A Dynamic Global Differential Grouping for Large-Scale Black-Box Optimization

摘要：基于合作协同演化框架的方法解决大规模黑盒优化（Large-Scale Black-Box Optimization，LSBO）问题是当前研究求解LSBO问题的一个重要手段。合作协同演框架求解LSBO分三个阶段，即分组、子问题优化和子问题合并，分组的恰当与否直接影响着算法最终的优化性能，分组阶段希望达到组内决策变量耦合关系强，组间决策变量耦合关系弱的理想分组效果。全局微分分组是一种基于多元函数偏导数思想的分组方法，并通过维系变量间的全局信息来实现问题的自动分解。然而全局微分分组的结果一经确定之后就不再更新，不会随算法的演化进程来自动调节，影响算法的优化效果。因此本文在全局微分分组的方法基础上根据算法的演化进程提出了一种动态调整策略，可以适当的更新变量的分组结果。通过对cec2010的20个测试函数的优化结果可知，本文所提的动态全局微分分组算法能够提升LSBO问题的求解性能。

关键词：大规模黑盒优化、合作协同演化、动态、微分分组

1. Introduction

大规模优化问题是指待优化问题的决策变量个数很多，通常有上千个，甚至更多[[1](#_ENREF_1)]。在大规模优化问题中还会存在目标函数没有显式解析表达式的情况[[3](#_ENREF_3)]，此时便是大规模黑箱优化(Large Scale Black-Box Optimization, LSBO)问题[[4](#_ENREF_4)]。在实际工程领域以及研究领域中存在大量的LSBO问题[[5](#_ENREF_5)]。比如在基于流体动力学的飞机机翼形状的优化设计中就需要使用2500多个变量[[6](#_ENREF_6), [7](#_ENREF_7)]。LSBO问题的优化求解面临着两个挑战：(1)问题的黑箱特性以及问题往往存在非线性、非凸、不可微等复杂特征；(2)大规模的决策变量。

基于分治(divide-and-conquer)思想的合作协同演化(Cooperative Co-evolution, CC)[[10](#_ENREF_10)]是一种通过问题的分解来求解大规模优化问题的算法框架。由于CC框架能够将大规模优化问题分解为优化算法有能力处理的多个子问题，因此其已经成为近年来求解LSBO问题的重要手段[[1](#_ENREF_1), [2](#_ENREF_2)]，而如何实现问题的有效分解是CC框架的关键问题之一[[2](#_ENREF_2), [11](#_ENREF_11)]。

CC框架下的LSBO问题优化求解研究主要表现为对问题分解策略的研究，可以分为固定分解、随机分解以及基于学习机制的分解这三个类别。固定分解策略是将问题分为若干组相同规模的子问题。比如对于*n*维的问题，*m*组*s*维的子问题(*n*=*m*\**s,* 1*≤s≤n*/2)[[20](#_ENREF_20)]。固定分解策略对于可分问题比较有效，但是子问题的规模及变量组成等都是人为确定，且不考虑变量之间的相关性，在不可分问题中难以表现出较好的优化能力。为了能够探测变量之间的相关性，以将有关联的变量尽可能放在一个子问题中，研究人员提出了一类随机分解策略，即子问题中的变量通过随机的方式来分配，且分配的变量会随着求解过程发生改变。Yang等在[[21](#_ENREF_21), [22](#_ENREF_22)]中设计了变量随机平均分配到固定数量的分组中的分解策略用以提高关联变量进入相同分组的概率，但当一个分组中关联变量的数量超过5个时，该分解方法的优化效果会变差[[23](#_ENREF_23)]。而后Yang等提出了多层次的协同进化方法(MLCC)以解决[[22](#_ENREF_22)]中最优分组数量难以确定的问题[[24](#_ENREF_24), [25](#_ENREF_25)]，但每个分组的大小还是相同的。基于学习机制的分解策略，即在优化算法执行之前或执行过程中，通过分析某些特征来获知变量之间的联系从而实现变量分组。Ray和Yao提出了依据适应度最优的前50%个体建立相关矩阵后对变量分组的方法[[34](#_ENREF_34)]，随后Singh和Ray在[[35](#_ENREF_35)]中对此变量分组方法作了改进。为了改进MLCC[[22](#_ENREF_22)]的变量分组方法，文献[[42](#_ENREF_42)]中研究了自适应获取变量分组的大小的方法。刘金鹏、唐柯等基于所采用演化算法的特征实现了问题分解[[13](#_ENREF_13), [43](#_ENREF_43)]。文献[[14](#_ENREF_14)]提出了一种基于多元函数偏导数思想的分组方法来实现问题的自动分解，该方法在CEC’2010[[45](#_ENREF_45)]的多数测试函数中得到了非常精确的分组结果，但在较大程度上依赖于人为设定阈值，且随着问题复杂程度(子问题规模不均匀、子问题的贡献差异大、变量间关联度增加)的提高，问题分解的准确程度会降低[[9](#_ENREF_9)]。Yi Mei在发表的论文[[5](#_ENREF_5)]中，提出了一种全局分组的方法，可以减少文献[[14](#_ENREF_14)]中的分解方法对阈值的依赖以及该方法会遗漏对变量关联性检测的缺陷，但同样对复杂优化问题的分解以及变量关联程度的判断上存在不足，而且变量分组结果一经确定就不再更新，不能反映演化过程对变量之间联系的影响。

本文将针对GDG方法的不足提出一种动态的GDG方法（Dynamic GDG，D-GDG），可以随着算法的演化过程对变量的分组进行动态调整，提高算法对LSBO问题的优化效果。本文的主要结构如下：LSBO问题的定义及CC框架的介绍在第二部分；第三部分介绍GDG并提出动态的GDG方法；实验结果在第四部分给出；最后在第五部分给出结论。

1. LSBO问题定义与合作协同演化框架

2.1 LSBO问题定义

大规模黑盒优化(LSBO)问题的数学描述为

(2-1)

其中目标函数：，为决策向量，由一系列决策变量构成，即：。为决策变量的个数，也是搜索空间的维度。的取值一般比较大，通常在几百到几千。

2.2 合作协同演化框架

合作型协同演化框架主要的步骤如下：

步骤1 问题划分：将原问题划分为多个子问题。划分方法就显得比较重要，不好的划分方法将会导致组间耦合关系较强，影响整个算法的优化性能，所以设计分制策略是提高算法优化性能的最关键的一步。

步骤 2 子问题的优化：对于每个子问题进行独立求解，这一步要选择性能比较显著的子优化器，可以循环轮流对子问题进行优化也可以用并行化程序独立优化。

步骤 3 子问题合并：用当前子问题的解与其他子问题最好解合成一个完整的解向量，并进行适应度评估。

合作型协同演化(CC)框架的伪代码。

|  |
| --- |
| 算法2.1 CC(*f*, lbounds, ubounds, *D*) |
| 1: groups grouping(*f*, lbounds, ubounds, *D*) ; //分组阶段 |
| 2: pop *rand*(popsize, *D*); //进入优化阶段，初始化 |
| 3: (best, bestval) *min*(*f*(pop)); //适应度评估 |
| 4: for *i*1 to Cycle do |
| 5: for *j*1 to *size*(groups) do |
| 6: indicies groups[*j*]; |
| 7: subpoppop[:,indicies]; |
| 8: subpop*optimizer*(best, subpop, FE); //子优化器优化 |
| 9: pop[:, indicies] subpop; |
| 10: (best, bestval) *min*(f(pop); |
| 11: end for |
| 12: end for |

1. The proposed dynamic GLOBAL DIFFERENTIAL GROUPING

无论采用何种分解方法，变量分解的主要目标都是将互相关联的变量放在相同的分组，没有互相关联的变量独立成组，故变量分解的核心思想就是找到变量之间的关联性。在LSBO问题中，变量之间会存在完全可分、完全不可分、部分可分、互有重叠等多种关联特征。在黑箱环境下，可以获取与问题相关的信息只有变量个数及其定义域、不同输入对应的目标函数评价结果，变量之间的关系是完全未知的。因此通过目标函数的评价结果来分析变量之间的关系是一种可行的方案。

* 1. GLOBAL DIFFERENTIAL GROUPING

假设有如下的优化问题：

(5-3)

微分分组方法在检测出与相互依赖后，会将两个变量分为一组，同时会从变量池中剔除变量；接下来再检测到相互依赖，放到另一组，这样就检测不到与之间的相互依赖关系，所以微分分组在有些优化问题中出现分组遗漏的现象；Mei在微分分组的基础上提出了一种全局微分分组(GDG)的方法，去掉了DG方法中的变量剔除操作，而是在算法中计算出所有目标函数值差值并得到一个完全矩阵，然后直接与分组阈值比较得到一个只含有0和1的关联矩阵，再进行链接操作进而来实现决策变量的自动分组，全局微分分组极大的提高了分组准确率。

* 1. Dynamic GDG (D-GDG)

GDG方法与DG方法相同的地方在于它们都是在优化算法执行前将变量分组确定，且分组结果会贯穿整个优化过程，两种方法都未考虑随着优化过程的推进来适当的调节变量的分组以适应当前的解的分布。本文将基于GDG方法的基本思想，即根据多元函数求偏导的一般思想得出反映两两变量之间联系的原始数据矩阵。通过原始数据矩阵的标准化及模糊关系矩阵的建立，由模糊聚类方法对变量实现动态聚类。并且通过设定变量分组规模的上下限对每个变量分组规模加以限制。根据算法在运行过程中的状态，自适应调节变量的分组以促进对问题的寻优。

本文。。。这里先文字描述两种类型的dynamic，一种是固定次数的动态调整，一种是完全动态的调整。要详细说明调整的时机与方法。最好也能有一个实例，给出矩阵来分析。

这里写一下类似gdg论文中对GDG的方法描述，即该文中的algorithm 3，来说明D-GDG.

首先， DGDG 算法 采用GDG思想中计算原始差值矩阵的方法得到差值矩阵lambda，对lambda矩阵中的每个元素通过排序和去重可以得到候选阈值向量，DGDG引入一个超参数，alpha，表示模糊聚类的次数，即整个算法执行过程中重新分组的次数，相邻两个被选中的阈值之间的间距，最后，对每一个阈值采用GDG的方法处理，例如，当前要处理的阈值是2

1. experimental results and discussions
   1. experimental settings

为了说明所提D-GDG方法在LSBO问题中的作用，本文在CEC’2010 LSGO测试函数上进行了仿真实验，采用PSO算法作为子优化器，对比了DGDG与GDG的方法。为了减少实验的随机性，每种分组方法在每个函数上均独立运行25次。每次实验以最大函数评估次数（maxFEs）做为终止条件；maxFEs=。

CEC’2010 LSGO测试函数包含20个1000维的复杂函数，这些函数分为3类，F1~F3为完全可分函数，F4~F18为部分可分函数，F19~F20为完全不可分函数。

* 1. results and discussions

表？给出了仿真实验结果，表中第一行表示算法在每个函数上独立运行25次的优化结果的最小值，第二行表示中间值，第三、四行分别表示平均值与方差。表中CC-DGDG-PSO表示采用了完全动态的分组方法，CC-GDG-PSO表示采用了GDG的分组方法，CC-DGDG-PSO3、CC-DGDG-PSO6与CC-DGDG-PSO9分别表示在演化过程中重新分组了3、6与9次。

由表1可以看出，。。。这里给一下分析结果，谁优谁劣，大概原因是什么。

表1 动态划分3次、6次、9次以及完全动态划分的实验结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Functions | | CC-DGDG-PSO | CC-GDG-PSO | CC-DGDG-PSO3 | CC-DGDG-PSO6 | CC-DGDG-PSO9 |
| f1 | Min | 5.30E-02 | 4.27E-01 | 8.00E-02 | 5.30E-02 | 1.30E-01 |
| Median | 1.82E-01 | 1.36E+03 | 1.74E+00 | 1.82E-01 | 4.65E-01 |
| Mean | 2.43E-01 | 1.06E+05 | 4.93E+00 | 2.43E-01 | 5.51E-01 |
| Std | 2.18E-01 | 4.72E+05 | 8.05E+00 | 2.18E-01 | 3.66E-01 |
| f2 | Min | 7.27E+03 | 7.36E+03 | 7.44E+03 | 7.27E+03 | 7.12E+03 |
| Median | 7.66E+03 | 7.85E+03 | 7.74E+03 | 7.66E+03 | 7.54E+03 |
| Mean | 7.62E+03 | 7.86E+03 | 7.74E+03 | 7.62E+03 | 7.56E+03 |
| Std | 1.85E+02 | 2.30E+02 | 1.52E+02 | 1.85E+02 | 2.15E+02 |
| f3 | Min | 4.35E+00 | 1.20E+01 | 6.01E+00 | 4.35E+00 | 3.63E+00 |
| Median | 5.10E+00 | 1.30E+01 | 6.60E+00 | 5.10E+00 | 4.14E+00 |
| Mean | 5.11E+00 | 1.30E+01 | 6.75E+00 | 5.11E+00 | 4.09E+00 |
| Std | 4.17E-01 | 8.00E-01 | 4.50E-01 | 4.17E-01 | 2.59E-01 |
| f4 | Min | 2.84E+11 | 4.69E+11 | 1.72E+11 | 2.84E+11 | 2.53E+11 |
| Median | 5.19E+11 | 1.06E+12 | 3.44E+11 | 5.19E+11 | 6.23E+11 |
| Mean | 5.61E+11 | 1.14E+12 | 3.78E+11 | 5.61E+11 | 6.54E+11 |
| Std | 1.94E+11 | 4.56E+11 | 1.23E+11 | 1.94E+11 | 2.02E+11 |
| f5 | Min | 3.62E+08 | 2.71E+08 | 2.85E+08 | 3.62E+08 | 3.13E+08 |
| Median | 4.56E+08 | 3.47E+08 | 5.32E+08 | 4.56E+08 | 4.80E+08 |
| Mean | 4.71E+08 | 3.46E+08 | 5.09E+08 | 4.71E+08 | 4.91E+08 |
| Std | 7.13E+07 | 3.57E+07 | 1.12E+08 | 7.13E+07 | 1.13E+08 |
| f6 | Min | 2.73E+06 | 2.43E+06 | 3.54E+06 | 2.73E+06 | 2.20E+06 |
| Median | 4.05E+06 | 3.47E+06 | 4.98E+06 | 4.05E+06 | 4.35E+06 |
| Mean | 5.95E+06 | 3.58E+06 | 8.78E+06 | 5.95E+06 | 8.04E+06 |
| Std | 5.17E+06 | 1.00E+06 | 6.66E+06 | 5.17E+06 | 6.42E+06 |
| f7 | Min | 9.01E-04 | 5.73E+03 | 4.40E-01 | 9.01E-04 | 3.80E-03 |
| Median | 2.92E-02 | 5.95E+05 | 3.26E+00 | 2.92E-02 | 1.02E-01 |
| Mean | 8.48E-02 | 1.21E+05 | 5.60E+00 | 8.48E-02 | 2.11E-01 |
| Std | 1.96E-01 | 1.61E+05 | 6.79E+00 | 1.96E-01 | 2.58E-01 |
| f8 | Min | 5.26E-04 | 1.63E+02 | 1.04E+02 | 5.26E-04 | 3.01E+02 |
| Median | 1.10E-02 | 3.43E+07 | 1.15E+07 | 1.10E-02 | 2.08E+06 |
| Mean | 9.50E+03 | 4.63E+07 | 1.53E+07 | 9.50E+03 | 1.77E+07 |
| Std | 4.70E+04 | 4.02E+07 | 2.04E+07 | 4.70E+04 | 3.36E+07 |
| f9 | Min | 4.70E+06 | 4.97E+06 | 5.61E+06 | 4.70E+06 | 4.33E+06 |
| Median | 6.11E+06 | 7.84E+06 | 7.52E+06 | 6.11E+06 | 5.78E+06 |
| Mean | 6.23E+06 | 7.86E+06 | 7.37E+06 | 6.23E+06 | 5.89E+06 |
| Std | 7.79E+05 | 1.31E+06 | 9.96E+05 | 7.79E+05 | 8.96E+05 |
| f10 | Min | 8.31E+03 | 7.44E+03 | 8.95E+03 | 8.31E+03 | 8.50E+03 |
| Median | 9.40E+03 | 7.81E+03 | 9.73E+03 | 9.40E+03 | 9.43E+03 |
| Mean | 9.50E+03 | 7.81E+03 | 9.82E+03 | 9.50E+03 | 9.44E+03 |
| Std | 4.93E+02 | 1.94E+02 | 5.05E+02 | 4.93E+02 | 4.50E+02 |
| f11 | Min | 1.02E+02 | 7.70E+01 | 1.11E+02 | 1.02E+02 | 1.07E+02 |
| Median | 1.20E+02 | 8.60E+01 | 1.32E+02 | 1.20E+02 | 1.24E+02 |
| Mean | 1.21E+02 | 8.60E+01 | 1.34E+02 | 1.21E+02 | 1.25E+02 |
| Std | 8.61E+00 | 4.62E+00 | 1.70E+01 | 8.61E+00 | 8.45E+00 |
| f12 | Min | 9.80E-01 | 4.60E+01 | 1.00E+01 | 9.80E-01 | 4.43E-01 |
| Median | 1.83E+00 | 4.57E+02 | 1.80E+01 | 1.83E+00 | 9.36E-01 |
| Mean | 2.00E+00 | 9.22E+02 | 2.10E+01 | 2.00E+00 | 9.15E-01 |
| Std | 7.70E-01 | 1.47E+03 | 1.20E+01 | 7.70E-01 | 2.41E-01 |
| f13 | Min | 1.15E+03 | 5.04E+02 | 3.96E+02 | 1.15E+03 | 1.50E+03 |
| Median | 1.91E+03 | 9.61E+02 | 1.06E+03 | 1.91E+03 | 2.86E+03 |
| Mean | 2.05E+03 | 1.01E+03 | 1.22E+03 | 2.05E+03 | 3.03E+03 |
| Std | 8.25E+02 | 3.99E+02 | 5.25E+02 | 8.25E+02 | 1.38E+03 |
| f14 | Min | 1.59E+07 | 1.55E+07 | 1.61E+07 | 1.59E+07 | 1.30E+07 |
| Median | 1.82E+07 | 1.81E+07 | 2.08E+07 | 1.82E+07 | 1.70E+07 |
| Mean | 1.89E+07 | 1.83E+07 | 2.12E+07 | 1.89E+07 | 1.66E+07 |
| Std | 2.26E+06 | 1.78E+06 | 2.48E+06 | 2.26E+06 | 1.67E+06 |
| f15 | Min | 9.92E+03 | 7.30E+03 | 1.03E+04 | 9.92E+03 | 9.96E+03 |
| Median | 1.08E+04 | 7.80E+03 | 1.09E+04 | 1.08E+04 | 1.11E+04 |
| Mean | 1.09E+04 | 7.76E+03 | 1.10E+04 | 1.09E+04 | 1.10E+04 |
| Std | 6.00E+02 | 2.67E+02 | 4.92E+02 | 6.00E+02 | 5.10E+02 |
| f16 | Min | 2.17E+02 | 1.34E+02 | 2.44E+02 | 2.17E+02 | 2.18E+02 |
| Median | 2.42E+02 | 1.44E+02 | 2.57E+02 | 2.42E+02 | 2.31E+02 |
| Mean | 2.42E+02 | 1.45E+02 | 2.60E+02 | 2.42E+02 | 2.33E+02 |
| Std | 1.00E+01 | 5.78E+00 | 1.10E+01 | 1.00E+01 | 1.40E+01 |
| f17 | Min | 2.10E+02 | 1.18E+03 | 3.18E+02 | 2.10E+02 | 2.85E+02 |
| Median | 3.23E+02 | 2.51E+03 | 4.36E+02 | 3.23E+02 | 4.19E+02 |
| Mean | 3.27E+02 | 2.73E+03 | 4.52E+02 | 3.27E+02 | 4.22E+02 |
| Std | 7.30E+01 | 1.26E+03 | 1.04E+02 | 7.30E+01 | 7.70E+01 |
| f18 | Min | 4.70E+03 | 1.32E+03 | 1.71E+03 | 4.70E+03 | 3.00E+03 |
| Median | 1.05E+04 | 2.23E+03 | 3.76E+03 | 1.05E+04 | 7.42E+03 |
| Mean | 1.12E+04 | 2.47E+03 | 4.84E+03 | 1.12E+04 | 9.42E+03 |
| Std | 4.83E+03 | 9.17E+02 | 2.51E+03 | 4.83E+03 | 6.32E+03 |
| f19 | Min | 7.63E+04 | 3.28E+05 | 8.31E+04 | 7.63E+04 | 8.31E+04 |
| Median | 1.14E+05 | 4.95E+05 | 1.14E+05 | 1.14E+05 | 1.08E+05 |
| Mean | 1.13E+05 | 4.85E+05 | 1.20E+05 | 1.13E+05 | 1.11E+05 |
| Std | 2.15E+04 | 8.61E+04 | 2.18E+04 | 2.15E+04 | 2.13E+04 |
| f20 | Min | 2.13E+03 | 5.09E+07 | 2.02E+03 | 2.13E+03 | 1.89E+03 |
| Median | 2.30E+03 | 2.63E+08 | 2.43E+03 | 2.30E+03 | 2.22E+03 |
| Mean | 2.35E+03 | 3.19E+08 | 2.43E+03 | 2.35E+03 | 2.22E+03 |
| Std | 1.68E+02 | 2.01E+08 | 1.83E+02 | 1.68E+02 | 2.56E+02 |

1. Conclusions

本文针对GDG方法在LSBO问题的变量分组中不能根据算法的演化过程进行动态调节的问题，提出了一种变量分解的动态的GDG方法，研究了固定次数动态调整与完全动态调整两种方式。通过对cec2010的20个测试函数的求解可知，通过动态调整变量的分组结果可以提升算法的优化效果。接下来我们将继续深入研究动态调整变量分组结果的策略以进一步提升求解效果。

References

[1] S. Mahdavi, M. E. Shiri, S. Rahnamayan. Metaheuristics in large-scale global continues optimization: A survey[J]. Information Sciences. 2015, 295: 407-428

[2] Z.-H. Zhou, N. V. Chawla, Y. Jin, G. J. Williams. Big Data Opportunities and Challenges: Discussions from Data Analytics Perspectives [Discussion Forum][J]. IEEE Computational Intelligence Magazine. 2014, 9(4): 62-74

[3] K. Scheerlinck, H. Vernieuwe, B. D. Baets. Zadeh's Extension Principle for Continuous Functions of Non-Interactive Variables: A Parallel Optimization Approach[J]. IEEE Transactions on Fuzzy Systems. 2012, 20(1): 96-108

[4] T. Schaul. Studies in Continuous Black-box Optimization [D]: Universität München, 2011

[5] Y. Mei, M. N. Omidvar, X. Li, X. Yao. A Competitive Divide-and-Conquer Algorithm for Unconstrained Large-Scale Black-Box Optimization[J]. ACM Transactions on Mathematical Software. To Appear

[6] A. Mucherino, M. Fuchs, X. Vasseur, S. Gratton. Variable neighborhood search for robust optimization and applications to aerodynamics. In: Large-Scale Scientific Computing: Springer; 2011:230-237

[7] G. N. Vanderplaats. Very large scale optimization[M]: National Aeronautics and Space Administration, Langley Research Center, 2002

[8] X.-S. Yang. Nature-inspired metaheuristic algorithms[M]: Luniver press, 2010

[9] M. N. Omidvar, X. Li, K. Tang. Designing benchmark problems for large-scale continuous optimization[J]. Information Sciences. 2015, 316: 419-436

[10] M. A. Potter, K. A. De Jong. A cooperative coevolutionary approach to function optimization. In: Parallel Problem Solving from Nature—PPSN III: Springer; 1994:249-257

[11] M. A. Potter, K. A. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents[J]. Evolutionary computation. 2000, 8(1): 1-29

[12] D. M. Cabrera. Evolutionary algorithms for large-scale global optimisation: a snapshot, trends and challenges[J]. Progress in Artificial Intelligence. 2016: 1-5

[13] J. Liu, K. Tang. Scaling Up Covariance Matrix Adaptation Evolution Strategy Using Cooperative Coevolution. In: Yin H, Tang K, Gao Y, et al., eds. Intelligent Data Engineering and Automated Learning – IDEAL 2013: Springer Berlin Heidelberg; 2013:350-357

[14] M. N. Omidvar, X. Li, Y. Mei, X. Yao. Cooperative co-evolution with differential grouping for large scale optimization[J]. IEEE Transactions on Evolutionary Computation. 2014, 18(3): 378-392

[15] A. LaTorre, S. Muelas, J.-M. Peña. A comprehensive comparison of large scale global optimizers[J]. Information Sciences. 2015, 316: 517-549

[16] X. Li, K. Tang, P. N. Suganthan, Z. Yang. Editorial for the special issue of Information Sciences Journal (ISJ) on Nature-inspired algorithms for large scale global optimization[J]. Information Sciences. 2015, 316(C): 437-439

[17] M. Lozano, D. Molina, F. Herrera. Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems[J]. Soft Computing. 2011, 15(11): 2085-2087

[18] Y. Liu, X. Yao, Q. Zhao, H. T. Scaling up fast evolutionary programming with cooperative coevolution[C]. IEEE Congress on Evolutionary Computation, 2001,1101-1108

[19] Y. J. Shi, H. F. Teng, Z. Q. Li. Cooperative Co-evolutionary Differential Evolution for Function Optimization. In: Wang L, Chen K, Ong Y, eds. Advances in Natural Computation: Springer Berlin Heidelberg; 2005:1080-1088

[20] F. van den Bergh, A. P. Engelbrecht. A cooperative approach to particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation. 2004, 8(3): 225-239

[21] Z. Yang, K. Tang, X. Yao. Differential evolution for high-dimensional function optimization[C]. IEEE Congress on Evolutionary Computation: IEEE, 2007,3523-3530

[22] Z. Yang, K. Tang, X. Yao. Large scale evolutionary optimization using cooperative coevolution[J]. Information Sciences. 2008, 178(15): 2985-2999

[23] M. N. Omidvar, X. Li, Z. Yang, X. Yao. Cooperative co-evolution for large scale optimization through more frequent random grouping[C]. IEEE Congress on Evolutionary Computation: IEEE, 2010,1-8

[24] Z. Yang, K. Tang, X. Yao. Multilevel cooperative coevolution for large scale optimization[C]. IEEE Congress on Evolutionary Computation: IEEE, 2008,1663-1670

[25] 杨振宇. 基于自然计算的实值优化算法与应用研究 [D]: 中国科学技术大学, 2010

[26] X. Li, X. Yao. Tackling high dimensional nonseparable optimization problems by cooperatively coevolving particle swarms[C]. IEEE Congress on Evolutionary Computation: IEEE, 2009,1546-1553

[27] X. Li, X. Yao. Cooperatively Coevolving Particle Swarms for Large Scale Optimization[J]. IEEE Transactions on Evolutionary Computation. 2012, 16(2): 210-224

[28] 胡善庆. 基于合作协同的演化算法研究 [D]: 中国科学技术大学, 2014

[29] 王旭, 赵曙光. 解决高维优化问题的差分进化算法[J]. 计算机应用. 2014, 34(1): 179-181

[30] C. Crăciun, M. Nicoară, D. Zaharie. Enhancing the Scalability of Metaheuristics by Cooperative Coevolution. In: Lirkov I, Margenov S, Waśniewski J, eds. Large-Scale Scientific Computing: Springer Berlin Heidelberg; 2010:310-317

[31] Y. Ren, Y. Wu. An efficient algorithm for high-dimensional function optimization[J]. Soft Computing. 2013, 17(6): 995-1004

[32] E. Glorieux, B. Svensson, F. Danielsson, B. Lennartson. Improved Constructive Cooperative Coevolutionary Differential Evolution for Large-Scale Optimisation[C]. 2015 IEEE Symposium Series on Computational Intelligence: IEEE, 2015,1703-1710

[33] 邱飞岳, 莫雷平, 江波, 王丽萍. 基于大规模变量分解的多目标粒子群优化算法研究[J]. 计算机学报. 2015, 38(108): 1-15

[34] R. T., X. Yao. A cooperative coevolutionary algorithm with Correlation based Adaptive Variable Partitioning[C]. IEEE Congress on Evolutionary Computation, 2009,983-989

[35] H. K. Singh, T. Ray. Divide and Conquer in Coevolution: A Difficult Balancing Act. In: Sarker R, Ray T, eds. Agent-Based Evolutionary Search: Springer Berlin Heidelberg; 2010:117-138

[36] W. Chen, T. Weise, Z. Yang, K. Tang. Large-scale global optimization using cooperative coevolution with variable interaction learning. In: Parallel Problem Solving from Nature, PPSN XI: Springer; 2010:300-309

[37] Y. Wang, B. Li. Two-stage based ensemble optimization for large-scale global optimization[C]. IEEE Congress on Evolutionary Computation: IEEE, 2010,1-8

[38] K. Zhang, B. Li. Cooperative Coevolution with global search for large scale global optimization[C]. IEEE Congress on Evolutionary Computation, 2012,1-7

[39] L. Sun, S. Yoshida, X. Cheng, Y. Liang. A cooperative particle swarm optimizer with statistical variable interdependence learning[J]. Information Sciences. 2012, 186(1): 20-39

[40] 王帆. 面向高维及多目标的协同细菌觅食算法研究 [D]: 大连理工大学, 2013

[41] M. N. Omidvar, X. Li, X. Yao. Cooperative co-evolution with delta grouping for large scale non-separable function optimization[C]. IEEE Congress on Evolutionary Computation: IEEE, 2010,1-8

[42] M. N. Omidvar, Y. Mei, X. Li. Effective decomposition of large-scale separable continuous functions for cooperative co-evolutionary algorithms[C]. IEEE Congress on Evolutionary Computation, 2014,1305-1312

[43] 刘金鹏. 面向大规模实值优化问题的CMA-ES算法及其分制策略研究 [D]: 中国科学技术大学, 2014

[44] Z. Cao, L. Wang, Y. Shi, X. Hei, X. Rong, Q. Jiang, H. Li. An effective cooperative coevolution framework integrating global and local search for large scale optimization problems[C]. IEEE Congress on Evolutionary Computation: IEEE, 2015,1986-1993

[45] K. Tang, X. Li, P. N. Suganthan, Z. Yang, T. Weise. Benchmark Functions for the CEC’2010 Special Session and Competition on Large-Scale Global Optimization; 2010