

文章编号:1671-6833(2023)05-0032-08

# 惯性分组和重叠特征选择辅助的昂贵大规模优化算法

邓传义<sup>1</sup>, 孙超利<sup>2</sup>, 刘晓彤<sup>2</sup>, 张晓红<sup>3</sup>, 李春鹏<sup>4</sup>

(1. 太原科技大学 应用科学学院, 山西 太原 030024; 2. 太原科技大学 计算机科学与技术学院, 山西 太原 030024;  
3. 太原科技大学 经济与管理学院 山西 太原 030024; 4. 山西吉成科技股份有限公司 山西 太原 030000)

**摘要:** 昂贵大规模优化问题存在着决策变量之间高度耦合、求解容易陷入局部最优以及目标评价昂贵等问题, 导致在资源有限的情况下很难获得全局最优解。为此, 基于合作型协同演化策略提出了一种惯性分组和重叠特征选择的方法来辅助求解昂贵大规模优化问题。首先, 采用重叠特征选择技术将一个大规模优化问题分解为若干个低维的重叠子问题, 并对每一个子问题进行独立的代理模型辅助的优化搜索。其次, 将每个子问题搜索获得的潜力个体合成一个完整的解, 对其使用昂贵目标函数进行评价。再次, 算法还采用惯性分组技术控制优化过程中重新分组的频率以延长分组方案的开发周期, 从而提升优化效果。最后, 为了测试所提算法的性能, 将其与求解昂贵大规模问题的 3 种优化算法在 CEC2013 的 15 个基准函数上获得的实验结果进行了对比。结果表明: 所提算法在求解昂贵大规模优化问题上具有一定的竞争力, 尤其适用于求解部分可分离、重叠或完全不可分离等问题。

**关键词:** 大规模优化; 昂贵问题; 重叠特征选择; 惯性分组; 代理模型; 合作型协同演化

**中图分类号:** TP18; TP301.6

**文献标志码:** A

**doi:** 10.13705/j.issn.1671-6833.2023.05.013

大规模优化问题广泛存在于生活中, 例如空气动力学形状优化<sup>[1]</sup>、大规模空中交通流量优化<sup>[2]</sup>、以及用于大规模图像识别的深层卷积网络优化<sup>[3]</sup>等。一般来说, 当决策变量维数超过 100 时, 该问题就被称为大规模优化问题<sup>[4]</sup> (large scale optimization problems, LSOP)。大规模优化问题的数学表达式如式(1)所示:

$$\begin{cases} \min f(\mathbf{x}); \\ \text{s. t. } \mathbf{x} \in \Omega. \end{cases} \quad (1)$$

式中:  $\Omega = \{\mathbf{x} \in \mathbf{R} \mid \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$  表示可行域,  $\mathbf{l}$  和  $\mathbf{u}$  分别表示可行域的下界和上界;  $\mathbf{x}$  表示  $n$  维的决策向量;  $f(\mathbf{x})$  表示目标函数。随着决策变量维数的增加, 搜索空间也会变得越来越复杂, 导致很难获得问题的最优解, 这也被称为“维度灾难”问题<sup>[5]</sup>。

目前用于求解大规模优化问题的方法大致可分为 2 类: 基于分解的方法<sup>[6]</sup> 和基于非分解的方法<sup>[7]</sup>。基于分解的方法是将一个高维的原始问题分解为若干个低维的子问题, 然后通过对于子问题的

独立优化与相互合作来寻找原始问题的最优解。合作型协同演化 (cooperative coevolutionary, CC)<sup>[8]</sup> 是基于分解方法的常用框架。基于非分解的方法则始终将高维的原始问题视为一个整体, 然后采用元启发式算法的变种算法加强算法在高维空间的搜索能力。例如竞争群体优化 (competitive swarm optimizer, CSO) 方法<sup>[9]</sup>、社会学习粒子群优化 (social learning particle swarm optimization, SL-PSO) 方法<sup>[10]</sup> 以及基于分层学习的群体优化 (level-based learning swarm optimizer, LLSO) 方法<sup>[11]</sup> 等都是基于非分解的方法。

当大规模优化问题的目标函数评价耗时或者评价成本昂贵时, 称之为昂贵大规模优化问题。由于进化算法在获得最优解之前都需要消耗大量评价次数, 因此, 它们并不适用于求解昂贵优化问题。为了节省目标函数评价, 通常引入代理模型以代替昂贵的真实目标函数评价。De Falco 等<sup>[12]</sup> 将代理模型嵌入到 CC 的框架中, 提出了一种代理模型辅助的协同进化算法 (surrogate-assisted CC, SACC), 用于求

收稿日期: 2023-02-16; 修订日期: 2023-04-24

基金项目: 国家自然科学基金资助项目 (61876123); 山西省重点研发计划项目 (202102020101002); 山西省自然科学基金资助项目 (202203021211194)

通信作者: 孙超利 (1978—), 女, 浙江诸暨人, 太原科技大学教授, 博士, 博士生导师, 主要从事智能计算、机器学习、代理模型辅助的进化优化及其在实际工程中的应用等研究, E-mail: chaoli\_sun@tyust.edu.cn。

引用本文: 邓传义, 孙超利, 刘晓彤, 等. 惯性分组和重叠特征选择辅助的昂贵大规模优化算法[J]. 郑州大学学报(工学版), 2023, 44(5): 32-39. (DENG C Y, SUN C L, LIU X T, et al. An inertial grouping and overlapping feature selection assisted algorithm for expensive large-scale optimization problems [J]. Journal of Zhengzhou University (Engineering Science), 2023, 44(5): 32-39.)

解大规模优化问题。Fu 等<sup>[13]</sup>提出了一种随机特征选择辅助的代理模型辅助进化算法 (surrogate-assisted evolutionary algorithm with random feature selection, SAEA-RFS), 该算法采用有放回的随机特征选择技术来构建若干个不同的子问题, 然后子问题在径向基函数神经网络的辅助下进行进化优化求解。Sun 等<sup>[14]</sup>提出了代理模型集成辅助的大规模昂贵进化算法 (surrogate ensemble assisted large-scale expensive evolutionary algorithm with a random grouping technique, SEA-LEEA-RG), 该算法采用不重复随机分组技术将大规模问题分解成若干大小不一的子问题, 然后在子问题上建立全局模型和局部模型的集成模型辅助问题优化。

然而, 目前求解昂贵大规模优化问题的算法仍存在以下不足: 一是频繁地执行分组操作会使得某些理想的分组方案在没有得到充分优化的情况下就会被下一个随机分组方案覆盖; 二是无放回的随机分组增加了相关变量划分到不同子问题的风险。为此, 本文在 CC 框架的基础上提出惯性分组和重叠特征选择辅助的优化算法 (inertial grouping and overlapping feature selection assisted optimization algorithm, IG-OFSA)。其中, 惯性分组技术通过设置参数来控制优化过程中重新分组的频率, 从而使得每一个分组方案都能得到较充分的优化; 重叠特征选择技术通过在子问题之间增加重叠变量, 使相同变量在不同子问题上进行优化, 一方面能够增加相关变量被划分到同一子问题中的概率, 另一方面重叠变量通过参与不同子问题的优化, 在一定程度上可防止其偏离正确的搜索方向。

1 相关工作

1.1 合作型协同演化策略

Potter 等<sup>[8]</sup>最早提出合作型协同演化 (cooperative coevolutionary, CC) 策略, 他们将 1 个  $n$  维的问题分解成了  $n$  个 1 维的问题进行求解。根据分解方式的不同, 基于 CC 框架下求解大规模优化问题的方法大致可以分为 2 类: 基于静态分组的方法和基于动态分组的方法。

基于静态分组的方法在优化开始前将大规模问题的决策变量分解成若干个低维的子组, 而这些子组在接下来的优化过程中保持不变。Van Den Bergh 等<sup>[15]</sup>提出了协同粒子群优化 (cooperative particle swarm optimizer, CPSO) 方法, 将决策变量随机分解成  $K$  个互不相交的子组后便不再执行分解操作。但当变量之间存在交互关系时, 随机分解的分

组精度会大大降低。Chen 等<sup>[16]</sup>提出了具有变量交互学习的协同演化 (CC with variable interaction learning, CCVIL) 方法, CCVIL 在执行分解前, 需要先判断决策变量之间的交互关系。差分分组 (differential grouping, DG)<sup>[17]</sup>是一种 CCVIL 的分组策略, DG 将具有直接交互关系的变量划分到同一子组。但是 DG 并没有检测变量之间的间接交互, 所以为了进一步提高分组精度, Omidvar 等<sup>[18]</sup>提出了一种改进的差分分组 (improved variant of the differential grouping, DG2) 算法, DG2 将所有具有直接交互和间接交互的变量都分配到同一子组。尽管 CCVIL 能够实现较为理想的分组, 但检测变量之间的交互需要消耗大量的目标函数评价, 所以, CCVIL 并不适用于直接求解昂贵大规模优化问题。

基于动态分组的方法则是在整个优化过程中动态地执行分解操作, Yang 等<sup>[19]</sup>提出的随机分组 (random grouping, RG) 策略和多级协作协同进化 (multi-level cooperative co-evolution, MLCC) 策略都是基于动态分组的方法, 在这些策略中, 子组的大小和成员在每一个协同周期都会发生改变。

1.2 差分进化算法

Storn 等<sup>[20]</sup>于 1997 年提出了差分进化 (differential evolution, DE) 算法, 该算法是一种基于群体差异的启发式搜索算法。DE 通过依次执行种群初始化、变异、交叉和选择这 4 种操作, 从而实现个体的进化和种群的更新。目前研究人员提出了许多 DE 变种, 它们的主要区别在于采用了不同的变异策略。本文采用“DE/rand-to-best/1”作为变异策略, 该变异操作如式 (2) 所示:

$$V_i^t = X_{r_1}^t + F \cdot (X_{best}^t - X_{r_1}^t) + F \cdot (X_{r_2}^t - X_{r_3}^t)。$$
 (2)

式中:  $V_i^t$  为在第  $t$  代时, 第  $i$  个目标个体  $X_i^t$  对应的变异向量;  $X_{r_1}^t$ 、 $X_{r_2}^t$ 、 $X_{r_3}^t$  为从第  $t$  代种群中随机选择的互不相同的个体, 且这些个体都与当前的目标个体  $X_i^t$  不同;  $F$  为一个大于 0 的缩放因子;  $X_{best}^t$  为当前种群中适应度最小的个体。

1.3 增量克里金模型

克里金 (Kriging) 模型<sup>[21]</sup>在得到模型预测值的同时, 还能得到该预测值的不确定度, 所以克里金模型被广泛用作代理模型以辅助优化算法。在训练克里金模型的过程中, 当获得新的样本点时, 为了更新模型, 传统的做法是从头开始训练一个新的克里金模型, 但当样本量的规模很大或者问题的维度很高时, 再次训练新模型花费的时间是巨大的, 为了缓解计算压力, Zhan 等<sup>[22]</sup>提出增量克里金辅助进化算法 (incremental Kriging assisted evolution algorithm,

IKAEA), IKAEA 将增量学习技术引入到克里金模型的更新过程中,使得更新克里金模型的时间复杂度由  $O(n^3)$  降为  $O(n^2)$ ,并且 IKAEA 的预测结果与标准的克里金模型的预测结果具有一致性。

## 2 惯性分组和重叠特征选择辅助的昂贵大规模优化算法

### 2.1 算法描述

CC 框架下求解大规模优化问题算法的性能取决于分组方案的合理性,所以分解操作是合作型协同演化策略的关键性操作之一。在求解决策变量不可分解的昂贵大规模优化问题时,有限的目标函数评价次数和变量之间的耦合关系对算法的搜索效率提出了更高的要求。具有变量交互学习的协同演化方法在问题的分解过程中需要消耗大量的目标函数评价,使其不适用于求解昂贵大规模优化问题。同时,变量的耦合关系会使得基于随机分组策略算法的性能大大下降。因此,为了提高求解昂贵大规模优化问题的性能和效率,本文提出了一种惯性分组和重叠特征选择辅助的优化算法。算法 1 给出了本文方法的步骤。算法 1 由 3 个部分组成:第一部分为初始化阶段(Step 1~Step 3),主要是使用拉丁超立方体采样采集初始样本并从初始样本中选取训练集和初始种群;第二部分为分解训练阶段(Step 4~Step 6),主要采用重叠特征选择技术将大规模优化问题分解成低维的重叠子问题,并在子问题上训练代理模型,同时采用惯性系数判断是否重新分组;第三部分为解的重构阶段(Step 7~Step 8),该阶段在所有子问题优化结束之后,将每一个子问题的期望提高(expected improvement, EI)值最大的个体合并成一个完整的解并对其进行真实的目标函数评价。在算法 1 中,  $sp_k$  表示第  $k$  个子问题。

#### 算法 1 IG-OFSA 算法。

输入:问题的维度  $D$ 、奇数子组的个数  $T$ 、子问题的个数  $K$ 、初始样本的个数  $NIS$ 、训练集的初始大小  $N_t$ 、种群的大小  $N_p$ 、最大的目标函数评价次数  $max\_FES$ 、在代理模型上搜索的迭代次数  $nIter$ 、惯性系数  $I$ ;

输出:最优解  $Xbest$ 、最优解的适应值  $f(Xbest)$ 。

Step 1 在解空间内用拉丁超立方体采样策略采集  $NIS$  个初始样本,计算每个样本的适应值并存入存档  $Arc$ ;

Step 2 在  $Arc$  中选择适应值排名前  $N_t$  的样本作为训练集  $S$ ;

Step 3 在  $Arc$  中选择适应度值排名前  $N_p$  的样本作为初始种群;

Step 4 采用重叠特征选择技术将原始问题分

解成  $K$  个子问题;

Step 5 若当前的协同周期是惯性系数  $I$  的整数倍,则转至 Step 2,否则,转至 Step 6;

Step 6 依次在每个子问题上训练/更新代理模型;

Step 7 依次在子问题上进行代理模型辅助的差分进化,并记录每一个子问题  $sp_k$  ( $k=1,2,\dots,K$ ) 中 EI 值最大的个体  $x_{best}^{sp_k}$ ;

Step 8 将所有  $x_{best}^{sp_k}$  合并成上下文向量  $cv$ ,并计算  $cv$  的适应值;

Step 9 将  $cv$  存入  $Arc$  和训练集  $S$  中;

Step 10 更新种群;

Step 11 如果算法不满足终止条件,则转至 Step 5,否则,转至 Step 12;

Step 12 输出全局最优解  $Xbest$  及其适应值  $f(Xbest)$ ,算法结束。

### 2.2 惯性分组

在求解昂贵大规模优化问题时,受资源限制,现有算法通常采用随机分组技术实现对大规模问题的分解。传统的合作型协同演化策略在每一轮协同演化周期开始前都会执行一次分解操作,这会使得某些合理的随机分组方案还没得到充分的利用就会被下一个随机分组方案取代,所以本文提出采用惯性系数  $I$  来控制分组频率,以延长每一个随机分组方案的开发周期,即只有在满足条件时才对大规模问题重新分组,使得合理的分组方案能够被充分地开发和利用,从而平衡随机分组方法的探索能力和开发能力。

### 2.3 重叠特征选择

如果强行将一个不可分解问题随机分解成若干个互不相交的低维子问题进行求解,算法极易陷入局部最优,因此,本文采用了一种重叠特征选择技术来改进随机分组技术。该技术首先将高维问题的决策变量分解成  $T$  个互不相交的子组,其序号用奇数表示,再从每一个子组中随机选取至少一个决策变量构成一个与该子组对应的子组,其序号用偶数表示,即该技术将高维问题分解成了  $2T$  个子组。将这些子组按序号形成一个闭环,并从序号为 2 的子组开始,两两组成一个子问题。算法通过重叠特征选择技术最终将高维问题分解成  $T$  个重叠的子问题。重叠特征选择技术通过重叠变量的选取,一方面增加相关变量被划分到同一子问题中的概率,另一方面重叠变量通过参与不同子问题的优化,在一定程度上可防止其偏离正确的搜索方向。图 1 为重叠特征选择技术分组过程的一个简单例子。



图 1 中假设原始问题含有 9 个决策变量,首先将原始问题随机均匀地分解成 3 个互不相交的子组:组 1、组 3、组 5,然后在组 1 中随机选取至少一个决策变量构成组 2,同理分别在组 3 和组 5 中选取变量构成组 4 和组 6。接着,由组 2 与组 3 中的变量构成子问题 1,由组 4 与组 5 中的变量构成子问题 2,由组 6 与组 1 中的变量构成子问题 3。对于重叠变量,在合并成上下文向量  $\mathbf{cv}$  时,采用其在不同子问题中得到的变量值的均值在  $\mathbf{cv}$  中相应的位置进行填充。

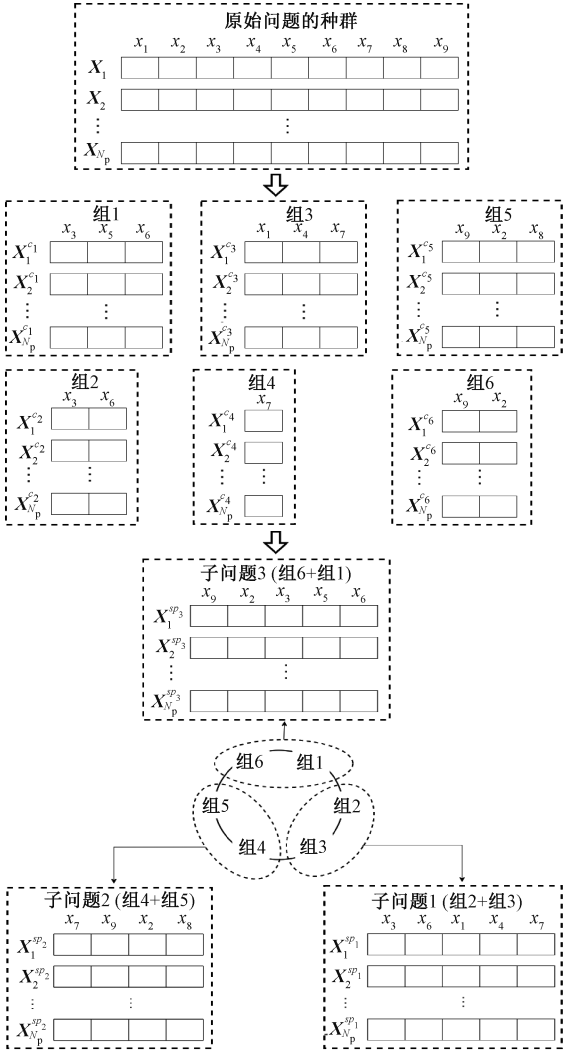


图 1 重叠特征选择分组技术

Figure 1 Overlapping feature selection grouping technique

2.4 计算复杂度分析

IG-OFSA 算法的运行成本主要来源于代理模型的训练和基于模型寻优 2 个部分。假设子问题的个数为  $K$ ,训练数据集的大小为  $N_t$ ,种群的规模为  $N_p$ 。训练增量克里金模型的时间复杂度为  $O(KN_t^2)$ ,模型寻优的时间复杂度为  $O(KN_p)$ 。一般情况下,  $N_t \gg N_p$ ,因此,IG-OFSA 的计算复杂度为  $O(KN_t^2)$ 。而训练 RBF 代理模型的计算复杂度为  $O(KN_t^2)$ ,训

练传统克里金模型的时间复杂度为  $O(KN_t^3)$ ,因此,与 RBF 模型辅助的或者传统克里金模型辅助的昂贵大规模优化算法相比,IG-OFSA 在提高优化效果的同时,并没有增加额外的计算复杂度。

3 实验与结果分析

3.1 参数设置

本文在 CEC2013 标准测试函数集上进行了实验,以验证本文算法的性能。奇数子组个数设置为 20,奇数子组的大小设置为  $sub\_size = \lfloor D/20 \rfloor$ ,偶数子组的大小取值为  $[1, sub\_size]$ ,初始化样本的个数和训练集的初始大小都设置为  $4 \cdot sub\_size$ 。本文采用增量克里金模型作为子问题的代理模型,采用差分进化算法作为优化器,缩放因子  $F$  设置为 0.8,交叉概率  $r$  设置为 0.95,种群的大小设置为 100。最大的目标函数评价次数为  $11 \cdot D$ ,这也是实验的终止条件。每一个算法在每一个基准函数上独立运行 25 次,并在具有 Bonferroni 校正的显著性水平为 0.05 的 Wilcoxon 秩和检验下检验不同算法得到的结果是否存在显著差异。

实验所用处理器为 AMD Ryzen 7 4800U CPU@1.8 GHz,内存为 16 GB。

3.2 参数敏感性分析

本节对本文所提算法中的 2 个关键参数进行了分析,一个是奇数子组的个数  $T$ ,另一个是惯性系数  $I$ 。奇数子组的个数  $T$  决定子问题的维度,  $T$  越小意味着子问题的维度就越高,在高维的子问题上建立代理模型存在困难,而  $T$  越大意味着子问题的个数也越多,那么训练所有子问题的代理模型的时间代价较大。从 CEC2013 的 5 类函数中各选出一个代表函数(分别为  $f_1, f_4, f_9, f_{13}$  及  $f_{15}$ )来进行实验分析,  $T$  分别设置为 10、20、30、40,其实验对比结果如表 1 所示。表中 *median* 和 *MAD* 分别为独立运行 25 次的结果的中位数和中值绝对偏差。由表 1 可以看出,当  $T$  设置为 20 时,算法得到的结果最好,所以本文将  $T$  设置为 20。

基于协同演化策略算法的性能还与分组方案相关,为了平衡分组方案的探索能力和开发能力,本文设置一个惯性系数  $I$  来决定分组的频率。  $I$  分别设置为 5、10、15、20,其在  $f_1, f_4, f_9, f_{13}$  及  $f_{15}$  上的实验对比结果如表 1 所示。从表 1 可以看出,当  $I$  设置为 10 时,算法获得的结果最优,因此,本文将惯性系数  $I$  设置为 10。

3.3 惯性分组技术的有效性

为了验证 IG-OFSA 算法中惯性分组技术的有

表 1 不同的  $T$  值和不同的  $I$  值的结果对比

Table 1 Comparison of results with different  $T$  values and different  $I$  values

测试函数	评价指标	$T=10$	$T=20$	$T=30$	$T=40$	$I=5$	$I=10$	$I=15$	$I=20$
$f_1$	median	<b>1.769E+09</b>	1.818E+09	2.747E+09	3.881E+09	1.853E+09	<b>1.818E+09</b>	2.012E+09	1.845E+09
	MAD	<b>3.989E+06</b>	1.002E+08	8.995E+07	8.753E+07	1.027E+07	<b>1.002E+08</b>	6.027E+07	3.451E+07
$f_4$	median	1.972E+11	<b>1.486E+11</b>	2.003E+11	2.570E+11	1.512E+11	<b>1.486E+11</b>	1.497E+11	1.523E+11
	MAD	1.963E+10	<b>6.568E+09</b>	2.061E+09	5.690E+09	1.089E+10	<b>6.568E+09</b>	2.732E+09	1.082E+10
$f_9$	median	9.132E+08	<b>7.964E+08</b>	1.040E+09	1.106E+09	8.697E+08	<b>7.964E+08</b>	8.109E+08	8.011E+08
	MAD	1.819E+07	<b>6.493E+07</b>	1.694E+07	7.223E+06	7.696E+07	<b>6.493E+07</b>	3.003E+07	1.505E+08
$f_{13}$	median	<b>1.658E+10</b>	2.254E+10	4.602E+10	4.256E+10	2.610E+10	<b>2.254E+10</b>	2.564E+10	3.194E+10
	MAD	<b>7.693E+08</b>	2.318E+09	1.479E+09	8.304E+09	3.249E+09	<b>2.318E+09</b>	5.306E+07	1.079E+10
$f_{15}$	median	4.317E+07	<b>3.999E+07</b>	5.039E+07	7.385E+07	4.292E+07	<b>3.999E+07</b>	4.199E+07	4.538E+07
	MAD	9.084E+06	<b>4.215E+06</b>	6.166E+06	7.690E+05	7.952E+04	<b>4.215E+06</b>	5.753E+04	1.459E+06

效性,将 IG-OFSA 算法的惯性分组技术移除后产生一个变种算法 OFSA。OFSA 可以视为 IG-OFSA 中惯性系数设置为 1 的特殊情况,即在每一轮协同演化周期开始时都执行一次分解操作。OFSA 的其他设置都与 IG-OFSA 的设置相同。同样,在 CEC2013 的 5 个代表函数上进行实验分析,OFSA 和 IG-OFSA 的实验结果对比如表 2 所示。从表 2 中可以看出,IG-OFSA 在 3 个代表函数上获得比 OFSA 更好的结果,说明惯性分组技术在解决大规模昂贵优化问题时能够发挥重要的作用,特别是在问题不可分的情况下。

表 2 OFSA 与 IG-OFSA 的结果对比

Table 2 Comparison of results obtained by OFSA and IG-OFSA

测试函数	评价指标	OFSA	IG-OFSA
$f_1$	median	<b>1.710E+09</b>	1.818E+09
	MAD	<b>8.374E+07</b>	1.002E+08
$f_4$	median	<b>1.373E+11</b>	1.486E+11
	MAD	<b>1.477E+09</b>	6.568E+09
$f_9$	median	8.860E+08	<b>7.964E+08</b>
	MAD	1.147E+07	<b>6.493E+07</b>
$f_{13}$	median	2.413E+10	<b>2.254E+10</b>
	MAD	5.373E+08	<b>2.318E+09</b>
$f_{15}$	median	4.243E+07	<b>3.999E+07</b>
	MAD	1.533E+06	<b>4.215E+06</b>

### 3.4 重叠特征选择技术的有效性

为了验证 IG-OFSA 中重叠特征选择技术的有效性,将 IG-OFSA 的重叠特征选择技术移除后产生一个变种算法 IGA。IG-OFSA 中的奇数序号子组构成 IGA 的全部子组,即 IGA 中的子组互不相交,且每一个子组独立构成一个子问题。IGA 的其他设置都与 IG-OFSA 的设置相同。这两个算法的实验结果对比如表 3 所示。从表 3 中可以看出,IG-OFSA 在 5 个代表函数上比 IGA 获得了更好的优化结果,表明重叠变量参与不同子问题的优化有助于找到更好的解。

表 3 IGA 与 IG-OFSA 的结果对比

Table 3 Comparison of results obtained by IGA and IG-OFSA

测试函数	评价指标	IGA	IG-OFSA
$f_1$	median	3.440E+09	<b>1.818E+09</b>
	MAD	1.557E+08	<b>1.002E+08</b>
$f_4$	median	3.376E+11	<b>1.486E+11</b>
	MAD	3.045E+10	<b>6.568E+09</b>
$f_9$	median	8.205E+08	<b>7.964E+08</b>
	MAD	4.552E+06	<b>6.493E+07</b>
$f_{13}$	median	4.274E+10	<b>2.254E+10</b>
	MAD	2.271E+09	<b>2.318E+09</b>
$f_{15}$	median	6.904E+07	<b>3.999E+07</b>
	MAD	8.422E+05	<b>4.215E+06</b>

### 3.5 2 种技术的贡献程度分析

为了分析本文所提出的 2 种技术对提升算法性能的贡献程度,本节在 CEC2013 的 5 个代表函数上,将 IGA 和 OFSA 得到的结果与近年来求解昂贵大规模优化问题的代表算法 SACC-RBFN<sup>[12]</sup>、SACC-GP<sup>[12]</sup>、SACC-QPA<sup>[12]</sup>、SACC-SVR<sup>[12]</sup>、SAEA-RFS<sup>[13]</sup>和 SEA-LEEA-RG<sup>[14]</sup>获得的结果进行了对比,实验对比结果如表 4 所示。从表 4 可以看出,IGA 和 OFSA 在 4 个代表函数上相比于对比算法均获得了更好的结果,说明单一的技术改进对提升算法的性能都是有效的。同时还可以发现,OFSA 在 4 个代表函数上能获得比 IGA 更好的结果,表明重叠特征选择技术对提升算法性能的贡献程度更高。

### 3.6 IG-OFSA 与其他算法的性能对比

为了验证本文所提算法的性能,本节将其与近年来求解昂贵大规模优化问题的代表性算法(包括 SACC-RBFN、SACC-GP、SACC-QPA、SACC-SVR、SAEA-RFS 和 SEA-LEEA-RG)在 CEC2013 测试函数上进行了实验结果对比。所有算法的停止条件为满足最大目标函数评价次数  $11 \cdot D$ 。所有算法在 CEC2013

测试函数上得到的优化结果如表 5 所示。使用 Wilcoxon 秩和检验分析所得结果是否相同,所有算法在同一测试函数上得到的最好的结果以粗体显示。

从表 5 中可以看出,IG-OFSA 在 10 个 CEC2013 测试函数上获得的结果明显优于其他对比算法。具体地,与对比算法进行两两比较,IG-OFSA 都能在多于

表 4 两种变种算法与其他算法的结果对比

Table 4 Comparison of results obtained by the two variation algorithms and other algorithms									
测试函数	评价指标	SAEA-RFS	SACC-SVR	SACC-QPA	SACC-RBFN	SACC-GP	SEA-LEEA-RG	IGA	OFSA
$f_1$	median	7.104E+09	1.346E+09	3.281E+09	2.090E+09	<b>9.522E+08</b>	5.058E+09	3.440E+09	1.710E+09
	MAD	5.528E+08	9.454E+07	2.911E+08	1.151E+08	<b>4.166E+07</b>	4.134E+08	1.557E+08	8.374E+07
$f_4$	median	1.198E+12	6.766E+12	7.975E+12	5.398E+12	4.842E+12	1.157E+12	3.376E+11	<b>1.373E+11</b>
	MAD	2.944E+11	2.294E+12	2.513E+12	1.114E+12	9.576E+11	2.769E+11	3.045E+10	<b>1.477E+09</b>
$f_9$	median	1.606E+09	2.015E+09	2.144E+09	1.992E+09	1.980E+09	1.750E+09	<b>8.205E+08</b>	8.860E+08
	MAD	2.320E+08	2.225E+08	2.807E+08	2.699E+08	2.367E+08	2.524E+08	<b>4.552E+06</b>	1.147E+07
$f_{13}$	median	1.089E+11	8.076E+11	2.935E+13	7.608E+11	1.091E+12	8.266E+10	4.274E+10	<b>2.413E+10</b>
	MAD	1.351E+10	4.345E+11	1.649E+13	3.780E+11	3.278E+11	1.371E+10	2.271E+09	<b>5.373E+08</b>
$f_{15}$	median	3.140E+08	3.475E+09	1.184E+10	4.133E+09	4.014E+09	1.849E+08	6.904E+07	<b>4.243E+07</b>
	MAD	8.164E+07	1.475E+09	4.414E+09	1.915E+09	1.570E+09	4.299E+07	8.422E+05	<b>1.533E+06</b>

表 5 IG-OFSA 与其他算法的结果对比

Table 5 Comparison of results obtained by IG-OFSA and other algorithms								
测试函数	评价指标	SAEA-RFS	SACC-SVR	SACC-QPA	SACC-RBFN	SACC-GP	SEA-LEEA-RG	IG-OFSA
$f_1$	median	7.104E+09	1.346E+09	3.281E+09	2.090E+09	<b>9.522E+08</b>	5.058E+09	1.818E+09
	MAD	5.528E+08	9.454E+07	2.911E+08	1.151E+08	<b>4.166E+07</b>	4.134E+08	1.002E+08
$f_2$	median	1.999E+04	<b>7.721E+03</b>	1.141E+04	9.562E+03	9.125E+03	1.713E+04	3.733E+04
	MAD	4.784E+02	<b>9.165E+01</b>	1.764E+02	1.472E+02	1.520E+02	3.944E+02	1.241E+03
$f_3$	median	2.092E+01	2.058E+01	<b>2.053E+01</b>	2.059E+01	2.057E+01	2.078E+01	2.162E+01
	MAD	1.190E-02	9.905E-03	<b>1.100E-02</b>	1.259E-02	1.508E-02	1.793E-02	5.378E-03
$f_4$	median	1.198E+12	6.766E+12	7.975E+12	5.398E+12	4.842E+12	1.157E+12	<b>1.486E+11</b>
	MAD	2.944E+11	2.294E+12	2.513E+12	1.114E+12	9.576E+11	2.769E+11	<b>6.568E+09</b>
$f_5$	median	2.394E+07	2.717E+07	2.787E+07	2.435E+07	2.676E+07	2.616E+07	<b>1.126E+07</b>
	MAD	2.347E+06	2.944E+06	1.854E+06	2.353E+06	3.491E+06	3.296E+06	<b>8.415E+05</b>
$f_6$	median	1.061E+06	1.066E+06	1.069E+06	1.067E+06	1.067E+06	<b>1.060E+06</b>	1.066E+06
	MAD	1.797E+03	1.698E+03	1.654E+03	1.333E+03	2.474E+03	<b>2.370E+03</b>	8.537E+02
$f_7$	median	6.932E+09	2.722E+10	2.945E+11	3.021E+10	3.417E+10	7.225E+09	<b>9.780E+08</b>
	MAD	2.143E+09	5.870E+09	1.478E+11	8.954E+09	1.741E+10	1.986E+09	<b>2.290E+08</b>
$f_8$	median	1.964E+16	2.784E+17	4.484E+17	2.933E+17	3.052E+17	3.880E+16	<b>1.070E+15</b>
	MAD	6.476E+15	8.500E+16	1.649E+17	1.226E+17	1.068E+17	1.326E+15	<b>2.790E+14</b>
$f_9$	median	1.606E+09	2.015E+09	2.144E+09	1.992E+09	1.980E+09	1.750E+09	<b>7.964E+08</b>
	MAD	2.320E+08	2.225E+08	2.807E+08	2.699E+08	2.367E+08	2.524E+08	<b>6.493E+07</b>
$f_{10}$	median	9.455E+07	9.597E+07	9.623E+07	9.607E+07	9.598E+07	<b>9.421E+07</b>	9.526E+07
	MAD	4.278E+05	2.478E+05	3.581E+05	3.167E+05	1.621E+05	<b>2.808E+05</b>	1.589E+05
$f_{11}$	median	8.861E+11	3.275E+12	2.320E+13	4.281E+12	4.113E+12	8.429E+11	<b>4.319E+10</b>
	MAD	3.005E+11	1.425E+12	1.038E+13	1.973E+12	1.689E+12	2.444E+11	<b>1.949E+10</b>
$f_{12}$	median	4.891E+11	8.172E+09	5.944E+10	2.625E+10	2.842E+10	1.011E+12	<b>1.335E+08</b>
	MAD	2.842E+10	7.661E+08	2.422E+09	2.222E+09	2.132E+09	1.361E+10	<b>9.620E+06</b>
$f_{13}$	median	1.089E+11	8.076E+11	2.935E+13	7.608E+11	1.091E+12	8.266E+10	<b>2.254E+10</b>
	MAD	1.351E+10	4.345E+11	1.649E+13	3.780E+11	3.278E+11	1.371E+10	<b>2.318E+09</b>
$f_{14}$	median	1.121E+12	3.786E+12	2.517E+13	4.937E+12	4.832E+12	1.128E+12	<b>2.489E+11</b>
	MAD	1.185E+11	5.884E+11	1.035E+13	1.178E+12	1.042E+12	1.704E+11	<b>4.154E+10</b>
$f_{15}$	median	3.140E+08	3.475E+09	1.184E+10	4.133E+09	4.014E+09	1.849E+08	<b>3.999E+07</b>
	MAD	8.164E+07	1.475E+09	4.414E+09	1.915E+09	1.570E+09	4.299E+07	<b>4.215E+06</b>

11 个测试函数上占优,尤其是在部分可分离函数、重叠函数和完全不可分离函数上更具竞争力。在函数  $f_4, f_7, f_8, f_9, f_{11}, f_{12}, f_{14}$  及  $f_{15}$  上得到的结果能够比其他对比算法低一个数量级,但是,也可以看到 IG-OFSA 在  $f_1, f_2$  及  $f_3$  这 3 个完全可分解函数上的性能较差,这可能是在求解完全可分解函数时,重叠变量增加了子问题之间不必要的交互,反而对解的搜索产生了干扰,使得该算法在可分解函数上难以获得较好解。

## 4 结论

本文提出了 IG-OFSA 用于求解昂贵大规模问题。该算法采用的惯性分组技术能够平衡探索分组方案多样性的能力和对具体分组方案的开发能力,而重叠特征选择技术通过重叠变量,增加了相关变量被划分到同一子问题的概率,加强了子问题之间的合作能力,从而有助于算法对全局最优解的搜索。在 CEC2013 标准函数上的实验结果表明:IG-OFSA 在求解昂贵大规模优化问题时,尤其是在部分可分离函数、重叠函数和完全不可分离函数上能获得更好解。

对于不同类型的昂贵大规模优化问题,最优的子问题的个数  $T$ 、最优的惯性系数  $I$  可能是不同的,因此,今后将考虑根据种群的状态自适应定义参数  $T$  和  $I$ ,以期更好地提升搜索性能。

## 参考文献:

- [1] CHAI T Y, JIN Y C, SENDHOFF B. Evolutionary complex engineering optimization: opportunities and challenges[J]. IEEE Computational Intelligence Magazine, 2013, 8(3): 12-15.
- [2] CAO Y, SUN D F. A parallel computing framework for large-scale air traffic flow optimization[J]. IEEE Transactions on Intelligent Transportation Systems, 2012, 13(4): 1855-1864.
- [3] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[EB/OL]. (2014-09-04) [2023-01-09]. <https://doi.org/10.48550/1409.1556>.
- [4] 梁静,刘睿,瞿博阳,等. 进化算法在大规模优化问题中的应用综述[J]. 郑州大学学报(工学版), 2018, 39(3): 15-21.
- LIANG J, LIU R, QU B, et al. A survey of evolutionary algorithms for large scale optimization problem[J]. Journal of Zhengzhou University (Engineering Science), 2018, 39(3): 15-21.
- [5] 孙迈,孙超利. 基于随机分组集成模型的大规模昂贵

- 问题优化算法[C]//第32届中国过程控制会议(CPCC2021)论文集. 北京:中国自动化学会,2021.
- SUN M, SUN C L. Large-scale expensive problem optimization algorithm based on stochastic grouping integration model[C]//Collection of Papers from the 32nd China Process Control Conference (CPCC2021). Beijing: Chinese Association Automation, 2021.
- [6] YANG Z Y, TANG K, YAO X. Large scale evolutionary optimization using cooperative coevolution[J]. Information Sciences, 2008, 178(15): 2985-2999.
- [7] DONG W S, CHEN T S, TIÑO P, et al. Scaling up estimation of distribution algorithms for continuous optimization[J]. IEEE Transactions on Evolutionary Computation, 2013, 17(6): 797-822.
- [8] POTTER M A, JONG K A. A cooperative coevolutionary approach to function optimization[M]//Parallel Problem Solving from Nature-PPSN III. Berlin: Springer Berlin Heidelberg, 1994: 249-257.
- [9] CHENG R, JIN Y C. A competitive swarm optimizer for large scale optimization[J]. IEEE Transactions on Cybernetics, 2015, 45(2): 191-204.
- [10] 田杰,孙超利,谭瑛,等. 基于多点加准则的代理模型辅助社会学习微粒群算法[J]. 控制与决策, 2020, 35(1): 131-138.
- TIAN J, SUN C L, TAN Y, et al. Similarity-based multipoint infill criterion for surrogate-assisted social learning particle swarm optimization[J]. Control and Decision, 2020, 35(1): 131-138.
- [11] YANG Q, CHEN W N, DA DENG J, et al. A level-based learning swarm optimizer for large-scale optimization[J]. IEEE Transactions on Evolutionary Computation, 2017, 22(4): 578-594.
- [12] De FALCO I, DELLA CIOPPA A, TRUNFIO G A. Investigating surrogate-assisted cooperative coevolution for large-scale global optimization[J]. Information Sciences, 2019, 482: 1-26.
- [13] FU G X, SUN C L, TAN Y, et al. A surrogate-assisted evolutionary algorithm with random feature selection for large-scale expensive problems[C]//International Conference on Parallel Problem Solving from Nature. Cham: Springer, 2020: 125-139.
- [14] SUN M, SUN C L, LI X B, et al. Surrogate ensemble assisted large-scale expensive optimization with random grouping[J]. Information Sciences, 2022, 615: 226-237.
- [15] Van Den BERGH F, ENGELBRECHT A P. A cooperative approach to particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 225-239.
- [16] CHEN W X, WEISE T, YANG Z Y, et al. Large-scale

- global optimization using cooperative coevolution with variable interaction learning [C] // International Conference on Parallel Problem Solving from Nature. Berlin: Springer, 2010: 300–309.
- [17] OMIDVAR M N, LI X D, MEI Y, et al. Cooperative coevolution with differential grouping for large scale optimization[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(3): 378–393.
- [18] OMIDVAR M N, YANG M, MEI Y, et al. DG2: a faster and more accurate differential grouping for large-scale black-box optimization[J]. IEEE Transactions on Evolutionary Computation, 2017, 21(6): 929–942.
- [19] YANG Z Y, TANG K, YAO X. Multilevel cooperative coevolution for large scale optimization [C] // 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). Piscataway: IEEE, 2008: 1663–1670.
- [20] STORN R, PRICE K. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces [J]. Journal of Global Optimization, 1997, 11(4): 341–359.
- [21] KLEIJNEN J P C. Kriging metamodeling in simulation: a review [J]. European Journal of Operational Research, 2009, 192(3): 707–716.
- [22] ZHAN D W, XING H L. A fast Kriging-assisted evolutionary algorithm based on incremental learning [J]. IEEE Transactions on Evolutionary Computation, 2021, 25(5): 941–955.

## An Inertial Grouping and Overlapping Feature Selection Assisted Algorithm for Expensive Large-scale Optimization Problems

DENG Chuanyi<sup>1</sup>, SUN Chaoli<sup>2</sup>, LIU Xiaotong<sup>2</sup>, ZHANG Xiaohong<sup>3</sup>, LI Chunpeng<sup>4</sup>

(1. School of Applied Science, Taiyuan University of Science and Technology, Taiyuan 030024, China; 2. School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China; 3. School of Economics and Management, Taiyuan University of Science and Technology, Taiyuan 030024, China; 4. Shanxi Jicheng Technology Co., Ltd., Taiyuan 030000, China)

**Abstract:** Challenges in expensive large-scale optimization problems, such as high coupling between variables, easy falling into local optimal solution, and computationally expensive objective function, resulted in the difficulty to achieve the global optimal solution. An inertial grouping and overlapping feature selection technique for cooperative coevolutionary (IG-OFSA) algorithms was proposed to solve expensive large-scale optimization problems. In the proposed algorithm, firstly, a large-scale optimization problem was decomposed into several low-dimensional overlapping sub-problems by using overlapping feature selection technology, and each sub-problem was optimized independently with the assistance of a surrogate model. Then, promising solutions found for each sub-problem would be merged into a context vector for expensive objective evaluation. In addition, an inertial grouping technology was used to control the frequency of regrouping during the optimization to extend the cycle of exploitation of the grouping scheme, and correspondingly improved the performance of optimization. The performance of IG-OFSA was tested on 15 CEC2013 benchmark problems and compared with three state-of-the-art algorithms. The experimental results showed that the performance of IG-OFSA was competitive to solve the expensive large-scale optimization problem, especially, good for solving problems with partially separable, overlapping or completely non-separable decision variables.

**Keywords:** large-scale optimization; expensive problems; overlapping feature selection; inertial grouping; surrogate models; cooperative coevolutionary