

种群退化现象的发生是由于当前种群中个体在绝大多数维度上出现趋同甚至重叠, 导致种群多样性降低, 因此称这种现象为维度缺失. 如何有效检测缺失维度是解决种群退化现象的关键所在, 主成分分析^[14]也

称主分量分析,旨在利用降维的思想,把多指标转化为少数几个综合指标,其中每个主成分都能够反映原始变量的大部分信息,且所含信息互不重复^[5]. 对于维数为 n , 个体数为 p 的种群 \mathbf{C} , \mathbf{R} 为种群标准化后的协方差矩阵,特征分解 \mathbf{R} 得到特征值 λ 和特征向量 \mathbf{V} , 若存在 λ 小于设定阈值 ε , 则该特征值对应的种群特征向量方向上存在缺失信息,称为维度缺失现象. 令 $\mathbf{C} = [c_{ij}]_{n \times p}$, 对 \mathbf{C} 标准化得到 \mathbf{C}' 如 $\mathbf{C}' = [c'_{ij}]$, 其中 $c'_{ij} = (c_{ij} - \bar{c}_i) / \sqrt{v_i}$, \bar{c}_i 和 v_i 是 \mathbf{C} 中第 i 行元素的均值和方差,标准化可减少实际问题中不同参数单位差异影响.

特征向量对应特征值是该特征向量方向上的方差,每个特征向量的期望方差是总方差的 $1/n$. 因此,如果一个特征向量的方差小于预期方差的 10%, 将其视为一个缺失维度.

特征值的大小代表了矩阵正交化之后所对应特征向量对于整个矩阵的贡献程度,因此特征值越大,对应的特征向量包含的信息量越大. 相反,当特征值 λ_i 小于阈值 ε , 则特征值 λ_i 对应的特征向量 V_i 视为缺失维度. 图1(a)是二维维度缺失图, v_1 和 v_2 为特征向量. 可以发现图中 5 个个体在 v_1 方向上呈线性,所以 v_1 对应的特征值大于 v_2 对应的特征值,当 v_2 对应特征值小于设定阈值,视 v_2 为缺失维度.

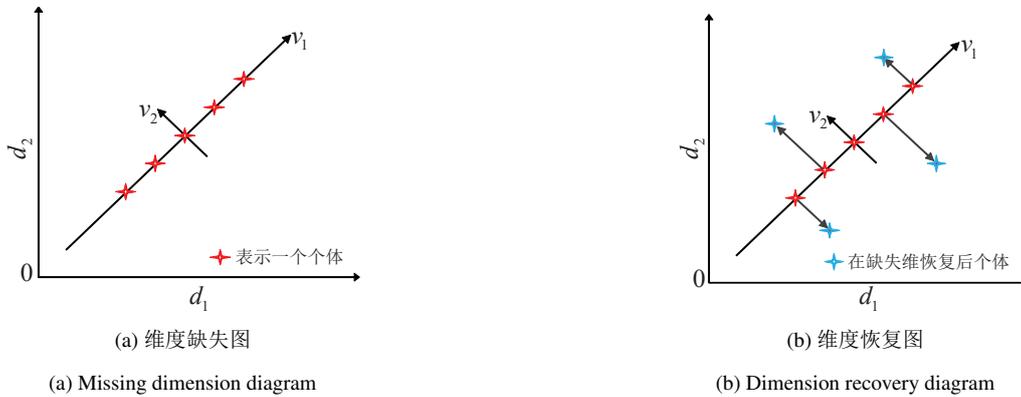


图1 维度缺失检测与恢复

Fig. 1 Dimensional missing detection and recovery

维度缺失检测是基于主成分分析方法的原理,对于一个 n 维问题,当投影到 k 个主成分时复杂度为 $O(kn^2)$,以二分组为例,子问题复杂度为 $O(kn^2/4)$,对于整个问题,复杂度为 $O(kn^2/2)$. 所以当结合分组策略后,组内规模会更小,则维度缺失检测的复杂度更低.

3 基于维度缺失检测与恢复的协同进化算法

由于 LSGO 问题有上千维决策变量,进行维度缺失检测复杂度高,利用分解策略分解高维问题;对分解后的子问题进行维度缺失检测;针对缺失维度,选取种群中较差的 $q\%$ 个体,对所有缺失维度同时进行恢复,选择优异的个体进入下一代. 图2是基于分组的维度缺失检测与恢复示意图,图中第一列是 LSGO 问题的所有维数,其中每一个小矩形代表一个维度,有相同灰度的矩形表示具有相关性;第二列是分组后的所有子问题,每个子问题由具有相关性的维度组成;第三列是对所有子问题进行维度缺失检测;第四列表示进行维度恢复操作后的维度缺失情况,其中无色矩形表示缺失维度.

3.1 分解策略

协同进化算法的关键是分解策略,已有分解策略主要分为静态分组,随机分组,动态分组.

静态分组是最早的分解策略,不考虑维数间相关性,直接将 n 维决策变量硬性分为组内规模为 1 的 n 组子问题或组内规模为 $n/2$ 的 2 组子问题 $n \rightarrow 1(n)$ 或 $n \rightarrow n/2(2)$.

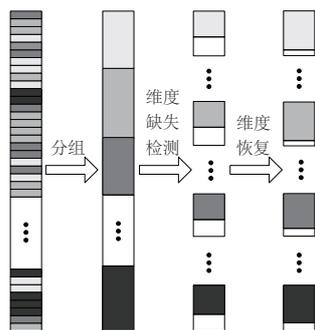


图 2 基于分组的维度缺失检测与恢复示意图

Fig. 2 Schematic diagram of group-based dimension missing detection and recovery

考虑到变量间相关性, 提出随机分组^[12], 将 n 维决策变量随机分为组内规模为 s 的 m 组子问题, 意味着每个变量都有相同的机会被分配到任何子问题中, $n = ms$.

动态分组是通过在优化过程之前或期间获得问题的特征经验来了解识别变量间的相关性, 并根据变量间相关性进行分组. 以差分分组为例, 从第一个变量开始, 分别检测其它变量与第一个变量之间的交互关系, 如果不可分(即具有交互关系), 从所有决策变量中将其排除, 放到一个子成分中, 重复这个过程, 直到所有与第一变量交互的变量都被检测出来, 形成第一个子成分. 对于不可分问题, $\forall a, b_1 \neq b_2, \delta \in \mathbb{R}, \delta \neq 0$, 若满足 $\varepsilon = \sigma/n_i$, 则 x_p 与 x_q 之间有相关性, 为不可分变量, 即

$$\Delta_{\delta, x_p} f(\mathbf{x})|_{x_p=a, x_q=b_1} \neq \Delta_{\delta, x_p} f(\mathbf{x})|_{x_p=a, x_q=b_2},$$

其中 $\Delta_{\delta, x_p} f(\mathbf{x}) = f(\dots, x_p + \delta, \dots) - f(\dots, x_p, \dots)$ 是 f 关于 x_p 间隔 δ 的前向差分. 从静态分组, 随机分组到动态分组, 分组正确性逐渐提升.

3.2 基于分组的维度缺失检测

基于分组的维度缺失检测旨在降低 LSGO 问题维度缺失检测复杂度. 对于 LSGO 问题, 决策变量维数常达 1 000 维以上, 因此, 将高维决策变量分解, 对低维子问题进行维度缺失检测则不存在复杂度高问题. 此处先标准化子种群并计算协方差矩阵, 并对协方差矩阵特征分解, 计算其特征值与特征向量, 当特征值小于阈值 ε 时, 该特征值对应的维度缺失, 阈值为

$$\varepsilon = \sigma/n_i, \tag{1}$$

其中 n_i 为第 i 个子问题维数, σ 为比例系数, $\sigma = 0.01$.

基于分组的维度缺失检测的算法步骤如下:

- 步骤 1 根据子种群生成矩阵 C .
- 步骤 2 对矩阵 C 标准化, 得到矩阵 p .
- 步骤 3 计算 p 的协方差矩阵 R .
- 步骤 4 对进行特征分解的特征值 λ 和特征向量 V .
- 步骤 5 将特征值降序排列, 特征向量也作相应排序.
- 步骤 6 将小于阈值的特征值对应的特征向量视为缺失维度.

3.3 维度恢复

检测出缺失维度后, 用所有特征向量建立新坐标系, 计算种群在新坐标系上的坐标表示, 在新坐标系对种群所有缺失维度进行恢复. 如图 1(b)所示, 以 v_1 和 v_2 为新坐标方向建立坐标系, 选择部分个体在缺失维度 v_2 上进行恢复, 具体步骤如下:

- 步骤 1 对任意子问题, 以特征向量为坐标向量建立坐标系空间 V^{n_i} .

步骤2 将子种群在基坐标空间坐标转换为 V^{n_i} 的坐标, 如

$$t = p(C^{-1})^T, \quad (2)$$

其中 t 为子种群在 V^{n_i} 中的坐标, p 为子种群在基坐标空间中的坐标, C 为过渡矩阵(基变换矩阵).

步骤3 选择 t 中 $q\%$ 较差个体在缺失维度同时进行恢复得到 t' , 对第 j 个体第 k 维度恢复, 即

$$v_{jk} \leftarrow v_{jk} \pm ra_{jk}, \quad (3)$$

其中 a_{jk} 为 $N(2, 1)$ 随机数, r 为子种群最大半径.

步骤4 对恢复后的 t' 转换为基坐标空间中坐标 p' , $p' \leftarrow t'C$.

步骤5 从 p 和 p' 中选择优异个体进入下一代.

维度恢复的算法步骤如下:

步骤1 根据子种群适应度的大小, 对子种群作降序排列.

步骤2 以特征向量为新的基, 建立一个新的坐标系.

步骤3 将种群中的个体转换到新的坐标系.

步骤4 选择适应度较差的部分个体在缺失维度上进行扩展恢复.

步骤5 将恢复后的个体转换到原来的空间. 以任意子种群中符合条件的第 j 个体 p_j 为例, p_j 在基坐标空间中坐标为 $(x_{j1}, x_{j2}, \dots, x_{jn_i})$. 根据式(2)转换为 V^{n_i} 中坐标, 即 $t_j = (v_{j1}, v_{j2}, \dots, v_{jn_i})$, 对个体在缺失维度同时进行扩展后坐标为 $t'_j = (v_{j1}, v_{j2}, \dots, v_{jk} \pm a_{jk}r, \dots, v_{jn_i} \pm a_{jn_i}r)$.

根据式(3)将恢复后的种群坐标转换回基坐标空间坐标, 即 $p'_j = (x_{j1}, x_{j2}, \dots, x_{jn_i})$.

3.4 基于维度缺失检测与恢复的协同进化算法

在对子问题分别进行维度缺失检测并进行维度恢复后, 结合协同进化算法框架, 进一步优化种群, 提出一种基于维度缺失检测与恢复的协同进化算法(co-evolutionary algorithm based on dimension monitoring and recovery, CCDMR). 全局维度恢复的协同进化算法步骤如下:

步骤1 初始化种群和所有参数.

步骤2 任选一种分组策略, 对问题维数进行分组.

步骤3 当算法连续2代进行停滞, 对所有子种群进行维度缺失检测.

步骤4 对所有子种群进行维度恢复.

步骤5 对所有恢复维度的子种群进行恢复.

步骤6 若没有达到终止条件, 返回步骤3.

4 实验与分析

为了分析 CCDMR 算法的性能, 在本文中, 分别对算法的有效性、多样性、参数选择有效性以及收敛精度进行实验比较与分析. 实验中仅限于 CEC2013 LSGO^[16]基准函数(f_{411})等8个不完全可分函数进行实验, 未包含完全可分函数(f_{13}), 重叠函数(f_{1214})和完全不可分函数(f_{15}). 对所有实验, 设置自变量维数 $n = 1000$, 种群规模为100, 进行全局维度恢复的个体数量 q 为50, 最大迭代次数为500. 在 DECC-G 和 CCDMR-G 中设定分组规模 $s = 50$, 每个测试函数独立运行30次, 记录结果的平均值和标准差.

4.1 有效性分析

CCDMR 算法有效性分析选择以静态分组, 随机动态分组和动态学习分组等三种分组方式为代表的 CCGA, DECC-G 和 DECC-DG^[15] 算法进行实验. 比较算法在加入维度缺失检测与恢复算子前后的结果, 算法优化函数均选择差分进化算法, 实验结果见表1, 两两比较将较好值用粗体显示.

表 1 CCDMR 结合三种不同分组方式与 CCGA, DECC-G 和 DECC-DG 优化结果

Table 1 CCDMR combines three different grouping methods with CCGA, DECC-G and DECC-DG optimization results

函数	统计值	CCGA	CCDMR	DECC-G	CCDMR-G	DECC-DG	CCDMR-DG
f_4	均值	5.434 8e+11	1.755 8e+11	1.675 3e+11	2.318 2e+11	2.476 4e+11	5.763 3e+10
	标准差	8.141 0e+10	3.049 1e+11	6.164 8e+10	5.922 3e+10	2.568 2e+11	2.278 6e+10
f_5	均值	1.053 0e+07	5.030 9e+06	4.814 2e+06	3.978 3e+06	8.4010e+06	3.854 7e+06
	标准差	7.568 9e+05	1.345 4e+06	7.413 6e+05	1.920 0e+06	7.971 5e+05	4.741 2e+05
f_6	均值	1.067 3e+06	1.057 0e+06	1.064 0e+06	1.048 4e+06	1.064 5e+06	1.055 9e+06
	标准差	1.124 7e+03	2.758 3e+03	1.899 2e+03	4.386 7e+03	1.375 5e+03	3.135 6e+03
f_7	均值	4.826 1e+09	2.386 6e+09	1.643 1e+09	1.281 7e+09	2.005 2e+09	1.528 7e+09
	标准差	5.693 2e+08	8.739 7e+08	3.438 2e+08	8.450 7e+08	4.901 9e+08	4.870 2e+08
f_8	均值	1.051 4e+16	1.020 0e+15	1.758 7e+15	2.208 7e+15	7.265 7e+14	9.631 0e+14
	标准差	2.548 8e+15	3.489 4e+14	9.500 7e+14	7.636 3e+14	3.864 6e+14	2.039 6e+14
f_9	均值	7.531 5e+08	3.412 2e+08	3.788 5e+08	2.766 9e+08	5.422 7e+08	1.971 2e+08
	标准差	5.315 9e+07	7.001 2e+07	4.494 8e+07	7.549 1e+07	1.399 8e+05	2.703 6e+07
f_{10}	均值	9.537 8e+07	9.427 4e+07	9.410 1e+07	9.379 6e+07	9.487 6e+07	9.343 4e+07
	标准差	2.266 0e+05	7.453 4e+05	1.793 6e+05	2.180 1e+05	1.528 2e+05	5.283 0e+05
f_{11}	均值	3.602 9e+11	3.091 8e+11	3.255 6e+11	2.316 4e+11	3.902 8e+11	1.110 1e+11
	标准差	1.270 1e+11	1.641 4e+11	2.287 2e+11	5.357 4e+10	1.310 5e+11	3.312 6e+10

注:加粗的数字为两两比较的到的较好值。

从表 1 的求解结果均值看,与 CCGA 算法相比,基于静态分组的 CCDMR 在 8 个测试函数上均优于 CCGA;与 DECC-G 算法相比,CCDMR-G 在 6 个测试函数($f_5, f_6, f_7, f_9, f_{10}, f_{11}$)上优于 DECC-G,2 个测试函数(f_4, f_8)上差于 DECC-G;与 DECC-DG 算法相比,CCDMR-DG 在 7 个测试函数($f_4, f_5, f_6, f_7, f_9, f_{10}, f_{11}$)上优于 DECC-DG,1 个测试函数(f_5)上差于 DECC-DG.这说明 CCDMR 算法在求解 LSGO 问题的数据结果要整体优于 CCGA, DECC-G 和 DECC-DG.

比较 CCDMR, CCDMR-G 和 CCDMR-DG 三种算法的数据可以发现,CCDMR 在选定的 8 个测试函数上结果普遍表现最差,而 CCDMR-DG 在大部分测试函数中结果要优于 CCDMR-G,仅在测试函数 f_6 和 f_7 上略差于 CCDMR-G.表明在解决 LSGO 问题时,静态分组的分组准确性最差,动态学习分组的分组准确性最好,随机动态分组的分组准确性介于两者之间.

图 3 是各算法在 8 个测试函数上运行 30 次结果平均值的收敛曲线图,横轴为进化代数,纵轴为函数值.

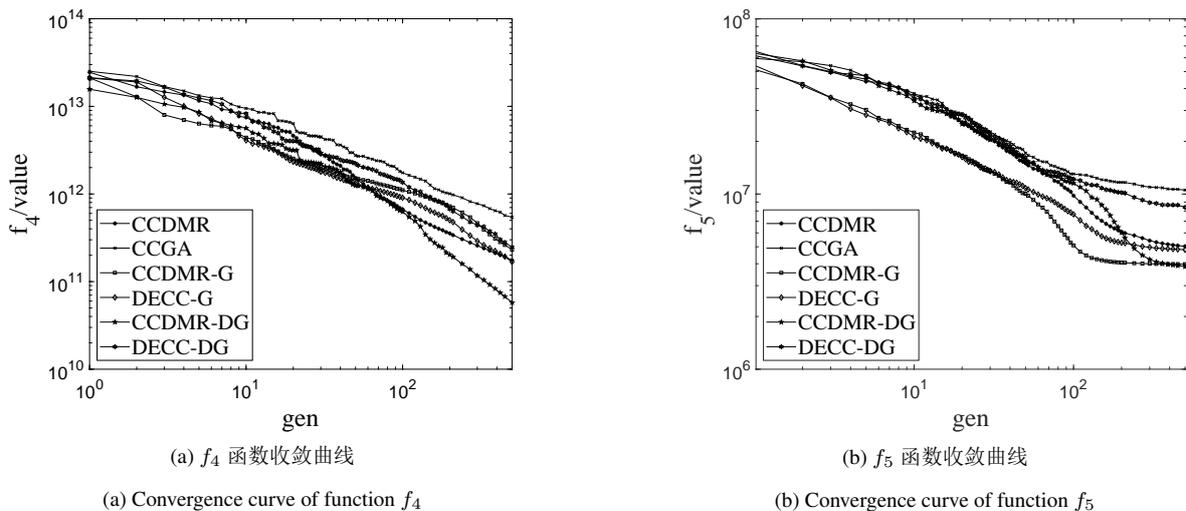
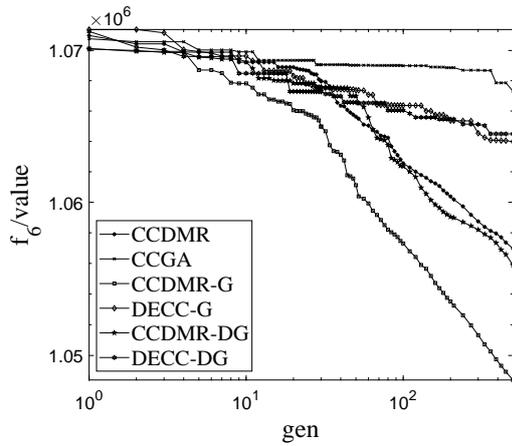


图 3 CCDMR, CCGA, CCDMR-G, DECC-G, CCDMR-DG 和 DECC-DG 算法在 CEC2013 中 $f_4 \sim f_{11}$ 收敛曲线对比

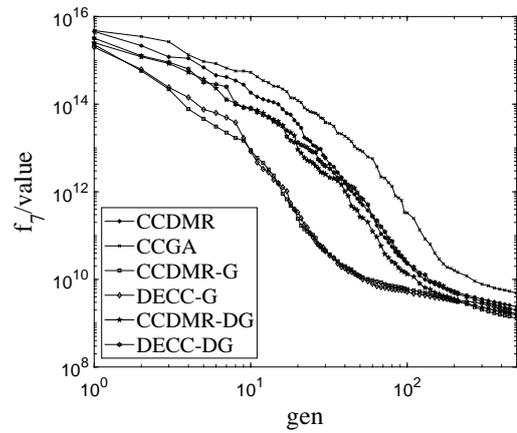
Fig. 3 Comparison of convergence curves of $f_4 \sim f_{11}$ in CEC2013 by CCDMR, CCGA, CCDMR-G,

DECC-G, CCDMR-DG and DECC-DG algorithms



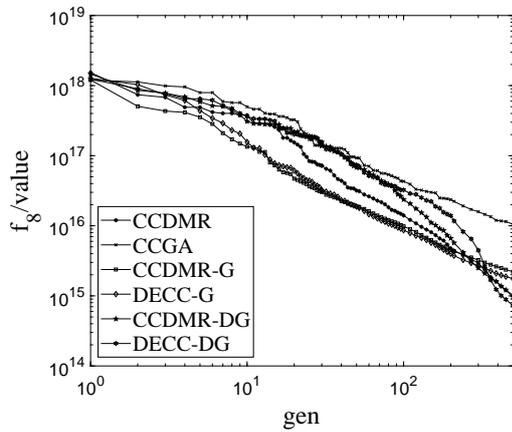
(c) f_6 函数收敛曲线

(c) Convergence curve of function f_6



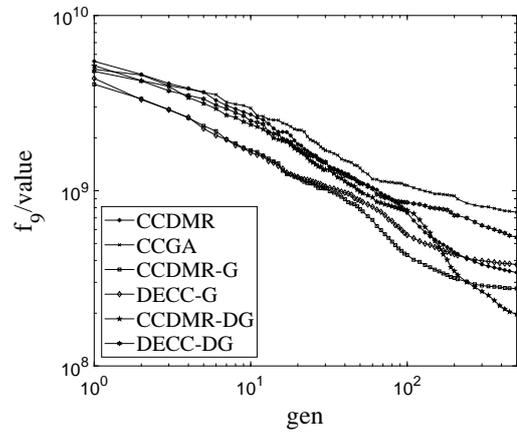
(d) f_7 函数收敛曲线

(d) Convergence curve of function f_7



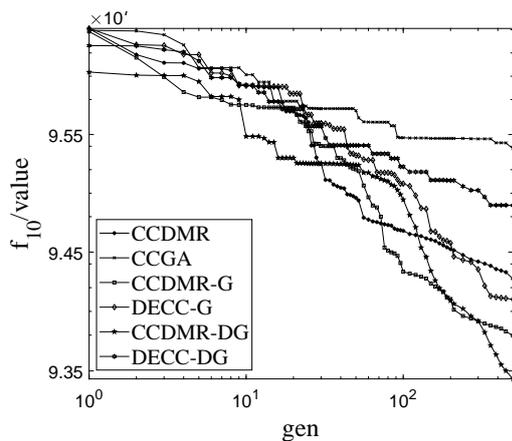
(e) f_8 函数收敛曲线

(e) Convergence curve of function f_8



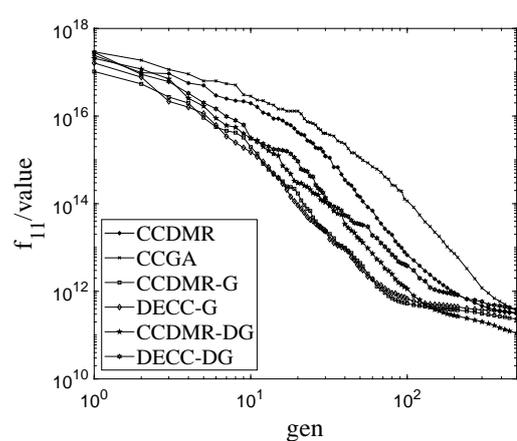
(f) f_9 函数收敛曲线

(f) Convergence curve of function f_9



(g) f_{10} 函数收敛曲线

(g) Convergence curve of function f_{10}



(h) f_{11} 函数收敛曲线

(h) Convergence curve of function f_{11}

续图 3

Fig. 3 Continues

从图 3(a), 图 3(b), 图 3(c), 图 3(d), 图 3(f), 图 3(g), 图 3(h)可以看出在 $f_4, f_5, f_6, f_7, f_9, f_{10}, f_{11}$ 等 7 个测试函数中, 均是本文提出的 CCDMR 结合不同分组方法的 CCDMR-DG 和 CCDMR-G 结果最优, CCGA 结

果最差. 8 个测试函数的求解结果说明 CCDMR 优于未加入维度缺失检测与恢复算子的算法. 由于选择分组方法不同, CCDMR 的优化结果也不同, 从图中可以看出, CCDMR-DG 优于 CCDMR-G 和 CCDMR, 所以动态学习分组要优于随机动态分组和静态分组. 从 8 幅图整体情况来看, 收敛曲线依然支持以上数据分析结果.

图 4 是各算法在 8 个测试函数上运行 30 次结果的 Box Plot 图, 横轴为对比算法, 纵轴为函数值.

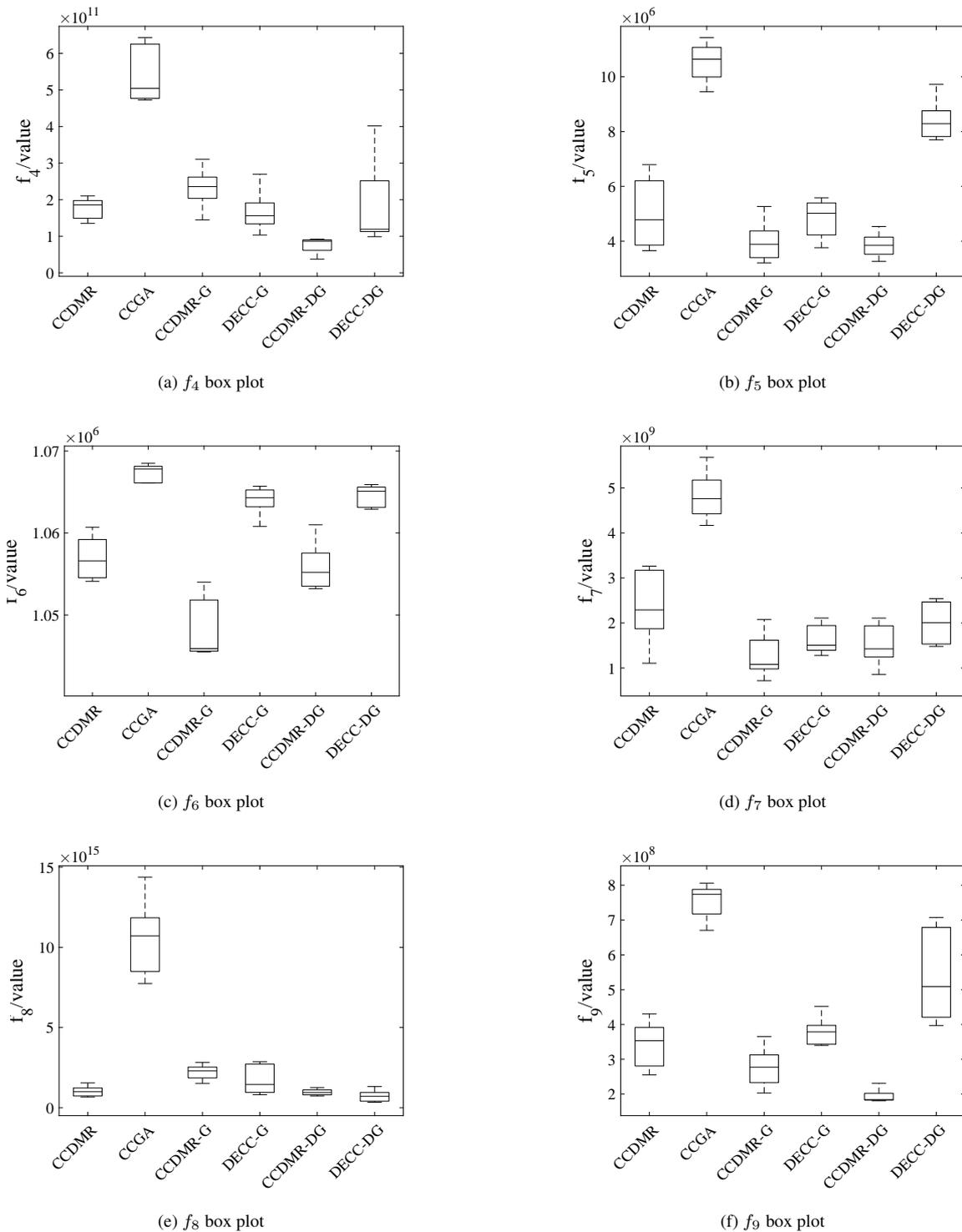
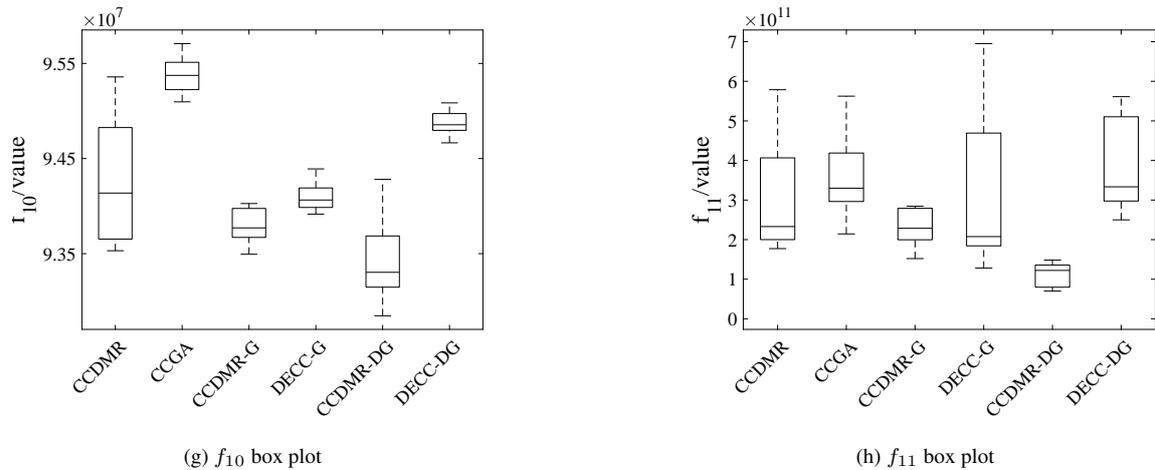


图 4 CCDMR, CCGA, CCDMR-G, DECC-G, CCDMR-DG, DECC-DG 算法在 CEC2013 中 $f_4 \sim f_{11}$ box plot 图

Fig. 4 Box plot of $f_4 \sim f_{11}$ in CEC2013 by CCDMR, CCGA, CCDMR-G, DECC-G, CCDMR-DG and DECC-DG algorithms



续图4

Fig. 4 Continues

从图4(a),图4(b),图4(e),图4(f),图4(h)中可以看出在 $f_4, f_5, f_8, f_9, f_{11}$ 函数中,CCDMR-DG不仅收敛性最好,并且最稳定;对于图4(c),图4(d)中 f_6, f_7 函数结果,CCDMR-G的收敛性最好,DECC-DG最稳定;对于图4(g)中 f_{10} 函数结果,CCDMR-DG的收敛性最好,DECC-DG最稳定.整体来看,维度缺失检测算子和维度恢复算子在各分组情况下对算法收敛性都有提升,不过对于不同测试函数,效果也不一样.

4.2 多样性分析

CCDMR算法在保证收敛的前提下,可以保持种群多样性,比较维度缺失检测与恢复前后种群多样性.此处用种群熵表征种群多样性,分别计算各算法在所选测试函数全局维度恢复前后种群熵的相对偏差,选择不同分组方式,分别为不分组,静态分组,随机分组和差分分组,对应种群熵的相对偏差如表2所示.

表2 CCDMR结合四种不同分组方式种群熵的相对偏差

Table 2 CCDMR combines the relative deviation of population entropy in four different grouping methods

函数	统计值	无分组	静态分组	随机分组	差分分组
f_4	均值	0.203 9	0.386 5	0.766 5	2.176 6
	标准差	0.047 5	0.218 1	0.664 0	1.208 2
f_5	均值	10.288 1	3.730 6	10.184 3	19.516 9
	标准差	1.821 2	0.643 3	8.149 9	4.057 3
f_6	均值	0.009 8	0.014 4	0.011 3	0.000 8
	标准差	0.007 3	0.012 0	0.000 0	0.000 0
f_7	均值	1.556 4	0.545 7	0.932 6	1.412 2
	标准差	0.732 3	0.163 3	0.485 0	0.718 8
f_8	均值	0.201 5	0.107 7	0.169 7	2.664 7
	标准差	0.056 3	0.029 9	0.022 9	3.323 0
f_9	均值	10.044 5	4.576 3	3.903 2	20.946 6
	标准差	2.088 3	0.878 1	2.852 2	1.630 3
f_{10}	均值	0.008 6	0.018 2	0.005 7	0.002 8
	标准差	0.006 3	0.014 2	0.005 2	0.002 3
f_{11}	均值	3.261 7	1.312 8	1.401 5	0.897 5
	标准差	3.381 2	0.724 3	0.742 6	0.297 6

从表2的数据可以发现,全局维度恢复方法结合不同分组方式在所有选取测试函数上种群熵均有所增大,说明种群多样性得以维持.其中 f_5, f_9 函数在CCDMR结合四种不同分组方式上种群熵的相对偏差较大, f_6, f_{10} 函数在CCDMR结合四种不同分组方式上种群熵的相对偏差不明显,因为 f_5, f_9 函数的基函数是单峰函数,而 f_6, f_{10} 函数的基函数是多峰函数,有大量局部最优值.

图 5 是 CCDMR 在 f_5 和 f_{11} 函数全局恢复前后种群熵变化曲线, 横轴为进化过程中出现的停滞次数, 纵轴为种群熵值.

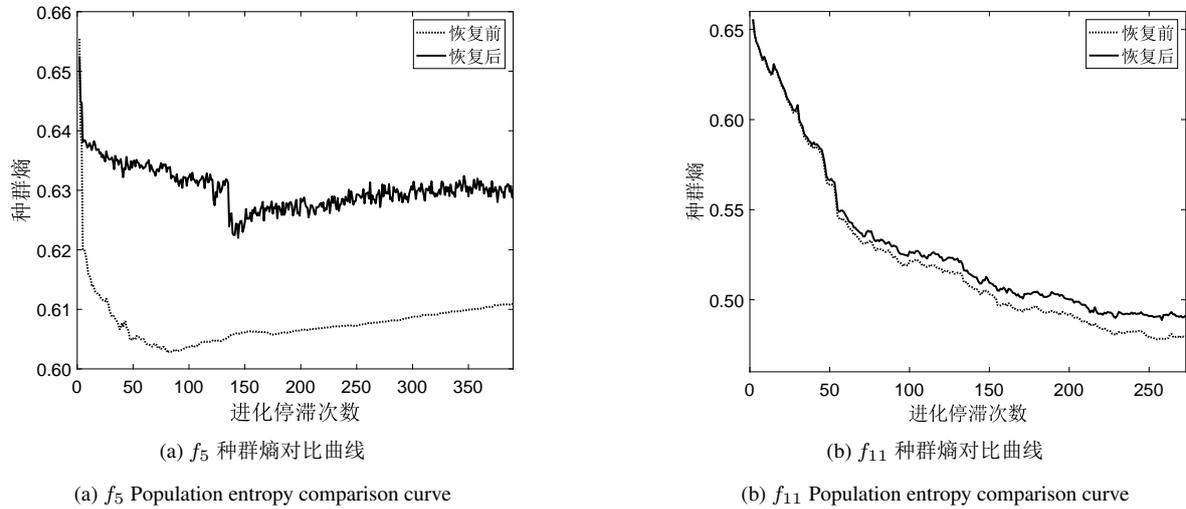


图 5 f_5, f_{11} 全局维度恢复前后种群熵对比曲线图

Fig. 5 f_5, f_{11} global dimension before and after population entropy comparison curve

从图 5(a)中可以看出 f_5 函数恢复后的种群熵明显大于恢复前的种群熵, 且恢复前的种群熵曲线较为平滑, 而恢复后的种群熵曲线出现上下波动. 这是由于算法在迭代过程中, 个体向最优值靠近, 种群熵减小, 导致种群多样性减小, 而全局恢复可以增大种群熵, 从而达到维持种群多样性的效果. 从图 5(b) 中也可以看出种群多样性得以维持.

4.3 比例系数对算法影响分析

比例系数 σ 大小决定维度缺失被检测的程度, 决定需要被恢复的维度, 对算法收敛速度有较大影响. σ 越大, 缺失维度阈值会对应提高, 则有较少维度被视为缺失维度, 反之, 较多维度被视为缺失维度. 对此, 分别对 $\sigma = 0.005, \sigma = 0.05, \sigma = 0.1$ 和 $\sigma = 0.01$ 进行实验分析, 结果对比如表 3 所示

表 3 比例系数对比结果
Table 3 Scale factor comparison results

函数	统计值	$\sigma = 0.005$	$\sigma = 0.05$	$\sigma = 0.1$	$\sigma = 0.01$
f_4	均值	1.739 5e+11	1.421 5e+11	2.591 7e+11	5.763 3e+10
	标准差	8.018 4e+10	3.424 2e+10	2.375 4e+11	2.278 6e+10
f_5	均值	8.920 6e+06	8.333 8e+06	8.310 7e+06	3.854 7e+06
	标准差	1.147 5e+06	5.636 4e+05	2.738 1e+05	4.741 2e+05
f_6	均值	1.064 2e+06	1.063 7e+06	1.064 3e+06	1.055 9e+06
	标准差	1.376 5e+06	7.190 3e+02	1.181 1e+03	3.135 6e+03
f_7	均值	3.379 2e+09	2.370 2e+09	1.670 4e+09	1.528 7e+09
	标准差	3.585 1e+09	6.440 9e+08	4.826 1e+08	4.870 2e+08
f_8	均值	1.3293e+15	1.0142e+15	1.3514e+15	9.6310e+14
	标准差	2.9551e+14	3.633 3e+14	1.111 1e+15	2.039 6e+14
f_9	均值	4.820 4e+08	5.720 2e+08	4.977 1e+08	1.971 2e+08
	标准差	1.523 7e+08	1.308 6e+08	6.619 0e+07	2.703 6e+07
f_{10}	均值	9.479 5e+07	9.479 5e+07	9.466 4e+07	9.343 4e+07
	标准差	1.528 4e+05	3.188 7e+05	1.970 2e+05	5.283 0e+05
f_{11}	均值	2.854 7e+11	2.706 2e+11	1.996 2e+11	1.110 1e+11
	标准差	1.123 6e+11	6.641 1e+10	8.884 3e+10	3.312 6e+10

根据表 3 的结果显示, 当 $\sigma = 0.01$ 时, 在所选择的 8 个测试函数上均获得最优结果. 则当 $\sigma = 0.01$ 时, 维度缺失检测与恢复效果较好, 可以跳出进化停滞概率, 提高算法收敛精度与收敛速度.

4.4 CCDMR与MOFBVE, CBCC3, CC-CNS对比

结合目前研究情况, 选择在 CEC2013 测试函数上进行评估的已有算法 MOFBVE, CBCC3 和 CC-CNS 与之进行比较. 比较结果如表 4 所示, 其中 CCDMR 算法的优化函数选择自适应的基于邻域搜索的差分进化算法(SaNSDE), 分组方式选择差分分组.

表 4 为本文方法数据与文献[7, 13, 17]中的数据对比.

表 4 CCDMR 与 MOFBVE, CBCC3 和 CC-CNS 结果比较

Table 4 Comparison of CCDMR with MOFBVE, CBCC3 and CC-CNS results

函数	统计值	MOFBVE ^[13]	CBCC3 ^[17]	CC-CNS ^[7]	CCDMR
f_4	均值	9.92e+09	4.08e+07	1.11e+11	2.23e+08
	标准差	5.01e+09	2.09e+06	3.65e+10	1.27e+08
f_5	均值	2.77e+06	2.34e+06	6.11e+06	2.16e+06
	标准差	4.19e+05	4.70e+05	4.06e+05	4.01e+05
f_6	均值	9.33e+04	8.65e+04	{8.19e+00	1.05e+06
	标准差	2.91e+04	1.88e+04	7.38e-01	1.42e+03
f_7	均值	5.85e+06	4.75e+07	7.53e+08	4.68e+05
	标准差	2.21e+06	3.38e+07	2.78e+08	9.45e+04
f_8	均值	1.81e+13	1.51e+10	4.08e+15	2.44e+10
	标准差	1.37e+13	2.87e+10	1.69e+15	2.44e+09
f_9	均值	2.65e+08	2.02e+08	5.33e+08	1.52e+08
	标准差	3.16e+07	5.09e+07	2.36e+07	1.43e+07
f_{10}	均值	2.09e+04	7.68e+01	9.61e-03	9.28e+07
	标准差	2.22e+04	1.24e+01	2.03e-03	1.84e+05
f_{11}	均值	4.55e+08	1.33e+09	6.50e+10	3.12e+06
	标准差	3.66e+08	1.41e+09	5.89e+10	6.78e+05

从表 4 中可以看到, 本文方法在 4 个测试函数(f_5, f_7, f_9, f_{11})上优于另外三个算法; CC-CNS 在 2 个测试函数(f_6, f_{10})上优于另外三个算法; CBCC3 在 2 个测试函数(f_4, f_8)上优于另外三个算法. 在所选测试函数中, 本文方法在一半测试函数上优于其它算法, 收敛精度高于其它算法.

5 结束语

为了解决 LSGO 问题中维度缺失、进化停滞问题, 对进化停滞种群进行维度缺失检测与恢复. 由于优化问题搜索空间随着维数增加成指数倍增长, 将维度缺失检测与恢复结合协同进化算法框架, 提出一种 CCDMR 算法. 根据理论分析及仿真结果可知: 1) CCDMR 算法结合协同进化算法框架, 可以降低维度缺失检测与全局维度恢复的复杂度, 加快收敛速度; 2) CCDMR 算法能够避免种群陷入局部最优, 维持种群多样性, 提高了算法的收敛精度.

本文提出的基于维度缺失检测与恢复的协同进化算法选用了已有的三种分组策略, 这三种分组策略尚不能够准确地将高维问题进行分组, 需研究出更加准确的分组策略, 这也是本文后续的研究内容.

参考文献:

- [1] Mahdavi S, Shiri M E, Rahnamayan S. Metaheuristics in large-scale global continues optimization: A survey. Information Sciences, 2015, 295(1): 407–428.
- [2] Sun Y, Kirley M, Halgamuge S K. On the selection of decomposition methods for large scale fully non-separable problems // Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation. ACM, 2015: 1213–1216.

- [3] 付亚平, 王洪峰, 黄敏, 等. 基于自适应多种群策略的混合多目标优化算法. 系统工程学报, 2017, 32(6): 738–748.
Fu Y P, Wang H F, Huang M, et al. Adaptive multipopulation strategy based hybrid multiobjective optimization algorithm. Journal of Systems Engineering, 2017, 32(6): 738–748. (in Chinese)
- [4] Zhou Y, Hao J K, Duval B. Opposition-based memetic search for the maximum diversity problem. IEEE Transactions on Evolutionary Computation, 2017, 21(5): 731–745.
- [5] Chu W, Gao X, Sorooshian S. A new evolutionary search strategy for global optimization of high-dimensional problems. Information Sciences, 2011, 181(22): 4909–4927.
- [6] Cheng R, Jin Y. A social learning particle swarm optimization algorithm for scalable optimization. Information Sciences, 2015, 291(1): 43–60.
- [7] Mahdavi S, Rahnamayan S, Deb K. Center-based initialization of cooperative co-evolutionary algorithm for large-scale optimization // Congress on Evolutionary Computation. IEEE, 2016: 3557–3565.
- [8] 徐小峰, 常玮洪, 孙燕芳. 基于关联维数和 Kolmogorov 熵的协同物流网络有序度控制模型. 系统工程学报, 2018, 33(2): 157–166.
Xu X F, Chang W H, Sun Y F. Order degree control model for collaborative logistics network based on correlation dimension and Kolmogorov entropy. Journal of Systems Engineering, 2018, 33(2): 157–166. (in Chinese)
- [9] Potter M A, De Jong K A. A cooperative coevolutionary approach to function optimization // International Conference on Parallel Problem Solving from Nature. Berlin, Heidelberg: Springer, 1994: 249–257.
- [10] Yang Z, Tang K, Yao X. Large scale evolutionary optimization using cooperative coevolution. Information Sciences, 2008, 178(15): 2985–2999.
- [11] Gardeux V, Omran M G H, Chelouah R, et al. Adaptive pattern search for large-scale optimization. Applied Intelligence, 2017, 47(2): 319–330.
- [12] Omidvar M N, Li X, Mei Y, et al. Cooperative co-evolution with differential grouping for large scale optimization. IEEE Transactions on Evolutionary Computation, 2014, 18(3): 378–393.
- [13] Mahdavi S, Rahnamayan S, Shiri M E. Multilevel framework for large-scale global optimization. Soft Computing, 2017, 21(14): 4111–4140.
- [14] 郭均鹏, 高成菊, 赵旻昊. 一种基于符号数据的群体推荐算法. 系统工程学报, 2015, 30(1): 127–134.
Guo J P, Gao C J, Zhao M H. Symbolic data analysis-based group recommendation algorithm. Journal of Systems Engineering, 2015, 30(1): 127–134. (in Chinese)
- [15] Li X, Yao X. Cooperatively coevolving particle swarms for large scale optimization. IEEE Transactions on Evolutionary Computation, 2011, 16(2): 210–224.
- [16] Li X, Tang K, Omidvar M N, et al. Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. Gene, 2013, 7(33): 8–32.
- [17] Omidvar M N, Kazimipour B, Li X, et al. CBCC3: A contribution-based cooperative co-evolutionary algorithm with improved exploration/exploitation balance // IEEE Congress on Evolutionary Computation. IEEE, 2016: 3541–3548.

作者简介:

陈昊(1982—), 男, 山东青岛人, 博士, 副教授, 研究方向: 智能计算理论与应用, 高维多目标进化计算的研究, Email: chenhaoshl@163.com;

陈园(1993—), 女, 安徽安庆人, 硕士, 研究方向: 智能计算理论与应用的研究, Email: 849598587@qq.com;

黎明(1965—), 男, 江西樟树人, 博士生导师, 教授, 研究方向: 人工智能, 模式识别等研究, Email: limingnchu@.edu.cn;

李军华(1974—), 男, 江西莲花人, 博士, 教授, 研究方向: 进化计算, 智能控制的研究, Email: jhlee126@126.com;

张聪炫(1984—), 男, 河南焦作人, 博士, 副教授, 研究方向: 图像检测与智能识别, Email: zcxds@163.com.