

基于形式概念分析的大规模全局协同进化优化算法

马连博¹⁾ 常凤荣¹⁾ 张桓熙¹⁾ 王兴伟²⁾ 黄敏³⁾ 郝飞⁴⁾

¹⁾(东北大学软件学院 沈阳 110819)

²⁾(东北大学计算机科学与工程学院 沈阳 110819)

³⁾(东北大学信息科学与工程学院 沈阳 110819)

⁴⁾(陕西师范大学计算机科学学院 西安 710062)

摘要 随着复杂网络及大数据技术的发展,大规模全局优化方法已成为复杂工程系统的重要支撑技术.解决大规模全局优化问题的关键在于如何识别决策变量之间的相互依赖关系并根据这些依赖关系进行有效的变量分组.针对该问题,本文提出了一种基于形式概念分析的大规模全局协同进化优化算法,首次将数据分析领域的形式概念分析思想引入决策变量依赖关系分析和自适应分组过程中,进而处理大规模全局优化问题.该算法基于协同进化框架,利用形式概念分析的外延与内涵结构特征,将决策变量转换为形式概念文本,通过文本向量矩阵运算进行决策变量的自适应分组,使其满足高内聚低耦合原则.最后,该算法在大规模(1000维)CEC2013测试集上进行了性能对比实验.实验结果表明,该算法在11个测试问题上(总计15个测试问题)获得了最好的解精度,同时求解的收敛性达到最优.

关键词 形式概念分析;协同进化;进化计算;优化算法;大规模全局优化

中图分类号 TP18 **DOI号** 10.11897/SP.J.1016.2021.01310

Formal Concept Analysis based Grouping Co-Evolutionary Optimization Algorithms for Large-Scale Global Optimization

MA Lian-Bo¹⁾ CHANG Feng-Rong¹⁾ ZHANG Huan-Xi¹⁾ WANG Xing-Wei²⁾ HUANG Min³⁾ HAO Fei⁴⁾

¹⁾(Software College, Northeastern University, Shenyang 110819)

²⁾(College of Computer Science, Northeastern University, Shenyang 110819)

³⁾(College of Information Science and Engineering, Northeastern University, Shenyang 110819)

⁴⁾(School of Computer Science, Shaanxi Normal University, Xi'an 710062)

Abstract With the development of complex networks and big data technologies, large-scale global optimization has become an important supporting technology for complex engineering systems. Evolutionary algorithm is generally used as a search engine to find the global optimal solution in a complex high-dimensional space for complex engineering optimization systems. However, evolutionary algorithms are difficult or even unable to deal with these problems, with the increase of decision variables and the complexity of the coupling relationship between the variables. Strictly speaking, it is a heuristic search algorithm based on natural biological evolution mechanism without the gradient information of the target problem. It has good robustness and is suitable for solving complex NP-hard problems. For small and medium-scale optimization problems, the evolutionary algorithm has achieved excellent performance in various industrial application systems. It can effectively handle complex optimization scenarios such as various nonlinearities, strong

收稿日期:2020-08-15;在线发布日期:2020-12-31.本课题得到国家自然科学基金(61773103,61702317)资助.马连博,博士,教授,中国计算机学会(CCF)会员,主要研究领域为进化计算、机器学习、工业系统优化. E-mail: malb@swc.neu.edu.cn.常凤荣(通信作者),硕士研究生,主要研究方向为进化计算和智能优化算法. E-mail: 1738919936@qq.com.张桓熙,本科生,主要研究方向为智能优化算法.王兴伟,博士,教授,国家杰出青年科学基金获得者,主要研究领域为计算机网络与机器学习.黄敏,博士,教授,国家杰出青年科学基金获得者,长江学者,主要研究领域为调度优化、物流仿真等.郝飞,博士,副教授,主要研究方向为社交计算、普适计算、大数据分析和处理.

couplings, and mixed variables. However, due to the limitation of search ability, evolutionary algorithm is also difficult to obtain satisfactory results in terms of solution quality and convergence efficiency when the scale of decision variables exceeds a certain order of magnitude. Even if it uses an improved global optimization operator strategy. Therefore, how to design an efficient large-scale global evolutionary optimization algorithm is an urgent problem to solve the application of complex engineering systems in the big data environment. At present, one of the main ideas for dealing with large-scale global optimization problems is to use grouping strategies to transform high-dimensional optimization problems into low-dimensional sub-problems to solve. Cooperative coevolution is an effective processing framework for grouping dimensionality reduction. The main principles: first, according to the coupling dependence between variables, the original high-dimensional problem is transformed into several low-dimensional sub-problems. Second, each sub-problem is solved separately. Finally, the local optimal solutions are combined into a global optimal solution. This method has achieved better performance in various large-scale global optimization problems. Compared with the conventional evolutionary algorithm structure, the cooperative coevolution framework has a stronger ability to solve large-scale global optimization and complex engineering problems. To sum up, the key to solving large-scale global optimization problems is how to identify the interdependencies between decision variables and group them effectively according to these relationships. In response to this key issue, this paper proposes a large-scale global co-evolutionary optimization algorithm based on formal concept analysis. For the first time, formal concept analysis ideas in the field of data analysis are introduced into the process of decision variable dependency analysis and adaptive grouping, and then processed large-scale global optimization problem. Based on the framework of co-evolution, the algorithm uses the extent and intent structure characteristics of formal concept analysis to convert decision variables into formal contexts, and realizes adaptive grouping of decision variables through text vector matrix operations. The grouping result meets the principle of high cohesion and low coupling. Finally, the algorithm performance comparison experiment was conducted on the large-scale (1000-dimensional) CEC2013 benchmark functions. Experimental results show that the algorithm achieves the best solution accuracy on 11 test problems (a total of 15 test problems), and the convergence of the solution is optimal.

Keywords formal concepts analysis; cooperative coevolution; evolutionary computation; optimization algorithm; large-scale global optimization

1 引言

1.1 研究问题

随着工业互联网与大数据等技术的发展,工业工程系统的规模和复杂性日益增加^[1-2],大规模全局优化(Large Scale Global Optimization, LSGO)技术受到了广泛关注. LSGO 是复杂工程优化系统的重要组成部分,一般采用进化算法(Evolutionary Algorithm, EA)作为搜索引擎,在复杂高维空间中寻找全局最优解. 然而,随着工程问题中决策变量的增加以及变量之间相互耦合关系的复杂化,EA 算

法难以处理,甚至无法处理这些问题. 该情况也称为维度灾难问题^[3-4]: 首先,目标问题的搜索空间随着决策变量数量的增长而呈指数级扩大,导致优化过程的局部最优解不断增加;其次,变量之间的耦合(或交互)关系更加复杂,难以使用已有 EA 进行降维优化. EA 是基于自然生物进化机制的启发式搜索算法,无需目标问题的梯度信息,具有较好的鲁棒性,适合求解复杂 NP 难问题^[5]. 对于中小规模的优化问题,EA 算法已在各类工业应用系统中取得了优异的性能表现,能够有效处理各种非线性、强耦合、混合变量等复杂优化场景^[5-6]. 然而,当目标问题的决策变量规模超过一定数量级(例如 1000 维)时,

常规 EA 算法(通常指针对维数小于 100 的问题而开发的算法^[7-8])由于搜索能力有限,即使采用改进的全局优化算子策略,在解的质量和收敛效率方面也难以获得满意的效果^[9-12]. 因此,如何设计高效的大规模全局 EA 优化方法是大数据环境下的复杂工程系统需要解决的一个迫切问题.

1.2 研究现状

目前处理 LSGO 问题的一个主要思路为采用分组策略将高维优化问题转化为低维的子问题来求解. 该方法是基于生物群体的社会分工协作机制,利用分治思想,对决策变量进行分组降维处理. 其中,协同进化(Cooperative Coevolution, CC)是一种有效的分组降维处理框架^[13]. 主要原理为:根据变量间的耦合依赖关系,将原始的高维问题转化为若干个低维的子问题,并对每个子问题分别求解,最后将各个局部最优解组合成全局最优解. 该方法在各种 LSGO 问题中取得了较好的求解性能. 相对于常规的 EA 算法结构来说,CC 框架具有更强的解决大规模复杂工程问题的能力. 因此本文主要基于 CC 框架,设计新型的分组策略与协同进化处理机制.

传统 CC 框架处理 LSGO 问题的主要步骤如下:

- (1) 将一个高维决策向量分解为 n 个子分量(子组);
- (2) 采用合适的进化优化策略对各子分量进行优化;
- (3) 集成所有子组的最优解,形成最终的可行解.

对于可分离的 LSGO 问题,决策变量可以被分解为若干个不相交的子分量^[2]. 每个子分量代表一个独立的子问题,各子问题的优化过程互不影响. 然而,很多实际 LSGO 问题是不可分离的^[3],即所有决策变量之间都存在一定的耦合依赖关系,而无法根据依赖关系进行分组. 对这种问题,很难采用传统分组方法进行降维处理,需要设计新的分组策略与协同进化机制.

目前为止,基于不同的分组学习策略,学者提出了一系列改进的 CC 算法^[1,14-15]. 例如,DECC-G^[14]采用随机分组策略,协同优化相互依赖的变量. 然而,随着决策空间的扩大,该算法会面临一个问题:相互作用的变量被分入同一组的概率大大降低,而不具有相互依赖关系的变量放入同一组的概率却有所提高. 实验结果表明^[15],随机分组策略由于随机性过大,处理大规模耦合变量时优化效率和求解精度都较低. 为此,有学者将链接学习引入 CC 框架中,提出变量交互学习分组^[16]算法:在优化之前基

于链路学习策略检测变量相关性. 该类方法还包括 CCVIL^[16]、Delta 学习^[17]、差分分组^[18]、全局差分分组^[19]和差分分组 II^[20]等. 实验结果表明,与随机分组相比,链接学习方法可获得明显的性能改进^[21],特别是在可分离的 LSGO 问题上求解结果较好. 然而,该算法不能高效地处理不可分离的 LSGO 问题^[21-22].

为了解决上述问题,学者们提出了基于贡献的协同进化(CBCC)^[23]方法及其改进算法,例如 CCFR 算法^[24]、dCC 算法^[25]. 该类算法的主要思想为先评估各个子组对全局适应度的贡献度,进而根据该贡献度来分配子种群的计算资源. 然而,子组贡献值会在进化过程中逐渐累积,导致某些贡献值过小的子种群无法获得计算资源而进化停滞,陷入局部最优或早熟收敛. 此外,学者还提出了基于阈值的分组策略:通过预先设置子组的相关阈值对变量分组. 例如 RDG^[26]、RDG3^[27]算法和 DGSC^[28]算法. RDG3 通过设置阈值规定子组的大小来解决重叠的 LSGO 问题. DGSC 算法预先设定子组的数量,利用聚类对决策变量进行分组,可以避免分组不均匀问题,节省计算空间. 但该类算法忽略了变量之间的耦合依赖关系,容易将密切依赖的变量划分至不同子组,难以达到理想的优化效果.

总体来说,CC 框架是处理 LSGO 问题的有效方法:可以将任何决策变量通过特定的分组方法静态地或动态地分配到不同的子组中,将复杂问题分解为简单子问题. 在分组过程中,每个子组代表一个独立的子问题,各子问题的优化过程互不影响,最后结合所有子组的解,形成最终的优化结果. 据此,变量分组策略的设计对最终优化性能有显著影响^[18]. 此外,LSGO 问题中大量的决策变量之间存在显式或隐式的相互依赖关系,增加了分组的难度^[29-34]. 因此,设计合理有效的分组方法是解决 LSGO 问题的一个关键点.

1.3 研究动机

形式概念分析(Formal Concept Analysis, FCA)是一种功能强大的数据分析技术^[35],最初应用于化学结构检索,用来揭示有机化学品相似的结构和功能信息^[36]. 随后,FCA 用于处理社区结构检测与社交网络分析问题,用来检测并可视化社交网络的社区结构^[37],提供清晰的拓扑结构视图^[38-39]. 值得注意的是,社交网络的节点拓扑结构与 LSGO 问题的变量关系具有相似性. 社交网络中,用户具有一定社交关系(例如用户有共同的兴趣或相似的资料),这

些用户节点会进行相互之间的社交互动. 此类社区结构具有如下特点: 互动较为密切的节点代表连接密集的顶点, 处于同一个社区; 而不同社区顶点之间的互动较少, 连接比较稀疏^[40-42]. 这些用户及相互之间的互动关系可映射为 LSGO 的变量节点及相互依赖关系. 因此, 我们可以利用社区结构检测原则来处理 LSGO 问题的分组问题, 寻找具有共同属性的节点集合.

基于以上考虑, 我们利用 FCA 技术对 LSGO 决策变量进行分组优化, 并将其引入 CC 框架中, 衍生出一种新的 LSGO 优化算法. 具体地, 我们利用网络拓扑结构表示决策变量之间的不同关系, 基于 FCA 思想将连接密集的决策变量划至同一组, 同时遵循高内聚低耦合原则, 保证子组内部具有较高的相关度而子组间的相关度尽可能低^[28].

1.4 本文贡献

本文的主要思路: 将 LSGO 变量分组问题映射为复杂网络的结构检测问题; 基于 FCA 方法及相关矩阵运算将具有相同外延和内涵的概念变量(第 2 部分详细介绍)作为一个子组处理; 将上述分组策略嵌入 CC 框架, 形成新的 LSGO 优化算法.

本文的主要贡献包括:

(1) 提出基于 FCA 的大规模变量分组学习方法(FCA-based Grouping, FCA-G). FCA-G 主要基于 FCA 来分析变量之间的依赖关系并进行分组. 该方法能够检测出高维空间中依赖关系较为密切的决策变量并将其划为一组, 使得同一子组的变量具有较高的相关度而子组之间的相关度尽可能低.

(2) 提出了一种改进的 FCA 方法, 构建基于对象-对象关系的形式文本概念. 概念变量被重新定义为基础概念和添加概念, 并可进行衍生操作, 进而提高优化效率.

该工作的首创性在于: 在进化计算领域内, 首次引入 FCA 与网络拓扑相结合的思路处理复杂 LSGO 问题. 实验结果表明, 本文所提出的方法在部分可分离、重叠以及完全不可分离的 LSGO 问题上获得了较好的性能表现.

1.5 文章结构

本文的结构如下: 第 2 节介绍 LSGO 问题, 形式概念分析(FCA)以及拓扑结构等背景知识; 第 3 节阐述改进的形式概念分析、基础概念与添加概念以及提出的 FCA-G 算法; 第 4 节通过实验验证 FCA-G 在 LSGO 测试集上的有效性与优势; 第 5 节得出结论以及对未来研究的展望.

2 背景知识

2.1 LSGO 问题

根据决策变量之间的相互依赖关系, LSGO 问题可以分为可分离的 LSGO 问题和不可分离的 LSGO 问题两类. 其中, 可分离 LSGO 问题可定义^[43]为

$$\arg \min_x f(\mathbf{x}) = (\arg \min_{x_1} f(\mathbf{x}_1, \dots), \dots, \arg \min_{x_m} f(\dots, \mathbf{x}_m)) \quad (1)$$

其中, $\mathbf{x} = (x_1, \dots, x_n)$ 是一个 n 维的决策向量, $\mathbf{x}_1, \dots, \mathbf{x}_m$ 是 \mathbf{x} 中不存在依赖关系的子向量, 并且 $2 \leq m \leq n$.

在 LSGO 问题中, 任意两个决策变量之间的依赖关系可以根据两者的差分关系来检测^[21, 30-31]. 具体地, 决策变量之间的依赖关系可定义为:

定义 1. 显式交互. 给定一个函数 $f(x_1, \dots, x_n)$, 若存在 $\mathbf{x}, a_1, a_2, b_1, b_2$ 满足式(2), 则认为变量 x_i 和 x_j 是显式交互关系:

$$\begin{aligned} f(\mathbf{x})|_{x_i=a_2, x_j=b_1} < f(\mathbf{x})|_{x_i=a_1, x_j=b_1} \wedge \\ f(\mathbf{x})|_{x_i=a_2, x_j=b_2} > f(\mathbf{x})|_{x_i=a_1, x_j=b_2} \end{aligned} \quad (2)$$

其中, $f(\mathbf{x})|_{x_i=a_2, x_j=b_1} \equiv f(x_1, \dots, x_{i-1}, a_2, \dots, x_{j-1}, b_1, \dots, x_n)$. 若存在 $\mathbf{x}, a_1, a_2, b_1$ 和 b_2 ($a_1 \neq a_2, b_1 \neq b_2$) 使得 $[f(\mathbf{x})|_{x_i=a_2, x_j=b_1}, f(\mathbf{x})|_{x_i=a_1, x_j=b_2}]$ 与 $[f(\mathbf{x})|_{x_i=a_1, x_j=b_1}, f(\mathbf{x})|_{x_i=a_2, x_j=b_2}]$ 之间建立了强支配关系^[31, 44], 则 x_i 与 x_j 显式交互.

变量被视为顶点, 变量之间的显式交互关系被视为连接顶点的边. 接下来, 所有变量之间的交互关系可以用图形表示. 当且仅当两个变量在图中彼此显式交互时, 两个变量之间会存在一条边. 根据图的特征, 我们可以检测顶点之间是否存在边. 因此, 我们可以很容易定义任何两个变量之间的其它交互关系.

定义 2. 隐式交互. 给定一个函数 $f(x_1, \dots, x_n)$, 若变量 x_i 和 x_j 没有建立直接的强支配关系, 但都与同一变量建立了强支配关系, 则称 x_i 与 x_j 隐式交互.

定义 3. 独立. 给定一个函数 $f(x_1, \dots, x_n)$, 若变量 x_i 与 x_j 既不存在显式交互也不存在隐式交互, 那么称它们为独立关系.

基于以上三个定义, LSGO 问题可以进一步划分为(我们以 10 个决策变量为例):

(1) 不可分离问题, 即不存在独立的变量, $m=1$, 如图 1 所示;

(2) 部分可分离问题,即存在部分独立的变量, $1 < m < n$,如图 2 所示;

(3) 完全可分离问题,即所有的变量均独立, $m = n$,如图 3 所示.

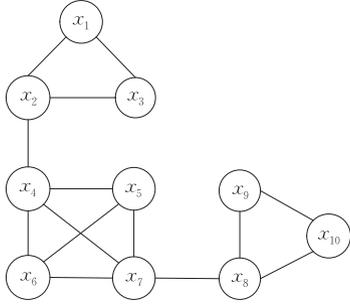


图 1 不可分离 LSGO 问题示意图

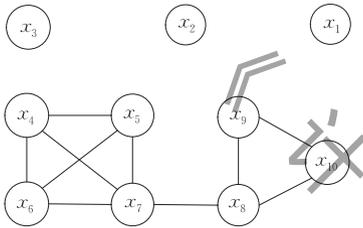


图 2 部分可分离 LSGO 问题示意图

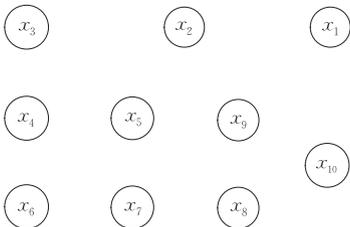


图 3 完全可分离 LSGO 问题示意图

在一个不可分离的 LSGO 问题中,所有决策变量之间都有显式或隐式的交互关系,这也会导致一些特殊情况的发生.例如,完全不可分离的 LSGO 问题和重叠的 LSGO 问题,这两个问题难以归类.

2.2 形式概念分析(FCA)

形式概念分析(FCA)是起源于哲学概念的一种数据分析技术.在 FCA 中,一个形式概念由外延和内涵两部分组成.其中,外延是指属于某一概念的所有对象的集合,内涵则是外延中所有对象共同拥有的特征的集合.概括来说,形式概念表示了域中对象与属性之间的关系^[45].FCA 的具体描述如下:

定义 4^[45-46]. 形式文本.形式文本 K 被定义为 $K = (G, M, I)$,其中 $G = \{x_1, x_2, \dots, x_n\}$ 是对象的集合, $M = \{a_1, a_2, \dots, a_m\}$ 是属性集合, I 是 G 与 M 的二元关系. $(x, a) \in I$ 表示对象 x 具有属性 a , $(x, a) \notin I$

表示对象 x 不具有属性 a ,其中 $x \in G, a \in M$.

如表 1 所示的形式文本: $G = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $M = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8\}$, I 表示 G 与 M 的二元关系.其中, $I(x_1, a_7) = 1$ 表示对象 x_1 具有属性 a_7 , $I(x_2, a_7) = 0$ 表示对象 x_2 不具有属性 a_7 .

表 1 形式文本

G/M	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
x_1	1	0	1	0	0	0	1	0
x_2	1	1	1	1	0	0	0	0
x_3	0	1	1	0	0	1	1	0
x_4	0	1	0	1	1	0	0	0
x_5	0	1	0	1	0	1	0	1
x_6	1	0	0	1	1	0	1	1

定义 5^[45-46]. 概念.若形式文本是一个三元组 $K = (G, M, I)$,其中 G 与 M 均是集合, $I \subseteq G \times M$ 是二元关系. G 中的元素称为对象, M 中的元素称为属性.

$$A^\uparrow = \{m \in M \mid \forall g \in A, (g, m) \in I\}, A \subseteq G \quad (3)$$

$$B^\downarrow = \{g \in G \mid \forall m \in B, (g, m) \in I\}, B \subseteq M$$

那么,满足以上关系的集合称为概念.每一个概念用一个集合对来表示,例如 (A, B) .其中, $A \subseteq G, B \subseteq M, A^\uparrow = B$ 并且 $B^\downarrow = A$. A 称为该概念的外延, B 称为该概念的内涵; A^\uparrow 称为 A 的内涵, B^\downarrow 称为 B 的外延.

2.3 拓扑结构

网络的拓扑结构是将所有实体抽象成“点”,把连接实体的线路作为“线”,用图的形式来表示点与线之间的关系.这些图被称为拓扑结构图.在本文中,我们利用网络拓扑结构的思想直观地表示和分析 LSGO 问题变量间的相互依赖关系:将决策空间中的变量抽象为网络节点,将变量之间的依赖关系抽象为线.这些点与线构建了一个反映 LSGO 变量属性的复杂网络,进而可以用网络拓扑相关的模型与方法进行变量分组处理.

3 基于形式概念的分组算法(FCA-G)

该章节首先给出相关定义,进而阐述 FCA-G 算法的基本原理与实现过程.

3.1 完全拓扑结构

“数”“形”结合的思想是解决复杂问题的有效方法,而拓扑结构也被认为是解决复杂网络问题的有效工具^[41]:它可以很好地反应数据之间的关系,将实体之间的具体关系用概括性的拓扑结构图表示,

可以简化问题^[39-41]. 基于此, 为有效地分析和解决复杂优化问题, 可以将大规模优化模型映射为一个复杂网络, 进而采用拓扑结构相关的方法对其要素进行分析和处理. 例如, 任意两个决策变量之间的相互依赖关系可以用拓扑结构来表示, 即拓扑结构是变量之间关系的显式表示. 那么, 完全拓扑结构就是显式交互的决策变量所构成的完全图.

完全图是指每一个顶点都与其它顶点有一条边连接的无向图, 也可定义为:

定义 6. 完全拓扑结构. 给定一个无向图 $M=(V, E)$, M 中的完全拓扑结构 S 是 M 的子集, 即 $S \subseteq M$. 对于任意两个顶点 $v_i, v_j \in S$, 均存在一条边将 v_i 与 v_j 相连.

根据以上定义, 我们可以得出以下结论: (1) 完全拓扑结构是在无向连通图的基础上形成的, 它是连通图的一部分. 当然, 连通图本身也可以是一个完全拓扑结构, 其内部也可以包含多个完全拓扑结构; (2) 完全图是任意两个顶点之间都有连接的完全拓扑结构. 在 LSGO 问题中, 任意两个变量可看作连通图中的任意两个顶点, 变量之间的显式交互关系可替代连接顶点的线. 基于此, 可以将一个具体的 LSGO 问题转换为特定的拓扑结构(无向连通图). 在复杂网络中, 完全图内的每个顶点都与其它顶点相连, 而图与图之间却具有较少的连接, 即满足高内聚低耦合特性. 那么, 由 LSGO 问题中的决策变量及其依赖关系所构成的完全拓扑结构也具有同样的性质: 同一完全图内的决策变量具有更加密集的交互关系, 而图之间的关系则趋向于稀疏.

下面举例说明利用拓扑结构思想处理 LSGO 问题的优势. 对于一个给定的函数优化问题:

$$f(x) = x_1 x_2 x_3 + x_2 x_4 + x_4 x_5 x_6 x_7 + x_7 x_8 + x_8 x_9 x_{10},$$

$$x \in [-1, 1],$$

我们可知该问题是一个完全不可分离的优化问题, 若直接求解, 则算法比较复杂, 且难以找到最优解. 然而, 我们发现该问题中的一部分变量具有隐式的、甚至是显式的交互关系. 若将函数 $f(x)$ 的表达式转化为相应的拓扑结构, 如图 1 所示, 我们可以直观地分析出变量之间的相互关系, 并利用分组学习策略对其快速求解. 由图 1 可知, 函数 $f(x)$ 的决策变量空间包含 3 个完全拓扑结构图, 分别为 $\{1, 2, 3\}$, $\{4, 5, 6, 7\}$ 和 $\{8, 9, 10\}$. 这 3 个完全拓扑结构图之间的联系相对稀疏, 近似忽略, 而其内部的联系相对密集. 因此, 在该问题中, 我们可以将决策变量划分为

3 个子组, 每个子组对应于图 1 中的一个完全拓扑结构, 分别求出这 3 个拓扑结构的最优解, 并形成最终的全局最优解.

3.2 改进的 FCA 与新邻接矩阵

(1) 研究动机

常规 FCA 方法获得的形式文本包含对象和属性两个标签. 此形式文本的内容是通过判断对象与属性之间的关系得到的包含“0”或“1”的统计数据. 形式文本的目的是分析出 LSGO 问题中变量之间的相互关系, 并尽可能合理地将这些变量进行分组处理, 降低问题的求解难度. 原则上, 我们首先可以通过定义 1、2、3 判断变量之间的交互关系. 然后, 将任意两个或多个变量之间的关系以形式文本方式来表示.

然而, 传统的形式文本结构是基于对象-属性关系, 并不符合 LSGO 问题变量的映射. 这就需要将对象-属性关系转化为对象-对象关系, 进而构建新的形式文本.

(2) 构建改进的形式文本与邻接矩阵

改进的 FCA 构建过程如下:

首先, 将所有的决策变量映射为 FCA 中的对象, 将变量之间的依赖关系映射为对象-对象的二元关系. 若两个变量之间存在显式交互关系, 则相应的二元关系为“1”, 否则为“0”. 最终可得到如表 2 所示的形式文本(与 3.1 节中“一个给定的函数优化问题”公式所对应). 进一步地, 表 2 的形式文本可以用三元组 K 表示, $K=(G, G, I)$, 其中 $G=\{x_1, x_2, \dots, x_{10}\}$, I 是 G 与 G 的二元关系. $(x_i, x_j) \in I$ 表示变量 x_i 与 x_j 显式交互, $(x_i, x_j) \notin I$ 表示变量 x_i 与 x_j 独立, 其中 $x_i \in G, x_j \in G$.

表 2 改进的形式文本

G/G	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
x_1	1	1	1	0	0	0	0	0	0	0
x_2	1	1	1	1	0	0	0	0	0	0
x_3	1	1	1	0	0	0	0	0	0	0
x_4	0	1	0	1	1	1	1	0	0	0
x_5	0	0	0	1	1	1	1	0	0	0
x_6	0	0	0	1	1	1	1	0	0	0
x_7	0	0	0	1	1	1	1	0	0	0
x_8	0	0	0	0	0	0	1	1	1	1
x_9	0	0	0	0	0	0	0	1	1	1
x_{10}	0	0	0	0	0	0	0	1	1	1

为了有效地控制决策变量和实现矩阵运算, 我们定义一个新邻接矩阵.

定义 7. 新邻接矩阵. 令 G 有 n 个变量, 这些

变量按照从 x_1 到 x_n 的顺序排列, 则新邻接矩阵 T 可以统一表示为

$$T = \begin{cases} t_{ij} = 1, & \text{if } x_i \text{ explicit interacts with } x_j, \text{ and } i \neq j \\ t_{ij} = 1, & \text{if } i = j \\ t_{ij} = 0, & \text{其他} \end{cases} \quad (4)$$

其中 $i, j \in [1, 2, \dots, n]$.

新邻接矩阵 T 与拓扑结构中原始邻接矩阵的主要区别在于: T 的所有对角元素都设置为“1”, 而原始邻接矩阵的主对角元素为“0”. 这是考虑到实际问题中任意一个决策变量与其本身都应为显式交互关系.

为了区分原始形式文本, 我们将新邻接矩阵 T 作新的形式文本 $FCA(G) = (G, G, D)$, 即 $FCA(G) \equiv T$. 因此, 与表 2 所对应的新邻接矩阵 T 可进一步表示为式(5).

$$T = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \quad (5)$$

3.3 基础概念与添加概念

矩阵运算是 FCA 的一个重要环节. 为此, 我们根据 LSGO 问题特点, 重定义了 FCA 的矩阵运算, 并根据已有知识衍生出基础概念和添加概念. 最后, 对其进行证明.

定义 8^[46]. 矩阵运算. 一个大小为 $n \times n$ 的矩阵 T 对应于一个包含 n 个变量的形式文本, T' 表示 T 的转置. 令 $C = T \otimes T'$, 则 $c_{ij} = \{n_k \in G \mid t_{ik} = 1, t'_{kj} = 1, k = 1, \dots, n\}$ ($i = 1, \dots, n, j = 1, \dots, n, c_{ij}$ 是矩阵 C 中的元素, t_{kj} 表示矩阵 T 中的元素).

由定义 8 可得到式(6)所示的矩阵运算结果 (“a”代表“10”).

证明. 由定义 8 可得: $c_{ij} = \{n_k \in G \mid t_{ik} = 1, t'_{kj} = 1, k = 1, \dots, n\}$, t_{kj} 表示矩阵 T 中的元素. 进一步可得 $t'_{kj} = t_{jk}$, $c_{ij} = \{n_k \in G \mid t_{ik} = 1, t'_{kj} = 1, k = 1, \dots, n\}$. 因此, c_{ij} 表示同时与第 i 个变量和第 j 个变量具有显式交互关系的变量. 证毕.

$C =$

$$C = \begin{pmatrix} 123 & 123 & 123 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 123 & 1234 & 123 & 24 & 4 & 4 & 4 & 0 & 0 & 0 \\ 123 & 123 & 123 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 24 & 2 & 24567 & 4567 & 4567 & 4567 & 7 & 0 & 0 \\ 0 & 4 & 0 & 4567 & 4567 & 4567 & 4567 & 7 & 0 & 0 \\ 0 & 4 & 0 & 4567 & 4567 & 4567 & 4567 & 7 & 0 & 0 \\ 0 & 4 & 0 & 4567 & 4567 & 4567 & 45678 & 78 & 8 & 8 \\ 0 & 0 & 0 & 7 & 7 & 7 & 78 & 789a & 89a & 89a \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 & 89a & 89a & 89a \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 & 89a & 89a & 89a \end{pmatrix} \quad (6)$$

定理 1^[46]. 一个 $m \times n$ 的矩阵 T 对应于一个形式文本, T' 表示 T 的转置, 令 $C = T \otimes T'$, 令 $X = \{x \in G \mid x I c_{ij}\}$, 则 (X, c_{ij}) 为基础概念.

证明. $X^\uparrow = \{m \in M \mid \forall x \in X, (x, m) \in I\} = c_{ij}$, $X \subseteq G$; $c_{ij}^\downarrow = \{x \in G \mid \forall m \in c_{ij}, (x, m) \in I\} = X$, $c_{ij} \subseteq M$, 则 (X, c_{ij}) 是一个概念. 证毕.

通过定理 1 提取出式(6)中的基础概念有: $(\{1, 1\}, \{1, 2, 3\})$; $(\{1, 2\}, \{1, 2, 3\})$; $(\{1, 3\}, \{1, 2, 3\})$; $(\{1, 4\}, \{2\})$; $(\{2, 2\}, \{1, 2, 3, 4\})$; $(\{2, 3\}, \{1, 2, 3\})$; $(\{4, 4\}, \{2, 4, 5, 6, 7\})$; $(\{4, 5\}, \{4, 5, 6, 7\})$; $(\{6, 7\}, \{4, 5, 6, 7\})$ 等.

定理 2^[46]. 若 (X_1, B_1) 和 (X_2, B_2) 均是概念, 则 $((X_1 \cup X_2)^{\uparrow\uparrow}, B_1 \cap B_2)$ 也是一个概念.

证明. (X_1, B_1) 和 (X_2, B_2) 均是概念, 可知 $X_1 = B_1^\downarrow$, $X_1^\uparrow = B_1$, $X_2 = B_2^\downarrow$, $X_2^\uparrow = B_2 \Rightarrow (X_1 \cup X_2)^{\uparrow\uparrow} = (B_1^\downarrow \cup B_2^\downarrow)^\uparrow = B_1 \cap B_2$; $(B_1 \cap B_2)^\downarrow = (X_1^\uparrow \cap X_2^\uparrow)^\downarrow = (X_1 \cup X_2)^{\uparrow\downarrow}$. 因此, $((X_1 \cup X_2)^{\uparrow\uparrow}, B_1 \cap B_2)$ 是一个概念. 证毕.

通过定理 2 提取出的添加概念有 $(\{1, 2, 3\}, \{1, 2, 3\})$, $(\{4, 6, 7\}, \{4, 5, 6, 7\})$, $(\{4, 5, 6, 7\}, \{4, 5, 6, 7\})$ 等.

3.4 分组过程

正如第 1 部分所描述, 大规模全局优化面临两个挑战: (1) 搜索空间随着决策变量数目的增长而呈指数级扩大, 导致优化过程中局部最优解的数量不断增加; (2) 变量之间的相互作用关系更复杂, 使用已有的分组优化策略失效. 处理这两类挑战的核心在于如何有效地识别出变量之间的依赖关系并合理地对其分组. 虽然目前已有很多分组策略^[23-25], 但是它们在某些问题上表现不佳, 例如部分可分离的 LSGO 问题和完全不可分离的 LSGO

问题等.

算法 1. FCA-G 算法整体框架.

BEGIN

1. 当前进化代数 $Gen=0$
 2. 迭代次数 T
 3. 随机生成的种群 POP
 4. 调用 FCA-G 算法
 5. 当前迭代次数 $Cycle=0$
 6. WHILE 评价次数 $Fes < \text{最大评价次数 } Max_Fes$
 7. 当前迭代次数 $Cycle$ 自加 1
 8. FOR $i=1$ TO $group_num / * group_num$ 为子
组数目 $*/$
 9. 其它子组保持不变, 优化 $group_i$
 10. WHILE 当前进化代数 $<$ 迭代次数 T
 11. 在子种群中进行交叉变异等操作^[47]
 12. 评估新种群^[47]
 13. 当前进化代数 Gen 加 1
 14. END WHILE
 15. END FOR
 16. 更新 Fes ^[47]
 17. END WHILE
- END

针对以上两类问题, 我们首次引入形式概念分析与完全拓扑结构相结合的思想, 提出基于形式概念分析的大规模全局协同进化优化算法 (FCA-G), 处理大规模决策变量分组与优化问题. 算法 1 是协同进化优化算法的整体框架. 对于复杂的 LSGO 问题, 由于难以直接获得满足条件的概念, FCA-G 根据相关知识将概念分为基础概念和添加概念, 由基础概念逐渐派生出添加概念 (定理 2), 可以简化获取概念的难度.

FCA-G 算法的具体过程如下:

(1) 将决策变量之间的交互关系抽象为对象-对象间的关系, 并将新的形式文本转化为矩阵 (算法 2 第 2 行).

(2) 执行矩阵运算 (定义 8), 并从结果矩阵中提取基础概念 (算法 3 第 2~8 行). 矩阵运算主要根据数据的位置, 为每个数据赋予唯一标记, 即定义 7 的排序值.

算法 2. FCA-G 算法.

输入: 变量集合 $G = \{x_1, x_2, \dots, x_n\}$

输出: 外延与内涵相等的概念

BEGIN

1. 声明一个空集合 Set
2. 根据定义 1 和定义 7 构建形式文本 $FCA(G)$, 并

转换为相对应的矩阵 T

3. 根据定义 8 做矩阵运算, 获得 C
 4. BEGIN
 5. 声明一个空集合 $conceptset$
 6. 调用基础概念算法, 将获得的值赋给集合 $conceptset$
 7. 调用添加概念算法
 8. END
 9. FOR 集合 $conceptset$ 中的每一个概念
 10. IF 概念的外延与内涵相等
 11. 将当前概念加入到集合 Set
 12. END IF
 13. END FOR
- END

(3) 为避免冗余操作, 算法在向集合 $conceptset$ 添加基础概念之前, 需要检验待加入的概念是否已存在于集合 (算法 3 第 4~5 行). 这样设计能够保证底层的概念具有唯一性. 另外, 由于矩阵 C 为对称矩阵, 提取基础概念时只需检测上三角元素 (算法 3 第 3 行). 所有添加概念均通过基础概念的集合运算获得 (算法 4 第 4~5 行). 同样, 在确认添加概念时, 也要进行检测 (算法 4 第 6~8 行), 保证集合 $conceptset$ 中无重复概念.

算法 3. 基础概念.

输入: 矩阵 C

输出: 基础概念 $conceptset$

BEGIN

1. 声明一个空集合 $conceptset$
 2. FOR $i=1$ TO n
 3. FOR $j=i$ TO n
 4. IF 集合 $conceptset$ 不包含当前的基础概念 THEN
 5. 将当前的基础概念添加到集合 $conceptset$
 6. END IF
 7. END FOR
 8. END FOR
- END

算法 4. 添加概念.

输入: 基础概念

输出: 添加概念

BEGIN

1. 将集合 $conceptset$ 赋值给集合 $conceptset1$
2. 声明一个空集合 $conceptset2$
3. DO
4. FOR 在集合 $conceptset1$ 中的任意两个概念 (X_1, B_1) ,
 (X_2, B_2)
5. B_1 与 B_2 取交集, 并赋值给 B
6. IF 集合 $conceptset$ 不包含 B 的外延与 B 所组成的概念
7. 将该概念加入到集合 $conceptset$ 和 $conceptset2$

8. END IF
 9. END FOR
 10. 将集合 *conceptset2* 赋值给集合 *conceptset1*
 11. 清空集合 *conceptset2*
 12. END DO
 13. UNTIL *conceptset1* 为空集
- END

与现有的静态以及随机分组方式不同, FCA-G 将数据分析技术与拓扑结构相结合, 图形化地分析变量之间的内在关系, 将完全图内的决策变量划分至同一子组. 该方法保证了同一子组内部变量均为显式交互关系, 不同子组间为隐式交互或独立关系, 同时也为搜索最优解提供了分组降维支持.

3.5 算法复杂度分析

学者对大规模问题的另一个担忧是时间复杂度. FCA-G 算法的时间复杂度主要体现在获取添加概念的过程. 如算法 4 所示: 算法从 *conceptset1* 中获得添加概念, 并且每向集合 *conceptset* 加入一个添加概念时都需要检测一次当前集合 *conceptset* 是否已经包含已经加入的添加概念. 另外, 为了保证获得所有的概念, 在算法 4 的最外层存在 DO 循环, 其终止条件是 *conceptset1* 中不再含有概念. 因此 FCA-G 的时间复杂度最大为: $O(N^3)$, 其中 N 为决策变量的数目.

4 实验验证

本章节主要通过实验仿真来验证 FCA-G 算法的搜索性能. 本文在测试算法性能时采用了 CEC2013 LSGO 基准测试集. CEC2013 大规模全局优化基准测试函数集被广泛用于测试 LSGO 算法的性能^[48]. 其中, $f_1 \sim f_3$ 用于测试完全可分离的大规模全局优化问题; $f_4 \sim f_{11}$ 用于测试部分可分离大规模全局优化问题 ($f_4 \sim f_7$ 表示有可分离子组的测试函数; $f_8 \sim f_{11}$ 表示无可分离子组的测试函数); $f_{12} \sim f_{14}$ 用于测试重叠的大规模全局优化问题; f_{15} 用于测试完全不可分离的大规模全局优化问题.

FCA-G 与其它对比算法均采用 SaNSDE 作为子组优化引擎. SaNSDE 是差分进化 (DE)^[49] 的变种算法, 在进化过程中根据问题本身的特点自动调整参数. 在参数设置方面, 每个算法在每个基准测试函数上独立运行 25 次. 根据文献^[48, 50] 的建议, 本文将最大评价次数 (*Max_Fes*) 设置为 3×10^6 , 种群

规模为 100, 决策变量的维度为 1000. 为了证明算法 FCA-G 的有效性, 我们使用的受到广泛认同且经典有效的比较算法有:

(1) SaNSDE^[47] 算法, 被广泛地用于优化协同进化 (CC) 算法中的子组.

(2) CBCC3-DG2^[20] 算法, 在著名的差分分组策略的基础上进行了改进, 并将分组策略与基于贡献的协同进化算法相结合.

(3) DECC-G^[14] 算法, 使用随机策略实现分组, 也是其它论文中使用最多的对比算法.

(4) SHADE-ILS^[51] 算法, 是一种新的混合算法. 它迭代地结合了差分进化算法和局部搜索方法, 并表现出了优异的效果.

(5) MOS^[52] 算法, 会自动为每个函数和搜索阶段选择最合适算法的混合算法.

表 3 列出了各个算法 (FCA-G、SaNSDE、CBCC3-DG2 和 DECC-G) 在 CEC2013 LSGO 基准测试函数上的均值对比结果. 根据表 3 提供的数据, 我们可知 FCA-G 算法的整体性能优于其它对比算法. 具体为: FCA-G 在完全可分离测试函数中有 2/3 的优化结果优于 SaNSDE 和 CBCC3-DG2 算法. FCA-G 在部分可分离测试函数中分别有 7/8 的数据优于这三种对比算法. 在完全不可分离测试函数中 FCA-G 优于对比算法 CBCC3-DG2 和 DECC-G.

表 3 FCA-G 与 SaNSDE、CBCC3-DG2 和 DECC-G 整体性比较 (W/T/L 表示胜/平/负的次数)

W/T/L	SaNSDE	CBCC3-DG2	DECC-G
$f_1 \sim f_3$	1/1/1	2/1/0	0/0/3
$f_4 \sim f_{11}$	7/1/0	7/1/0	7/1/0
$f_{12} \sim f_{14}$	3/0/0	3/0/0	2/0/1
f_{15}	0/0/1	1/0/0	1/0/0
Sum	11/2/2	13/2/0	10/1/4

接下来, 我们将结合其它统计的实验数据进行更进一步的对比分析.

4.1 有效性比较

4.1.1 与 SaNSDE、CBCC3-DG2、DECC-G 比较

表 4 给出了算法 FCA-G 以及基础比较算法的实验数据 (“min” 表示最好的优化值, “mean” 表示均值 “std” 表示标准差). 从表 4 可知, DECC-G 算法在问题 f_1, f_2 (完全可分离的测试函数) 中表现最好, 取得了最好的均值. 但在部分可分离和重叠的测试函数中, DECC-G 没有表现出特别大的优势. 这是因为它采用的是随机分组机制, 着重于分组数量与分组大小, 忽略了变量之间的交互性. 在测试函数 f_2

和 f_3 中, 我们提出的算法与其它算法的优化结果相比差别较小. 这也说明了我们的算法在完全可分离的大规模优化问题中也有一定的效果. 当然这也反映出另一个问题: 在 CEC 2013 LSGO 基准测试函数中对完全可分离的 LSGO 问题来说, 在评价次数一定的情况下, 子组的规模与数量对最终的优化结果有影响, 但是不是关键性因素.

表 4 FCA-G, SaNSDE, CBCC3-DG2 和 DECC-G 在 CEC2013 LSGO 基准函数上运行的结果(最佳结果已通过加粗表示)

function	States	SaNSDE	CBCC3-DG2	DECC-G	FCA-G
f_1	min	6.14E+03	3.08E+04	7.59E-07	2.15E+04
	mean	3.52E+04	1.60E+05	3.59E-06	9.63E+04
	std	2.62E+04	2.59E+05	3.18E-06	8.28E+04
f_2	min	8.70E+03	8.86E+03	1.29E+03	8.21E+03
	mean	8.98E+03	9.61E+03	1.31E+03	8.51E+03
	std	3.60E+02	6.94E+02	3.56E+01	3.75E+02
f_3	min	2.08E+01	2.08E+01	2.02E+01	2.08E+01
	mean	2.08E+01	2.08E+01	2.02E+01	2.08E+01
	std	9.09E-03	4.40E-03	4.28E-03	8.66E-03
f_4	min	3.85E+09	5.05E+07	5.43E+10	2.32E+07
	mean	4.14E+09	7.17E+07	9.61E+10	3.69E+07
	std	2.65E+08	3.21E+07	5.07E+10	2.19E+07
f_5	min	2.75E+06	1.65E+06	7.30E+06	1.45E+06
	mean	3.36E+06	1.67E+06	7.95E+06	1.61E+06
	std	5.67E+05	1.55E+05	1.05E+06	1.91E+05
f_6	min	1.05E+06	1.05E+06	1.05E+06	1.05E+06
	mean	1.05E+06	1.05E+06	1.06E+06	1.06E+06
	std	4.77E+03	1.91E+03	3.21E+03	1.48E+03
f_7	min	2.41E+06	2.12E+04	1.76E+08	1.46E+03
	mean	2.99E+06	4.10E+04	2.26E+08	1.71E+03
	std	6.78E+05	3.31E+04	6.17E+07	3.00E+02
f_8	min	1.29E+12	7.55E+09	1.09E+15	3.50E+09
	mean	2.56E+12	4.01E+10	2.18E+15	1.85E+10
	std	1.19E+12	3.12E+10	1.08E+15	2.53E+10
f_9	min	2.52E+08	1.06E+08	3.99E+08	8.35E+07
	mean	2.64E+08	1.46E+08	5.50E+08	1.26E+08
	std	1.24E+07	3.03E+07	9.46E+07	2.90E+07
f_{10}	min	9.29E+07	9.28E+07	9.27E+07	9.27E+07
	mean	9.31E+07	9.36E+07	9.29E+07	9.28E+07
	std	1.59E+05	8.43E+05	1.56E+05	1.17E+05
f_{11}	min	1.14E+08	1.16E+07	3.08E+10	4.25E+06
	mean	1.88E+08	6.99E+07	5.53E+10	8.29E+06
	std	6.76E+07	1.95E+08	2.70E+10	4.05E+06
f_{12}	min	1.43E+04	1.41E+05	3.66E+03	4.44E+03
	mean	7.93E+04	5.73E+05	4.84E+03	4.53E+03
	std	9.80E+04	4.52E+05	2.68E+03	8.85E+01
f_{13}	min	7.99E+07	2.87E+08	5.73E+09	5.51E+07
	mean	1.55E+08	3.61E+08	7.11E+09	6.76E+07
	std	1.21E+08	8.85E+07	1.46E+09	1.37E+07
f_{14}	min	7.03E+07	1.33E+08	4.62E+10	2.82E+07
	mean	1.54E+08	1.89E+08	8.72E+10	3.73E+07
	std	1.22E+08	4.00E+07	3.45E+10	1.30E+07
f_{15}	min	3.15E+06	3.95E+06	7.59E+06	3.84E+06
	mean	3.68E+06	4.29E+06	1.68E+07	4.19E+06
	std	6.07E+05	3.66E+05	1.51E+07	2.63E+05

在部分可分离的测试函数集($f_4 \sim f_{11}$)中, 我们所提出的算法明显优于其它三种对比算法, 其次是 CBCC3-DG2 算法. CBCC3-DG2 采用深度优先搜索策略将具有交互关系的变量划分至同一子组. 对于函数 f_6 来说, 算法 FCA-G 与其它三种算法有着及其接近的优化效果. 结合图 5(f_6), 我们可以看到 FCA-G 算法在优化初期没有达到最好的效果, 随着优化的进行, 在四种算法中逐步取得最好的表现. FCA-G 在函数 f_7, f_9, f_{10} 以及 f_{11} 中也有着惊人的表现, 收敛速度以及收敛值均达到最优. 在重叠以及完全不可分离的测试函数集中, 我们提出的算法同样表现优异. 在 $f_{12} \sim f_{14}$ 这三个测试函数中, FCA-G 在最优解、均值以及稳定性方面都表现出了很好的性能, 优化效果远优于其它对比算法. FCA-G 在函数 f_{15} 中的优化效果也优于 CBCC3-DG2 和 DECC-G.

FCA-G 取得以上优异效果的原因主要在于它所采用的分组思想: FCA-G 将彼此之间有显式交互关系的决策变量(即满足构成完全拓扑结构条件的变量)划分于同一个子组, 相互之间独立(即不存在交互)或者存在隐式交互的变量划为至不同的子组. 这样的分组方式, 保证了在同一子组内的变量相互影响、交互密集, 不同子组间的交互稀疏. 每个子组在不受外界影响的条件下独立进化, 在一定的评价次数内找到最优解. 虽然 SaNSDE、CBCC3-DG2 和 DECC-G 算法也使用了分组策略, 但是它们的分组结果不能满足高内聚低耦合的原则, 即没有保证同一子组内的变量彼此间均是显式交互关系. 因此, 这些分组策略得到的子组内部决策变量不足以构成完全拓扑结构. 这也造成了 SaNSDE、CBCC3-DG2 和 DECC-G 的分组结果不够合理, 影响了最终的优化结果. 将彼此之间为显式交互关系的变量分在同一子组, 使得子组间的依赖性相对较低, 子组内变量的依赖性足以达到构成完全拓扑结构的条件是我们解决 LSGO 问题的出发点, 也是 FCA-G 算法的核心.

4.1.2 与 SHADE-ILS、MOS 比较

接下来, 我们将 FCA-G 与最新的 SHADE-ILS 和 MOS 算法在 CEC2013 LSGO 基准测试函数集上进行对比实验. 由于 SHADE-ILS 和 MOS 算法的原始实验数据没有公布, 我们在 TACO(优化器自动比较工具包)^[53] 网站上进行了对比, 并且参考了 SGCC^[4] 中关于 SHADE-ILS 和 MOS 的数据. FCA-G 算法与 SHADE-ILS 和 MOS 的对比结果如表 5 所示.

表 5 算法 FCA-G 与 SHADE-ILS 和 MOS 比较
(最佳结果已通过加粗表示)

function	FCA-G	SHADE-ILS	MOS
f_1	8.04E+04	2.69E-24	0.00E+00
f_2	8.42E+03	1.00E+03	8.32E+02
f_3	2.08E+01	2.01E+01	9.17E-13
f_4	2.87E+07	1.48E+08	1.74E+08
f_5	1.53E+06	1.39E+06	6.94E+06
f_6	1.06E+06	1.02E+06	1.48E+05
f_7	1.86E+03	7.41E+01	1.62E+04
f_8	7.99E+09	3.17E+11	8.00E+12
f_9	1.05E+08	1.64E+08	3.83E+08
f_{10}	9.28E+07	9.18E+07	9.02E+05
f_{11}	7.61E+06	5.11E+05	5.22E+07
f_{12}	4.56E+03	6.18E+01	2.47E+02
f_{13}	6.55E+07	1.00E+05	3.40E+06
f_{14}	2.07E+07	5.76E+06	2.56E+07
f_{15}	3.81E+06	6.25E+05	2.35E+06

由表 5 可知, SHADE-ILS 的性能表现更好一些, 但是, FCA-G 也具有优势. 在函数 f_4, f_5, f_7, f_8, f_9 和 f_{11} 中, FCA-G 具有比 MOS 更出色的性能, 并且在函数集 f_4, f_8 和 f_9 中优于其他两种算法. 值得注意的是, 这两种对比算法是混合算法: SHADE-ILS 使用重新启用技术, 迭代地结合了差分进化算法和局部搜索方法; MOS 则会自动为每个函数和搜索阶段选择最合适算法. FCA-G 仅采用基本分组策略和 SaNSDE 优化器. 因此, 如果将 FCA-G 与这些先进技术相结合, 则有很大可能取得更好的优化效果.

4.2 性能比较

4.2.1 稳定性比较

实验数据的标准差反映了一组数据与其平均值的偏离程度, 进而可以反映算法的稳定性. 因此, 我们通过比较 FCA-G 算法与 SaNSDE、CBCC3-DG2 和 DECC-G 算法的实验数据标准差来判断其稳定性. 首先, 我们以单个测试函数为基准分析了各个算法的稳定性. 图 4 给出了 FCA-G、SaNSDE、CBCC3-DG2 和 DECC-G 算法在各个测试函数上的实验数据——标准差. 从该图, 我们可以看出 FCA-G 在问题 $f_4, f_5, f_7, f_8, f_{11}, f_{12}, f_{13}$ 和 f_{14} 上取得的标准差(稳定性)要优于其它对比算法. 在 f_2, f_3, f_6, f_9 和 f_{15} 问题中, 各个算法的稳定性基本相同. FCA-G 在以上测试函数中有着较高的稳定性是因为: 它根据 LSGO 问题本身的特点, 利用变量之间的依赖关系实现分组降维. 它将完全拓扑结构内的变量划分至同一子组, 保证了同一子组内的变量有着最紧密的交互关系, 而不是通过设置阈值等忽略变量间内部关系的方式实现分组.

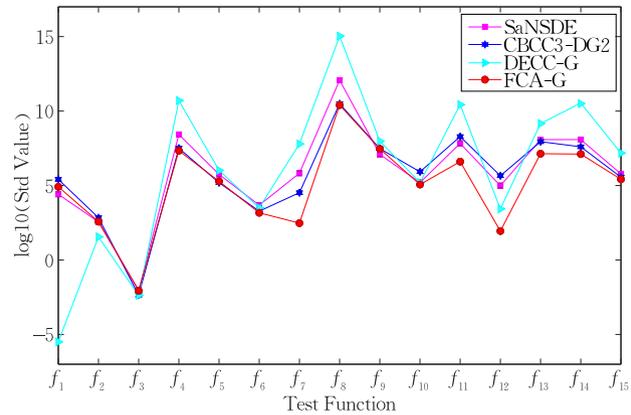


图 4 FCA-G 与 SaNSDE、DECC-G 和 CBCC3-DG2 稳定性比较

在问题 f_1 中, DECC-G 的稳定性要优于其他对比算法, 这得益于它所采用的分组过程以及优化策略: DECC-G 将高维变量随机分解成若干恒定大小的组, 然后采用自适应加权策略, 使用优化引擎优化权重向量. 当子组的规模较小时, DECC-G 能很好地解决问题, 但是随着维度的增加, 子组的规模也逐步扩大, DECC-G 不能够高效地解决问题. 表 4 和图 4 分别反映出 DECC-G 目前在寻找最优解和稳定性方面存在不足之处. DECC-G 在 f_1 和 f_2 测试函数中表现较好, 但是在其他测试函数上性能一般. 另外, 从算法整体的稳定性来看, DECC-G 相比于其它算法波动较大, 稳定性不高.

相对于其它对比算法, CBCC3^[54] 是基于贡献的优化框架, DG2^[20] 是一种改进的微分分组 (DG)^[2] 算法, 首先通过估计舍入误差的大小来寻找一个可靠的阈值, 然后结合深度优先搜索策略进行分组. 在 DG2 中, 阈值的大小能够随着问题的改变而改变. 但是在某些极端情况下, 会造成阈值向一方倾斜, 导致分组不合理, 在某些问题上算法性能达不到预期效果. 表 4 中的实验结果也显示出 CBCC3-DG2 算法在处理完全可分离、重叠及完全不可分离的测试问题时, 性能还有待提高.

4.2.2 收敛性比较

收敛性是判定算法性能好坏的重要指标, 反映算法能否快速有效地收敛至最优解. 图 5 分别给出了各算法在 15 个测试函数中的收敛曲线. 每条收敛曲线上的 30 个点(等间距获取)都为算法运行 25 次的平均值.

从这些收敛曲线可以看出, FCA-G 算法在绝大部分测试函数上表现出更快的收敛速度, 并且收敛至较小的优化值. 特别是在部分可分离函数 ($f_4 \sim f_{11}$) 中, FCA-G 的收敛速度以及收敛值均发挥了最优

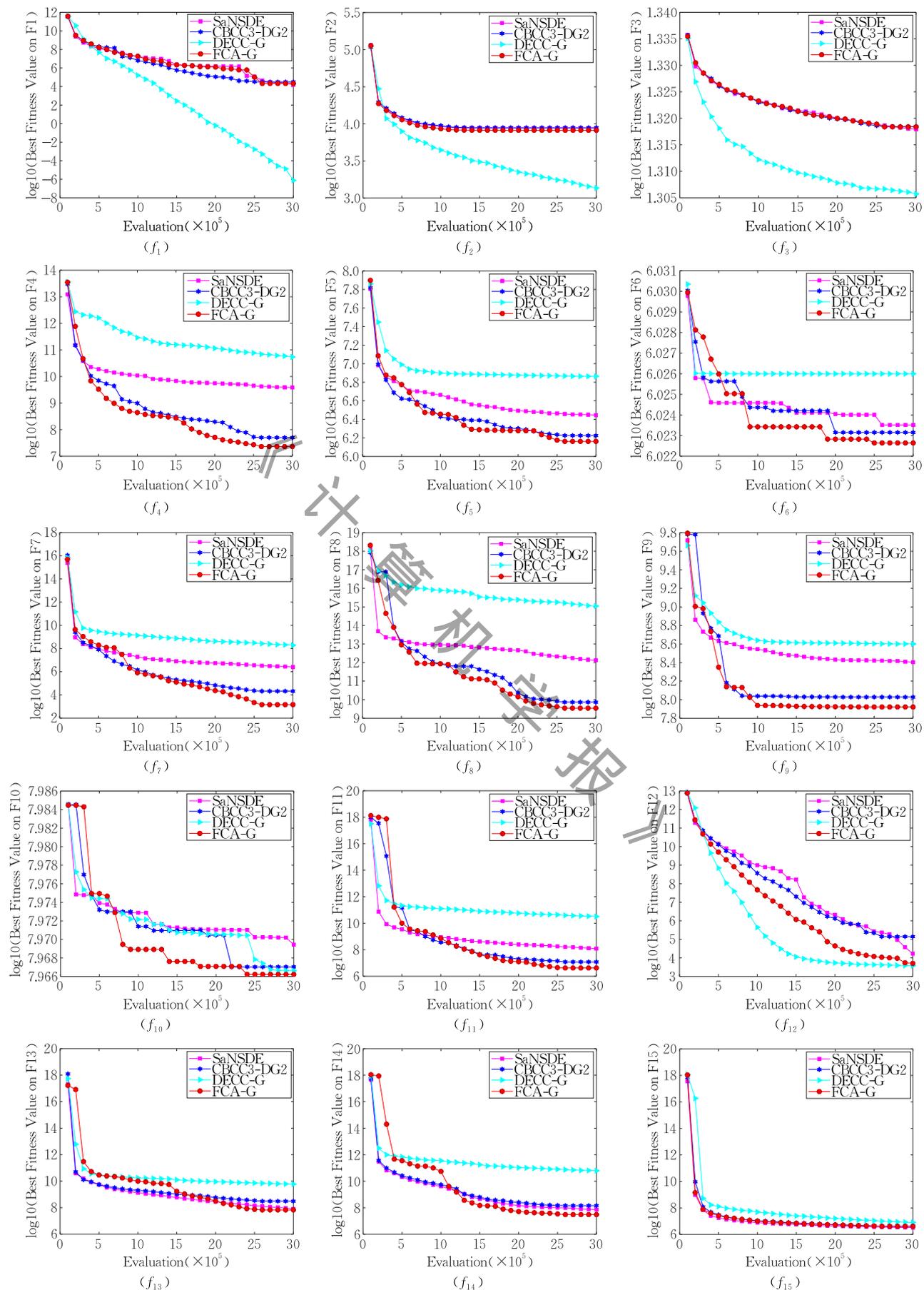


图 5 FCA-G 与 SaNSDE、DECC-G 和 CBCC3-DG2 收敛性比较

的表现力. 在重叠函数($f_{12} \sim f_{14}$)中, 每个算法都有各自的优势. DECC-G 在 f_{12} 中, 收敛速度以及收敛值均达到最优. SaNSDE 与 CBCC3-DG2 在 f_{13}, f_{14} 上最先实现收敛, 但是最终的收敛结果差强人意. 在这两个函数中, FCA-G 在进化前期没有较好的表现, 但是它在整个进化过程中一直保持着良好的收敛趋势, 并且最终取得了更优的收敛值. FCA-G、SaNSDE 和 CBCC3-DG2 算法在函数 $f_1 \sim f_3$ 和 f_{15} 中具有极其相似的收敛曲线和收敛值. 总体而言, FCA-G 在 CEC2013 LSGO 基准测试函数中的优化性能超过了其它对比算法.

对于以上结果的内在原因, 我们可进行如下总结: DECC-G 算法采用基于随机分组方式, 在完全可分离的测试问题中取得较好的效果. 但是当决策变量之间存在复杂的交互关系时, DECC-G 没有达到预期效果. FCA-G 的优异表现, 一方面要归功于其所采用的分组策略: 它保证了子组内的任意两个决策变量均是显式交互关系, 充分发挥变量间正向促进作用. 另一方面, FCA-G 所使用的 SaNSDE 优化引擎不但考虑了小步长和大步长之间的权衡, 而且能够根据进化过程中的统计学习经验对突变策略和交叉概率进行自适应调整, 进而提升寻优能力.

5 总 结

本文首次将形式概念分析(FCA)的思想引入大规模决策变量的依赖关系分析与分组学习过程, 提出基于 FCA 的大规模全局协同进化优化算法, 解决复杂的大规模全局优化问题. 首先, 基于 FCA 的分组学习框架主要基于 FCA 的外延与内涵等形式, 对决策变量进行形式文本转换, 进而根据该文本向量的特征信息进行矩阵运算, 识别变量之间的依赖关系和对其自适应的分组. 该方式以高内聚低耦合为基本原则, 保证子组内部具有较高的相关度而子组间的相关度尽可能低. 最后, 将提出的分组方法嵌入到 CC 框架中, 形成新的大规模全局优化方法(FCA-G).

在实验中, FCA-G 与 SaNSDE、CBCC3-DG2、DECC-G、SHADE-ILS 和 MOS 等算法进行了充分的对比分析. 实验结果表明, FCA-G 在处理部分可分离, 重叠以及完全不可分离大规模问题时取得了较好的优化效果. 然而, 我们也看到 FCA-G 在某些问题中(例如函数 f_1)的优化性能需要进一步提升.

因此, 在未来研究中, 我们将 FCA-G 与更多的优化策略相结合(例如基于贡献的大规模全局优化策略等), 提高算法的搜索能力与鲁棒性. 此外, 将 FCA-G 应用于实际工程问题的求解也是下一步的研究重点.

参 考 文 献

- [1] Li X, Yao X. Tackling high dimensional nonseparable optimization problems by cooperatively coevolving particle swarms//Proceedings of the 2009 IEEE Congress on Evolutionary Computation. Trondheim, Norway, 2009: 1546-1553
- [2] Liu H, Wang Y, Fan N. A hybrid deep grouping algorithm for large scale global optimization. IEEE Transactions on Evolutionary Computation, 2020, 24(6): 1112-1124
- [3] Ge H, Sun L, Tan G, et al. Cooperative hierarchical PSO with two stage variable interaction reconstruction for large scale optimization. IEEE Transactions on Cybernetics, 2017, 47(9): 2809-2823
- [4] Liu W, Zhou Y, Li B, et al. Cooperative co-evolution with soft grouping for large scale global optimization//Proceedings of the 2019 IEEE Congress on Evolutionary Computation. Wellington, New Zealand, 2019: 318-325
- [5] Fister I, Suganthan P N, et al. Artificial neural network regression as a local search heuristic for ensemble strategies in differential evolution. Nonlinear Dynamics, 2016, 84(2): 895-914
- [6] Fister I, Perc M, Kamal S M, et al. A review of chaos-based firefly algorithms. Applied Mathematics and Computation, 2015, 252: 155-165
- [7] Ali M Z, Awad N H, Suganthan P N, Reynolds R G. An adaptive multipopulation differential evolution with dynamic population reduction. IEEE Transactions on Cybernetics, 2017, 47(9): 2768-2779
- [8] Strasser S, Sheppard J, Fortier N, Goodman R. Factored evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 2017, 21(2): 281-293
- [9] Fan J C, Wang J, Han M. Cooperative coevolution for large-scale optimization based on kernel fuzzy clustering and variable trust region methods. IEEE Transactions on Fuzzy Systems, 2014, 22(4): 829-839
- [10] Cheng R, Jin Y C. A competitive swarm optimizer for large scale optimization. IEEE Transactions on Cybernetics, 2015, 45(2): 191-204
- [11] Yang Q, Chen W N, Gu T, et al. Segment-based predominant learning swarm optimizer for large-scale optimization. IEEE Transactions on Cybernetics, 2017, 47(9): 2896-2910
- [12] Zhang Y F, Chiang H D. A novel consensus-based particle swarm optimization-assisted trust-tech methodology for large-scale global optimization. IEEE Transactions on Cybernetics, 2017, 47(9): 2717-2729

- [13] Potter M A, De Jong K. A cooperative coevolutionary approach to function optimization//Proceedings of the Parallel Problem Solving from Nature. Berlin, Germany, 1994: 249-257
- [14] Yang Z, Tang K, Yao X. Large scale evolutionary optimization using cooperative coevolution. *Information Sciences*, 2008, 178(15): 2985-2999
- [15] Liu J, Tang K. Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution//Proceedings of the Intelligent Data Engineering and Automated Learning (IDEAL 2013). Berlin, Germany, 2013: 350-357
- [16] Chen W, Weise T, Yang Z, Tang K. Large-scale global optimization using cooperative coevolution with variable interaction learning//Proceedings of the Parallel Problem Solving from Nature (PPSN XI). Berlin, Germany, 2010: 300-309
- [17] Omidvar M N, Li X, Yao X. Cooperative co-evolution with delta grouping for large scale non-separable function optimization//Proceedings of the IEEE Congress on Evolutionary Computation. Barcelona, Spain, 2010: 1-8
- [18] Omidvar M N, Li X, Mei Y, Yao X. Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Transactions on Evolutionary Computation*, 2014, 18(3): 378-393
- [19] Mei Y, Omidvar M N, Li X, Yao X. A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization. *ACM Transactions on Mathematical Software*, 2016, 42(2): 13:1-13:24
- [20] Omidvar M N, Yang M, Mei Y, Li X, Yao X. DG2: A faster and more accurate differential grouping for large-scale black-box optimization. *IEEE Transactions on Evolutionary Computation*, 2017, 21(6): 929-942
- [21] Jia Y H, Zhou Y R, Lin Y, et al. A distributed cooperative co-evolutionary CMA evolution strategy for global optimization of large-scale overlapping problems. *IEEE Access*, 2019, 7: 19821-19834
- [22] Zhang X Y, Gong Y J, Lin Y, et al. Dynamic cooperative coevolution for large scale optimization. *IEEE Transactions on Evolutionary Computation*, 2019, 23(6): 935-948
- [23] Omidvar M N, Li X, Yao X. Smart use of computational resources based on contribution for cooperative co-evolutionary algorithms//Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference. Ireland, Dublin, 2011: 1115-1122
- [24] Yang M, et al. Efficient resource allocation in cooperative co-evolution for large-scale global optimization. *IEEE Transactions on Evolutionary Computation*, 2017, 21(4): 493-505
- [25] Jia Y H, et al. Distributed cooperative co-evolution with adaptive computing resource allocation for large scale optimization. *IEEE Transactions on Evolutionary Computation*, 2019, 23(2): 188-202
- [26] Sun Y, Kirley M, Halgamuge S K. A recursive decomposition method for large scale continuous optimization. *IEEE Transactions on Evolutionary Computation*, 2018, 22(5): 647-661
- [27] Sun Y, Li X, Ernst A, Omidvar M N. Decomposition for large-scale optimization problems with overlapping components //Proceedings of the 2019 IEEE Congress on Evolutionary Computation. Wellington, New Zealand, 2019: 326-333
- [28] Li L, Fang W, Wang Q, Sun J. Differential grouping with spectral clustering for large scale global optimization//Proceedings of the 2019 IEEE Congress on Evolutionary Computation. Wellington, New Zealand, 2019: 334-341
- [29] Weise T, Chiong R, Tang K. Evolutionary optimization: pitfalls and booby traps. *Journal of Computer Science & Technology*, 2012, 27(5): 907-936
- [30] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712-731
- [31] Ma X, Liu F, Qi Yu. A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. *IEEE Transactions on Evolutionary Computation*, 2016, 20(2): 275-298
- [32] Ma L, Cheng S, Shi Y. Enhancing learning efficiency of brain storm optimization via orthogonal learning design. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020, <http://dx.doi.org/10.1109/TSMC.2020.2963943>
- [33] Cheng S, Ma L, Lu H, Lei X, Shi Y. Evolutionary computation for solving search-based data analytics problems. *Artificial Intelligence Review*, 2020, <http://dx.doi.org/10.1007/s10462-020-09882-x>
- [34] Ma L, Hu K, Zhu Y, Chen H. Cooperative artificial bee colony algorithm for multi-objective RFID network planning. *Journal of Network and Computer Applications*, 2014, 42: 143-162
- [35] Ganter B. *Formal Concept Analysis: Mathematical Foundations*. New York: Springer-Verlag, 1999
- [36] Tang P, Hui S C, Fong A C M. A lattice-based approach for chemical structural retrieval. *Engineering Applications of Artificial Intelligence*, 2015, 39: 215-222
- [37] Shi J, Wan J, Yan H, Suo H. A survey of cyber-physical systems//Proceedings of the 2011 International Conference on Wireless Communications and Signal Processing. Nanjing, China, 2011: 1-6
- [38] Snásel V, Horák Z, Abraham A. Understanding social networks using formal concept analysis//Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. Sydney, Australia, 2008: 390-393
- [39] Snasel V, Horak Z, Kocibova J, et al. Analyzing social networks using FCA: Complexity aspects//Proceedings of the 2009 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. Milan, Italy,

- 2009; 38-41
- [40] Zurada J, Mazurowski M A, Ragade R, et al. Building virtual community in computational intelligence and machine learning. *IEEE Computational Intelligence Magazine*, 2009, 4(1): 43-54
- [41] Nan Du, Bin Wu, Xin Pei, et al. Community detection in large-scale social networks//*Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*. New York, USA, 2007: 101-102
- [42] Hao F, Pei Z, Yang L. Diversified top- k maximal clique detection in social Internet of Things. *Future Generation Computer Systems*, 2020, 107: 408-417
- [43] Omidvar M N, Li X, Tang K. Designing benchmark problems for large-scale continuous optimization. *Information Sciences*, 2015, 316: 419-436
- [44] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 1999, 3(4): 257-271
- [45] Hao F, Yau S, Min G, Yang L. Detecting k -balanced trusted cliques in signed social networks. *IEEE Internet Computing*, 2014, 18(2): 24-31
- [46] Li K, Du Y, Xiang D, et al. A method for building concept lattice based on matrix operation. *International Conference on Intelligent Computing*, 2007, 4682: 350-359
- [47] Yang Z, Tang K, Yao X. Self-adaptive differential evolution with neighborhood search//*Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. Hong Kong, China, 2008; 1110-1116
- [48] Li X, Tang K, Omidvar M N, et al. Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. <https://www.researchgate.net/publication/261562928>
- [49] Storn R, Price K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal Global Optimization*, 1997, 11(4): 341-359
- [50] Tang K, Li X, Suganthan P N, et al. Benchmark functions for the CEC2010 special session and competition on large-scale global optimization. *Nature Inspired Computation and Applications Laboratory*, Hefei, Anhui; Technical Report, 2009
- [51] Molina D, LaTorre A, Herrera F. SHADE with iterative local search for large-scale global optimization//*Proceedings of the 2018 IEEE Congress on Evolutionary Computation*. Rio de Janeiro, Brazil, 2018: 1-8
- [52] LaTorre A, Muelas S, Pea J. Large scale global optimization: Experimental results with MOS-based hybrid algorithms//*Proceedings of the 2013 IEEE Congress on Evolutionary Computation*. Cancun, Mexico, 2013: 2742-2749
- [53] Molina C D. Taco: Toolkit for automatic comparison of optimizers. <https://tacolab.org/>
- [54] Omidvar M N, Kazimipour B, Li X, Yao X. CBCC3—A contribution-based cooperative co-evolutionary algorithm with improved exploration/exploitation balance//*Proceedings of the 2016 IEEE Congress on Evolutionary Computation*. Vancouver, Canada, 2016: 3541-3548



MA Lian-Bo, Ph. D., professor. His current research interests include evolutionary computation, machine learning and industrial system optimization.

CHANG Feng-Rong, M. S. candidate. Her research interests include evolutionary computation and intelligent algorithms.

ZHANG Huan-Xi, B. S. candidate. His research interest is intelligent algorithms.

WANG Xing-Wei, Ph. D., professor. His research interests include computer networks and machine learning.

HUANG Min, Ph. D., professor. Her research interests include schedule optimization and logistics engineer.

HAO Fei, Ph. D., associate professor. His research interests include social computing, pervasive computing, big data analysis and processing.

Background

In this paper, we study the problem of large-scale global optimization (LSGO). It mainly faces two problems: first, the search space expands with the exponential growth of the number of decision variables. This also leads to the increasing number of local optimal solutions in the optimization process. Second, the interaction relationship between the variables crisscross, more complex, difficult to use the existing way of

grouping optimization. For small and medium-scale optimization problems, the evolutionary algorithm has achieved excellent performance in various industrial application systems. It can effectively handle complex optimization scenarios such as various nonlinearities, strong couplings, and mixed variables. However, due to the limitation of search ability, evolutionary algorithm is also difficult to obtain satisfactory results in

terms of solution quality and convergence efficiency when the scale of decision variables exceeds a certain order of magnitude. Even if it uses an improved global optimization operator strategy.

In the existing large-scale optimization method, cooperative coevolution (CC) algorithm is the most popular frameworks currently dealing with LSGO. The CC framework is an effective method to deal with LSGO problems; any decision variable can be statically or dynamically assigned to different subgroups through a specific grouping method, and complex problems can be decomposed into simple sub-problems. In the grouping process, each sub-group represents an independent sub-problem, and the optimization process of each sub-problem does not affect each other. Finally, the solutions of all sub-groups are combined to form the final optimization result. According to this, the design of variable grouping strategy has a significant impact on the final optimization performance.

In addition, there are explicit or implicit interdependencies among a large number of decision variables in the LSGO problem, which increases the difficulty of grouping. So far, there have been many new CC algorithm have been proposed and developed. They use different heuristic random grouping strategy. It is worth noting that linkage learning works well on partially separable problems, but not so well on indivisible ones. In this paper, formal concept analysis (FCA) is introduced into CC to solve the LSGO problem. The main idea is to treat the set with the same extent and intent as a subgroup through FCA technology and matrix operation to solve the grouping problem between interacting variables. Experimental results show that the proposed algorithm is effective in dealing with LSGO problems.

This research is supported by the National Natural Science Foundation of China (61773103, 61702317).

计算机学报