

引用格式: 江璞玉, 刘均, 周奇, 等. 大规模黑箱优化问题元启发式求解方法研究进展[J]. 中国舰船研究, 2021, 16(4): 1-18.

JIANG P Y, LIU J, ZHOU Q, et al. Advances in meta-heuristic methods for large-scale black-box optimization problems[J]. Chinese Journal of Ship Research, 2021, 16(4): 1-18.

大规模黑箱优化问题元启发式求解方法研究进展



扫码阅读全文

江璞玉¹, 刘均¹, 周奇², 程远胜^{*1}

1 华中科技大学 船舶与海洋工程学院, 湖北 武汉 430074

2 华中科技大学 航空航天学院, 湖北 武汉 430074

摘要:大型复杂工程装备的优化设计通常为高复杂度、高维度的优化问题, 即所谓的大规模黑箱优化问题, 其特点是目标函数和/或约束函数解析式不可知且设计变量维度很高。近年来, 大规模黑箱优化问题在各领域引起了学者们的兴趣, 而元启发式算法被认为是求解该问题的有效方法。为此, 全面总结了近年来求解该问题的元启发式算法的研究进展, 包括使用与不使用分解策略的元启发式算法, 以及处理大规模昂贵优化问题的代理模型辅助元启发式算法, 并指出了针对此问题的元启发式求解方法未来可能的研究方向。

关键词:大规模优化; 黑箱优化; 元启发式算法; 代理模型辅助优化

中图分类号: U662.9

文献标志码: A

DOI: 10.19693/j.issn.1673-3185.02248

Advances in meta-heuristic methods for large-scale black-box optimization problems

JIANG Puyu¹, LIU Jun¹, ZHOU Qi², CHENG Yuansheng^{*1}

1 School of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

2 School of Aerospace Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Abstract: The optimal design of complex engineering equipment usually faces high-complexity, high-dimensional optimization problems – the so-called "large-scale black-box optimization problems (LBOPs)" – which are characterized by unavailable mathematical expressions of objective functions and/or constraint functions, and high dimensionality of design variables. The LBOPs have attracted the interest of scholars in various fields in recent years, and meta-heuristic algorithms are considered effective methods for solving these problems. This paper comprehensively summarizes recent research progress in meta-heuristic algorithms for solving LBOPs, including meta-heuristic algorithms with and without decomposition strategies, and meta-heuristic algorithms for handling computationally expensive large-scale optimization problems. Finally, possible future research directions of meta-heuristic methods for solving LBOPs are proposed.

Key words: large-scale optimization; black-box optimization; meta-heuristic algorithms; surrogate-assisted optimization

收稿日期: 2020-12-31 修回日期: 2021-03-29 网络首发时间: 2021-07-16 09:12

作者简介: 江璞玉, 男, 1994年生, 博士生。研究方向: 进化计算, 代理模型辅助的进化计算。

E-mail: puyujiang1994@qq.com

刘均, 男, 1981年生, 博士, 副教授, 博士生导师。研究方向: 结构分析与优化, 结构抗爆抗冲击, 结构振动与控制。E-mail: hustlj@hust.edu.cn

周奇, 男, 1990年生, 博士, 副教授, 博士生导师。研究方向: 装备智能优化设计, 代理模型理论与方法。

E-mail: qizhouhust@gmail.com

程远胜, 男, 1962年生, 博士, 教授, 博士生导师。研究方向: 结构分析与轻量化设计, 结构冲击动力学与防护设计, 基于代理模型的优化方法。E-mail: yscheng@hust.edu.cn

*通信作者: 程远胜

0 引言

船舶优化设计是业界十分关注的问题之一,例如船舶水动力性能的优化^[1-2]、舱段结构优化^[3-4]、声性能优化^[5]等。船舶优化设计的目标函数或约束条件通常是由数值仿真方法(例如计算流体力学(CFD)方法、有限元方法(FEM)、边界元方法(BEM)等)计算得到,一般具有如下显著特点:一是无法采用显式解析式得到;二是计算时间较长。学界将具有前者特点的优化问题称为黑箱(black-box)优化问题,后者称为昂贵优化问题。通常,将单一目标约束优化问题定义如下:

$$\begin{aligned} \min y &= f(\mathbf{x}), \quad \mathbf{x} = (x_1, \dots, x_d) \\ \text{s.t. } g_i(\mathbf{x}) &\leq 0, \quad i = 1, \dots, n \\ h_j(\mathbf{x}) &= 0, \quad j = 1, \dots, m \end{aligned} \quad (1)$$

式中, $y = f(\mathbf{x})$ 为目标函数, $\mathbf{x} = (x_1, \dots, x_d)$ 为设计变量,下标 d 表示设计变量的维度; $g_i(\mathbf{x})$, $h_j(\mathbf{x})$ 分别为第 i 个不等式约束和第 j 个等式约束; n, m 分别表示不等式约束和等式约束的个数。

由于约束优化问题可以通过惩罚函数法(penalty function method, PFM)、拉格朗日乘子(Lagrange multiplier)法等转化成无约束优化问题,故不失一般性,本文主要讨论单一目标无约束优化问题。

对于黑箱优化问题,式(1)中的目标函数或约束函数解析式与梯度无法获得,传统的基于梯度的优化方法不再适用于黑箱优化问题的直接求解。而元启发式算法(meta-heuristic algorithm)通过对决策空间的部分子集进行采样来搜索优化解,具有不需要目标函数的梯度信息和适用性广等特点^[6],故成为了解决黑箱优化问题的有效途径。其中,代表性的元启发式算法有遗传算法(genetic algorithm, GA)^[7]、差分进化(differential evolution, DE)算法^[8]、粒子群算法(particle swarm optimization, PSO)^[9]、进化策略(evolutionary strategies, ES)^[10]等。图1所示为元启发式算法的流程图。

如果黑箱优化问题设计变量的维度很高,则称之为大规模黑箱优化问题(large-scale black-box optimization problem, LBOP),例如,大型船舶的中剖面优化、舱段结构优化。对于求解LBOP问题,若以各板列板的厚度、骨材间距和骨材型号作为设计变量,可能会有成百上千的设计变量,就目前而言,求解包括了如此多设计变量的优化问题将非常具有挑战性。鉴于黑箱优化问题的特性,元启发式算法仍然是优秀的求解算法之一。然而,对于求解LBOP问题,在低维优化问题上表

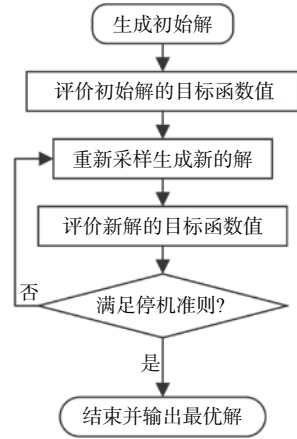


图1 元启发式算法流程图

Fig. 1 Flowchart of meta-heuristic algorithm

现优异的大部分算法的性能下降十分严重。其原因是,随着维度的增加,设计空间体积也随之呈指数增长。若设计方案的目标函数评价没有达到足够的次数,现有优化算法就很难对如此巨大的空间进行充分探索。关于设计变量的维度达到多少可定义为大规模优化问题,目前尚无定论。但对于进化计算,一般认为设计变量的维度达到1 000~5 000即可定为LBOP问题^[11]。

可见,提出高效求解LBOP问题的元启发式算法,对于提高船舶优化设计效率、缩减设计周期以及提升设计质量都具有重大意义。本文拟对目前处理LBOP问题的元启发式算法进行总结分析,提出未来可能的研究方向或重点,以便为相关领域感兴趣的学者提供参考。

1 处理大规模优化问题的方法

一般地,根据是否基于分解策略方法来处理LBOP问题,可以将元启发式算法分为两大类,表1列出了归纳汇总的相关文献,下文将分别对这两个大类算法中的代表性算法进行综述。

表1 大规模黑箱优化问题元启发式算法研究文献汇总
Table 1 Summary of research advances using meta-heuristic algorithms for LBOPs

| 是否基于分解策略 | 使用的优化框架 | 典型文献 |
|----------|----------------|-----------|
| 否 | DE算法 | [12-28] |
| | PSO算法 | [29-39] |
| | 其他及混合型算法 | [40-49] |
| 是 | 合作协同进化算法(CCEA) | [50-105] |
| | 非CCEA算法 | [106-118] |

1.1 不使用分解策略的方法

此类方法中将LBOP问题作为一个整体来处理,通常是结合局部搜索、缩减搜索空间、改进行

化算子等途径来克服 LBOP 问题引起的各种困难。具有代表性的是 MA-SW-Chains 算法^[40], 该算法曾在 IEEE CEC 2010 年年会 LBOP 问题竞赛中获奖, 性能十分优秀。MA-SW-Chains 算法也是对 MA-CMA-Chains 算法^[41]的改进, 属于文化基因算法(memetic algorithm, MA), 通常是指进化算法(evolutionary algorithm, EA)和局部搜索的混合体。MA-SW-Chains 算法在 GA 中使用了 SW (Solis-Wets) 算法^[119]作为局部搜索方法, 通过建立不同的局部搜索链, 给种群中每个个体分配一个取决于其特征的局部搜索强度, 以此提高算法对 LBOP 问题的优化性能。而不使用分解策略的元启发式算法又主要分为 EA 算法和群智能算法 (swarm intelligence algorithm, SIA) 两类, 其中又以 DE 和 PSO 为两个代表性算法。下文将主要介绍基于 DE 和 PSO 算法的元启发式算法。

1.1.1 基于 DE 的改进算法

DE 算法是 EA 算法中的典型算法之一, 其一般流程如图 2 所示。

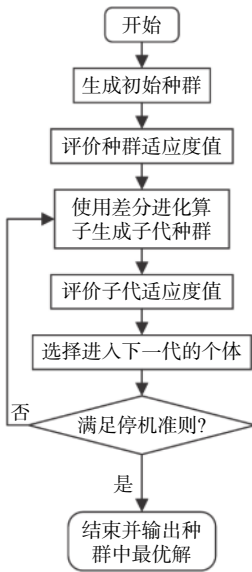


图 2 差分进化算法流程图

Fig. 2 Flowchart of differential evolution algorithm

DE 算法因其良好的全局搜索能力和易于使用的特点, 成为了被研究的热门算法。事实上, 图 1 所示流程也是绝大多数 EA 算法的流程, 只需要将 DE 算子变为其他的进化算子来产生后代即可。DE 算法的核心操作在于交叉和变异算子 (crossover and mutation operator), 大部分对 DE 的改进都着重改进交叉和变异算子, 以此增强处理 LBOP 的能力。例如, 拥有可选外部库的自适应差分进化 (adaptive differential evolution with optional external archive, Jade) 算法就是对 DE 算法的改进^[12], 其框架与基本 DE 框架大体相同。与后者相

比, Jade 算法增加了额外的样本库 A, 用以存储进化过程中被淘汰的个体。Jade 算法改进了基本 DE 的变异算子, 将其改为如下形式:

$$\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F_i \cdot (\mathbf{x}_{\text{best},g}^p - \mathbf{x}_{i,g}) + F_i \cdot (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}) \quad (2)$$

式中: $\mathbf{v}_{i,g}$ 为变异向量, 下标 i, g 分别表示个体编号和当前代数; $\mathbf{x}_{i,g}$ 为第 g 代的第 i 个个体; $\mathbf{x}_{\text{best},g}^p$ 为当前种群中前 $p\%$ 好的个体, 其中 p 表示算法需要的参数; F_i 为对应的缩放因子; $\mathbf{x}_{r1,g}, \mathbf{x}_{r2,g}$ 为第 g 代随机选择的两个个体。此外, Jade 算法的交叉概率和差分权重是根据进化过程自适应调整的。

Takahama 等^[13]提出了一种函数模态检测方法, 通过在种群中心点和最佳个体形成的直线上进行采样, 以此来判断该直线上的目标函数是单峰还是多峰, 并根据判断结果自适应地调整 DE 变异算子中缩放因子的值。Brest 等^[14]提出的自适应差分进化算法 (self-adaptive differential evolution algorithm, SaDE) 使用种群规模缩减技术, 根据随机选取的向量适应度值, 可以以一定概率改变缩放因子的符号。Wang 等^[15]提出的量化正交叉 DE 算法首先量化两个父代个体的搜索范围, 再根据量化结果调整交叉和变异算子中各参数因子的值。Zamuda 等^[16]还提出了一种使用对数正态自适应控制参数的合作协同差分进化 (differential evolution combined with cooperative co-evolution) 算法。Yang 等^[17]则在研究分析以往的自适应方法, 并结合其他方法优点的基础上, 提出了一种通用的差分进化参数自适应框架。徐广治^[18]提出了多级扰动的 DE 算法, 该算法利用随机正态分布对个体最优解进行扰动, 以此来增加种群多样性。关于改进 DE 算法进化算子的研究还有文献 [19-21, 28] 等, 限于篇幅, 这里不再赘述。

除上述着重于改进 DE 算子的研究外, 其他研究还包括注重 DE 算法与其他算法混合使用^[42-43], 以及改变原有的单一种群进化机制^[22-23]等。

1.1.2 基于 PSO 的改进算法

SIA 算法是与 EA 算法类似的另一种元启发式算法, 其中 PSO 算法是 SIA 算法的代表性算法。图 3 所示为经典的 PSO 算法流程图。

在 PSO 算法中, 粒子的位置和速度更新手段是该算法的核心。改进方向也都集中于此。Yang 等^[29]还提出了一种 PSO 算法的变体——基于层次的学习粒子群算法 (level-based learning swarm optimizer, LLSO), 该算法中种群按照其适应度值的大小进行排序, 并分成多个层级, 其中, 除最好一层的粒子直接进入下一代之外, 其他粒子的位置和速度按式(3)更新。

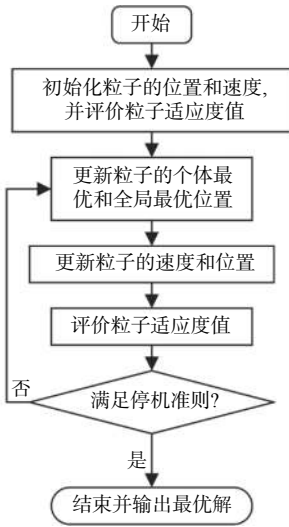


图3 粒子群算法流程

Fig. 3 Flowchart of particle swarm optimization

$$v_{i,j}^d = r_1 v_{i,j}^d + r_2 (x_{r_1, k_1}^d - x_{i,j}^d) + \phi r_3 (x_{r_2, k_2}^d - x_{i,j}^d)$$

$$x_{i,j}^d = x_{i,j}^d + v_{i,j}^d \quad (3)$$

式中: $X_{i,j} = [x_{i,j}^1, \dots, x_{i,j}^d, \dots, x_{i,j}^D]$, $V_{i,j} = [v_{i,j}^1, \dots, v_{i,j}^d, \dots, v_{i,j}^D]$ 分别为第 i 层第 j 个粒子的位置和速度, 其中 d 为优化问题的维度; r_1, r_2, r_3 为 3 个随机数; r_{l_1}, r_{l_2} 为随机选择且比当前粒子所在层数更好的两个层, 其中 r_{l_1} 优于 r_{l_2} , 若更新后的粒子位置适应度值优于原来位置, 则替换之; k_1, k_2 为随机数; ϕ 为控制参数。

此外, LLSO 算法还包括一种自适应调整层数和控制参数 ϕ 的可动态变化的基于层次学习粒子群算法 (dynamic level-based learning swarm optimizer, DLLSO)^[29]。Hsieh 等^[30] 提出基于高效种群利用策略的粒子群 (efficient population utilization strategy for particle swarm optimization, EPUS-PSO) 算法, 定义了种群管理器和解共享策略。其中, 种群管理器能够自适应地根据当前优化情况去除种群中的冗余粒子或增加新粒子, 而在解共享策略中, 将标准 PSO 算法速度更新方程的第 3 项——个体历史最优位置 (*pbest*), 按一定的概率改变为另一个粒子的个体历史最优位置。同时, 作者还提出了一种搜索范围的共享策略, 用于跳出局部最优解。

García-Nieto 等^[31] 提出了采用速度调整和重启策略的 PSO 算法, 该算法中的速度调整策略用于控制粒子在有限区域内定向运动, 而重启策略用于防止出现早熟问题, 若整个种群中粒子适应度标准差或目标函数值总体变化非常低, 则执行重启策略。Cheng 等^[32] 使用基于成对的随机竞争机制的竞争粒子群算法 (competitive particle swarm optimizer, CSO) 来求解 LBOP 问题。在算法的每

轮优化中, 从当前种群中随机选择粒子开展竞争, 获胜粒子被传递到下一种群中, 通过向获胜粒子学习, 更新失败粒子的下一个位置及其速度。

周建宏^[33] 提出了通过对搜索的子空间进行叠加来覆盖整个搜索空间的频繁覆盖策略, 并将该策略与随机漂移粒子群优化算法和具有量子行为的 PSO 算法的变体相结合。Chu 等^[34] 对高维复杂优化问题中的随机、吸收和反射 3 种边界处理方法进行了分析比较。Tian 等^[36] 基于 CSO 算法提出了一种新的两阶段粒子位置的更新策略, 用于处理大规模多目标优化问题, 其效率优于基于问题的转换算法 (problem-based transformation algorithm)、基于决策变量聚类的算法 (decision variable clustering-based algorithm)、PSO 算法和分布式估计算法 (estimation of distribution algorithm, EDA) 等。

Deng 等^[39] 根据学习者与榜样间适配度差最大化的原理, 提出了一种改进 PSO 算法, 即 RBLSO (ranking-based biased learning swarm optimizer) 算法, 该算法含排名配对学习 (ranking paired learning, RPL) 和偏心学习 (biased center learning, BCL) 两种学习策略。其中: RPL 策略是让较差粒子根据其排名从较好粒子中对等学习, 以加快收敛速度; BCL 策略是让每个粒子都向整个种群的适应度加权中心 (偏置中心) 学习, 以加强算法的探索能力。

Huang 等^[37] 提出了用于控制 PSO 算法收敛速度的策略, 当检测到算法收敛速度过快或过慢时, 自适应地调整算法的收敛速度。计算结果表明, 该策略可以帮助 PSO 算法在收敛速度与种群多样性间保持平衡。Wang 等^[38] 提出的 DGLDPSO (dynamic group learning distributed particle swarm optimization) 算法是将整个种群分为若干组, 通过“主-从”分布式模型共同进化, 该算法采用动态组学习策略 (dynamic group learning, DGL) 平衡多样性和收敛性。此外, DGLDPSO 算法被应用到了大规模云的工作流调度优化中, 作者结合资源特性还进一步提出了自适应重编号策略 (adaptive renumber strategy, ARS)。

总之, 相比于基于 DE 的方法, 基于 PSO 的方法收敛速度通常更快, 但更易陷入局部最优。

1.1.3 其他算法

除了 DE 和 PSO 分别作为 EA 及 SIA 算法的典型算法得到了大量研究外, 还有研究将其他算法改进后用于提升所研算法求解 LBOP 问题时的性能。例如, 协方差矩阵自适应进化策略 (covariance matrix adaptation evolution strategy, CMA-ES)^[120]

在处理中、低维问题时有着卓越的性能,但在处理 LBOP 问题时,随着维度的增加,协方差矩阵计算复杂度也随之迅速增加。因此, Ros 和 Hansen^[44]提出了 sep-CMA-ES 算法,该算法只计算协方差矩阵的对角元素,以降低计算复杂度,即计算复杂度仅随问题的维度线性增加,从而解决了 CMA-ES 算法在处理高维问题时的可用性。但是, Omidvar 等^[121]将 sep-CMA-ES 算法与几种 CCEA 算法进行比较后,发现当 LBOP 问题的维度增加时, sep-CMA-ES 算法的性能明显下降。为此, Li 等^[49]提出了一种 CMA-ES 算法的快速变体,通过使用一个低阶矩阵与几个向量逼近协方差矩阵,并使用其中 2 个向量生成新解,实现了计算复杂度随问题的维度线性增加,试验结果表明其优于其他 CMA-ES 优化算法的变体。

除了上述文献研究总结得到的方法外,还有大量未使用分解策略的元启发式算法。例如,多重轨迹搜索 (multiple trajectory search, MTS)^[46]、LSEDA-gl (EDA for large scalar global optimization)^[47]、基于对立的差分进化 (opposition-based differential evolution, ODE)^[24]、SOUPDE (shuffle or update parallel differential evolution)^[25]、改进社会蜘蛛算法 (improved social spider algorithm, ISSA)^[48] 等。关于不使用分解策略的方法更为详尽的总结可见文献 [122-123]。

总体上,未使用分解策略的方法可以通过以下途径提高对 LBOP 问题的搜索性能: 1) 加入局部搜索策略,例如 MA-SSW-Chains^[45]、μDSDE (micro differential evolution with a directional local search)^[26]、jDEsps (self-adaptive differential evolution algorithm with a small and varying population size)^[27] 等; 2) 改进交叉、变异、选择等进化算子,例如融合了反向学习的 PSO 算法 (GOPSO)^[35]、GaDE (generalized adaptive differential evolution)^[17] 等; 3) 混合不同算法,以取长补短,例如 MOS-CEC 2013^[42]。

1.2 使用分解策略的方法

基于分解策略求解 LBOP 问题的算法采用了“分而治之”(divide-and-conquer)的思想。首先,将 LBOP 问题分解为一系列较低维的子问题,然后,分别对子问题进行优化,最终,迭代逼近真实的最优解。这里,以最小化问题为例,若函数 $f(x_1, \dots, x_d)$ 满足以下条件:

$$\arg \min_{(x_1, \dots, x_d)} f(x_1, \dots, x_d) = \left(\arg \min_{(x_1)} f(x_1, \dots), \dots, \arg \min_{(x_d)} f(\dots, x_d) \right) \quad (4)$$

则称函数 $f(x_1, \dots, x_d)$ 是可分的^[124-125],亦即若一个函数单独优化各维度的变量得到的最优解与其他变量的取值无关,且等于该函数全局最优解在该维度上的取值,则认为该函数是可分的。根据函数可分的定义,可以得到加法可分的定义。加法可分的函数可以方便地表示许多具有模块化性质的实际工程问题^[126],其数学形式简单,有利于理论推导。

若函数 $f(x_1, \dots, x_d)$ 可以写为如式 (5) 所示的形式,则称之为部分加法可分的函数。

$$f(\mathbf{x}) = \sum_{i=1}^s f_i(\mathbf{x}_i) \quad (5)$$

式中: \mathbf{x}_i 为函数 f_i 的决策向量,且 \mathbf{x}_i 之间无相互重叠的决策变量,即 \mathbf{x}_i 互斥; $\mathbf{x} = \langle x_1, \dots, x_d \rangle$, 为全局决策变量; s 为独立子问题的个数。若所有独立的子问题都是一维的,则将函数 $f(\mathbf{x})$ 称为完全加法可分的。若独立的子问题的个数只有一个,则称 $f(\mathbf{x})$ 为完全不可分的。

1.2.1 合作协同进化算法

合作协同进化算法 (cooperative co-evolutionary algorithm, CCEA) 运用生物协同演化的思想将问题分解为一系列子问题 (即子种群),此方法在解决 LBOP 问题时表现出了高效性,是解决 LBOP 问题的主要和热门方法之一。文献 [127] 对 2018 年以前提出的 CCEA 算法进行了较为详细和全面的总结。在此基础上,本节对 2018 年至今在求解 LBOP 问题方面采用 CCEA 算法所取得的研究进展进行总结补充,如表 2 所示。

表 2 合作协同进化算法研究文献汇总
Table 2 Summary of research advances of CCEA

| 研究内容 | 典型文献 |
|----------|--------------------|
| 分组策略 | [50-73, 128] |
| 合作者选择策略 | [50-51, 61, 74-91] |
| 适应度评价策略 | [88-89, 92-100] |
| 计算资源分配策略 | [50, 101-105] |

图 4 所示为典型的 CCEA 算法的框架^[129]。

如图 4 所示, CCEA 算法将一个高维问题分解为 s 个低维子问题,并为每个子问题建立一个用于进化的对应子种群,以串行或并行方式,利用 EA 算法对每个子种群单独优化 (称为一轮优化)。由于每个子问题的决策变量仅是全部决策变量的一部分,所以在优化子问题评价某个体的适应度值时,需要从对应的合作者库中选择合作者 (图中浅蓝色的个体部分),补齐剩余的设计变量,使其形成一个完整的解以评价其适应度值。

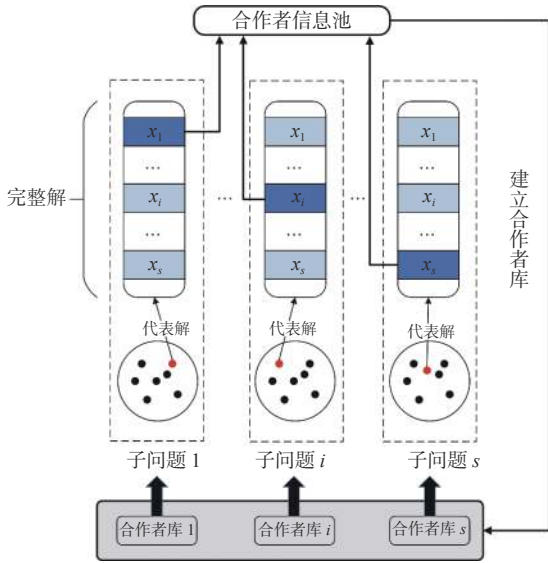


图4 合作协同进化算法框架

Fig. 4 Framework of cooperative co-evolutionary algorithm

合作者库是根据合作者信息池构建,而后者又是由各子问题中选出的代表解(图中深蓝色的个体部分)组成的,二者在优化过程中不断更新,不同子种群之间就可以以此方式实现信息交换。值得指出的是,每个子种群可选择多个代表解,并将其纳入合作者信息池。评价子种群个体适应度值时,也可以与多个合作者形成多个完整的解,来综合评价其适应度值。总之,通过不断重复上述一轮优化过程,直至获得最终的优化解。对于CCEA的研究方向主要有分组策略、合作者选择策略等,下文将分别介绍此方面的研究工作。

1) 分组策略研究。

很显然,如何分组及分组的正确与否对于算法性能有着重大的影响。若本来彼此之间无相互影响(可分)的变量被分到同一个子问题内,将无法体现“分而治之”方法的优势。若本来彼此之间存在相互影响(不可分)的变量被分到不同的子问题内,将会丢失子问题的适应度函数特性,使得合作者的改变引起子问题适应度函数特性的大幅度改变,并且算法也将收敛到纳什均衡(Nash equilibrium),而非全局最优解^[51]。第1个合作协同进化算法CCGA由Potter和De Jong^[50]提出,该算法是将一个 d 维问题分解为 d 个一维子问题,并使用GA算法轮流对子问题进行优化。这种最原始的分组方法在处理不可分问题(例如Rosenbrock和Griewank函数^[11])时的性能非常差。

关于分组策略的研究内容大致可分为动态和静态分组方法两类。其中,动态分组方法指的是原问题的分组在协同进化优化过程中会发生变化的方法。

(1) 动态分组方法。

随机分组(random grouping)是这类分解方法的代表。Yang等^[51]结合随机分组和CCEA算法提出了DECC-G算法,该算法将一个 n 维LBOP问题随机分为 k 组,每个子问题包含 n/k 维度的变量。具体而言,每个协同进化框架在循环开始前(即下一轮子问题开始优化前),将再次将包含 n 维的问题重新随机分成 k 组,以开始新一轮优化。通过每轮优化之间不断改变分组方案,以此提高不可分变量被分到同一组中的概率。此外,DECC-G算法还在每轮优化结束后,为各子问题分配一个权重,然后,分别对整个大种群的最优、最差、随机的个体优化该权重值,以得到更好的全局最优解。

学者们基于DECC-G算法提出了许多提高DECC-G性能的算法。例如,多层合作协同进化(multilevel cooperative coevolution, MLCC)^[52]使用分组方案库来代替DECC-G算法中每轮优化都重新分组的策略。该算法中,分组方案库中每个方案的概率选择取决于此方案的历史性能表现,即历史性能表现好的方案具有更大的被选中的概率。然而,Omidvar等^[53]指出,MLCC算法继承自DECC-G算法的自适应优化权重部分,对算法性能几乎无提升,而将已使用的函数计算次数用于更频繁的重新随机分组将更有效。最后,还有不少基于随机分组或改进DECC-G算法的研究^[54-56],但限于篇幅,不再赘述。

除了随机分组外,还有Ray和Yao^[57]提出的一种基于相关矩阵的分组方法。此方法在最开始的几轮优化中未使用分解方法,而是直接采用EA算法对原问题整体优化,然后计算种群中前50%个体的相关矩阵,并根据相关矩阵分组。对于相关系数大于某个阈值的设计变量,可将其分配到同一组,分组完成后,再采用正常的CCEA算法进行优化。自适应可变分区的协同进化算法(CCEA with adaptive partitioning, CCEA-AVP)^[58]是对Ray和Yao所提方法的改进,二者的区别在于,前者在协同进化的每一轮优化开始前都会利用已经计算过的个体来更新相关矩阵,并根据更新后的相关矩阵重新分组。但是,有研究也指出基于相关矩阵的方法计算资源消耗太大,且无法识别非线性相关的不可分变量^[121, 130]。

(2) 静态分组方法。

静态分组方法指的是一旦分组方案确定,例如将 n 维的问题分解为 n 个一维的子问题,或者将 n 维的问题分解为 k 个 n/k 维的问题,则在后续优化过程中分组方案不再发生变化的分组方法。

很显然,静态分组这种先入为主的分解方法效果并不理想,故许多研究者提出了不少自动分辨可分与不可分变量的算法。例如,Weicker等^[131]提出了一种判断相互影响变量的简单方法,其假设 $best$ 为当前已知最优解, new 为协同进化的优化器对第 i 维变量优化后的个体, $rand$ 为种群中随机选择的个体。通过式(6),可以生成两个新的个体 x' 和 x ,其第 j 维的值由下式给出:

$$x_j = \begin{cases} new_i, & j = i \\ best_j, & j \neq i \end{cases} \quad x'_j = \begin{cases} new_i, & j = i \\ rand_k, & j = k \\ best_j, & j \neq i, j \neq k \end{cases} \quad (6)$$

若 x' 的目标函数值 $f(x')$ 优于 x 的目标函数值 $f(x)$,则可认为第 j 个变量和第 k 个变量相互影响的概率增加了。

基于上述思想,Chen等^[59]提出了基于变量交互学习的合作协同进化(cooperative co-evolution with variable interaction learning, CCVIL)算法。该算法分为学习阶段和优化阶段,其中,在学习阶段利用与文献[131]中类似方法识别不可分变量并分组。Omidvar等^[60]则提出了Delta分组方法与基于Delta值的合作协同进化框架,其中,Delta值由连续两轮优化中的每个维度的绝对变化量计算得到,基于此,再根据对应的Delta值对决策变量进行排序,最终,按排序后的delta值对变量进行分组。然而,Omidvar等^[61]的研究也表明,若存在多个不可分离的子问题,Delta分组的性能会下降。因此,作者又提出采用差分分组(differential grouping, DG)方法从数学上推导出如下定理。

假设目标函数 $f(x)$ 为部分加法可分的函数,对于 $\forall a, b_1 \neq b_2, \delta \in \mathbf{R}, \delta \neq 0$,若下述条件成立:

$$\Delta_{\delta, x_p} [f](x) \Big|_{x_p=a, x_q=b_1} \neq \Delta_{\delta, x_p} [f](x) \Big|_{x_p=a, x_q=b_2} \quad (7)$$

则 x_p 和 x_q 是不可分的, p 和 q 为被选择用于判断的两个维度下标。其中:

$$\Delta_{\delta, x_p} [f](x) = f(\cdots, x_p + \delta, \cdots) - f(\cdots, x_p, \cdots) \quad (8)$$

上述推导的定理非常重要,因为它是DG方法的核心理论,Omidvar等^[61]证明了由该定理可以推导出使用非线性检测用于实数编码遗传算法的联动识别(linkage identification by nonlinearity check for real-coded genetic algorithms, LINC-R)算法^[128]。

虽然DG方法可大幅度提高分组的正确率,但也有不足之处:1)处理完全可分的问题时的计算资源消耗太大;2)无法对有重叠变量的子问题进行检测;3)对数值计算的舍入误差非常敏感;4)需要使用者事先定义一个阈值。

为了解决上述缺陷,许多学者开展了提高DG方法的效率和性能的研究。例如,提出了DG方法的两种变体形式——全局差分分组(global differential grouping, GDG)^[62]和扩展差分分组(extended differential grouping, XDG)^[63]。XDG方法是通过确定变量间的间接相互作用来处理Rosenbrock函数,但缺点是它继承了DG的敏感性特点,且其判断变量交互作用的方法可能会将可分离的变量视为不可分离的变量。

GDG方法是通过考虑计算误差来解决DG存在的敏感性问题,但该方法使用某个全局参数来检测所有的交互作用,使其不适合于不平衡的函数。不仅如此,GDG方法还需要通过检查所有变量对的交互作用,来解决识别子问题间存在重叠变量的函数。为此,Omidvar等^[64]又提出了DG方法的升级版DG2,在方法中引入了一个“原始交互结构矩阵”。首先,其自适应寻找阈值;然后,将原始交互结构矩阵转换为设计交互矩阵;最后,通过设计交互矩阵决定最终的分组方案。结果显示,DG2方法可以有效缓解原有方法存在的上述4个缺点。

在其他分组策略方面,刘礼文^[65]提出了一种再分组策略的算法,即将原问题分解为一系列子问题后,对维度仍较高的子问题再分组。此外,有学者还另外提出了一系列DG类型的算法(详见文献[66-69]),以进一步降低DG算法计算资源的消耗。而且,还有学者将设计变量和目标变量视为随机变量,首先利用统计模型统计分析变量和/或目标函数,再根据不同统计值对变量进行分组,这些统计值包括Pearson相关系数^[70]、互信息(mutual information)^[71]、最大信息系数(maximum information coefficient, MIC)^[72]、Kullback-Leibler散度^[73]等。

2) 合作者选择策略研究。

除了研究分组策略外,还有大量学者专注于合作者选择策略的研究。如上所述,在对原问题进行分解后,使用CCEA算法优化每个子问题时需要从其他子问题中选择合作者凑成一个完整的解来评估适应度。因此,如何从子问题中选择合适的合作者对于算法性能有着重要的影响。

第1种策略也是最简单直观的,其选择每个子问题中的最优个体作为合作者,将子问题个体与剩余子问题中的最优个体凑成完整的解来评价该个体的适应度(详见文献[50-51, 61, 74])。从研究结果来看,采用此策略处理完全可分的问题时非常有效,但对于不可分问题可能会导致算法陷入纳什均衡或局部最优^[75-76]。

第2种策略是随机选择每个子问题中的个体作为合作者,将该个体与剩余子问题中随机选择的个体凑成完整的解来评价子问题个体的适应度。采用该策略能有效保持种群的多样性且可防止算法早熟的问题,具体可见文献[77-80]。从研究结果来看,采用此策略处理外部环境改变很快的动态优化问题时的表现好于选择最优个体的策略,但随机选择合作者会降低算法的收敛速度^[81]。

第3种策略即所谓精英选择策略,其也被视为第1和第2种策略的结合体。该策略首先从每个子问题中挑选前 K 个最优个体形成合作者池,然后从合作者池中随机选取个体形成完整的解^[82]。值得指出的是,当 $K=1$ 时,此策略退化为第1种策略;当 K 等于子种群个体数时,则退化为第2种策略。其次, K 值不一定保持不变,例如,在算法初始阶段可以选取较大的 K 值以保证多样性,而后期阶段可以选取较小的 K 值以提高收敛速度。另外,对于选择单一合作者的策略,还包括类似于GA算法中的轮盘赌/锦标赛选择策略^[78,83]、固定选择合作者策略^[84-85]、邻域选择策略^[85]等,限于篇幅,不再赘述。

除了选择单一合作者外,也可选择多个合作者,与当前个体形成多个完整的解来对个体进行评估。为此,一种策略是选出剩余子问题中的所有个体作为合作者,具体详见文献[76,86-87]。在子种群个体数足够大时,此策略的优点是可保证收敛到全局最优^[87],但缺点是计算资源消耗巨大。例如,假设有 s 个子问题,每个子种群个体数为 N_p ,则在优化某一个子问题的过程中,仅仅评价一个个体的适应度就需要 N_p^{s-1} 次的真实函数计算。另一种策略是构建合作者库,例如基于帕累托支配的合作协同进化算法(pCCEA)^[88]是以非支配者库为基础选择合作者。在每代种群中,pCCEA算法用非支配者库中的所有个体来评估子种群中的每个个体,即生成 N_a 个完整解(其中, N_a 是子种群非支配者库中的个体数)。然而,pCCEA算法的缺点是,随着算法的推进, N_a 趋于无穷大。而改进后的合作协同进化算法(iCCEA)^[79]通过记录“信息量大”的合作者,即记录能够改变另一个子种群中两个个体的相对等级的个体解决了合作者数量趋于无穷大的问题。为此,文献[89-91]提出了大小固定的共享参考库,所有子种群从库中选择合作者。

3) 其他策略研究。

除了研究分组策略和合作者选择策略外,关于合作协同进化策略的研究还包括但不限于如下内容:

(1) 适应度评价策略。如上所述,选择单一合作者策略基本上都是选择与单一合作者凑成完整解的目标函数值作为适应度值,但是,当有多个合作者时,就会有多种评价适应度的方法。例如,若有多个合作者形成多个完整的解时,可选取多个完整解目标函数之中的最优值^[89]、平均值^[92]、最差值^[93]作为其适应度值,或基于帕累托支配的概念分配适应度值^[88-89,94-95]。常规进化算法中使用的适应度评价技术(例如,适应度继承^[96-97]、小生境技术^[98-100]等)也可应用于CCEA算法。

(2) 计算资源分配策略。即给每个单独优化的子问题分配计算资源,包括均匀分配^[50]、随机分配^[101]、基于对原问题贡献程度的分配^[102-105]等。

1.2.2 其他算法

CCEA算法是基于分解思想的最典型、研究最多的算法之一。此外,仍有部分同样利用了分解思想不属于CCEA的算法,例如代表性的EDA算法^[106]。EDA算法通过概率模型描述候选解的空间分布,利用有希望的解集来估计变量分布和变量相互作用,然后,根据学习到的变量分布及变量之间的相互作用,生成新候选解^[107]。最简单的EDA算法(例如紧致遗传算法(compact genetic algorithm, cGA)^[108]、基于种群的增量学习(population-based incremental learning, PBIL)^[109]等)中每个变量都是独立的,不过,这些算法不适合处理不可分的问题。

Dong等^[110]提出的采用模型复杂度控制技术的EDA算法(EDA-MCC)首先根据线性相关系数来识别交互作用较弱的变量,然后将剩余的交互作用较强的变量强行划分为若干个不重叠的子集,以减小搜索空间。EDA-MCC算法还可对所有的变量子集进行不同概率模型的演化。结果表明,在一批基准测试函数上,EDA-MCC算法的性能优于传统的EDA算法和其他一些高效的算法。Xu等^[111]进一步对EDA-MCC算法进行了改进,其采用互信息代替线性相关系数来识别交互作用较弱的变量,这样可以检测到变量间的非线性依赖关系。

Yang等^[112]结合分解策略与代理模型技术,提出了一种自我评价进化(self-evaluation evolution, SEE)算法,即将多维度问题相应地划分为一维多个子问题,通过局部搜索算子分别对每个子问题进行演化,通过训练每对子代和父代的一部分得到简单的代理模型,从而高效地实现子代解的适应度评价。最近,Yang等^[113]还采用并行计算技术以增强SEE算法,进一步发展出一种高效算法——自然并行的分而治之算法(naturally paralleled di-

vide-and-conquer, NPDC)。

在基于原问题分解的条件分布来处理不可分问题的算法中, 因子化分布算法 (factorized distribution algorithm, FDA)^[114-115] 在加法可分问题上表现较好, 但需要问题架构的先验知识, 计算成本极高。

此外, 还有一部分研究是利用专门设计的进化运算符、表征和机制, 将变量分成若干组。例如, Schaffer 和 Morishima^[116] 在个体染色体的每个基因上附加一个标志位, 用以指示交叉点, 相邻的两个交叉点间的基因被合并为同一组。在此基础上, Levenick^[117] 扩展了上述方法, 引入额外的标志位以表示选择一个位置作为交叉点的概率。

Smith 和 Fogarty^[118] 提出采用联动进化遗传算子 (linkage evolving genetic operator, LEGO) 自适应调整变量的分组。该方法通过为每个基因施加两个布尔标志来表示其在染色体上与哪一个邻居 (左邻或右邻) 相互作用, 而相互作用的邻接基因被认为是同一组的一部分。

Wu 等^[132] 提出的新混合算法则是采用现有分组算法将 LBOP 问题划分为若干个小规模问题, 采用设计的改进自适应离散扫描方法对全部搜索空间进行粗略扫描, 且着重搜索存在希望的区域。该混合搜索方法可自适应选择一维搜索方案或协方差矩阵自适应进化策略, 以分别解决可分离问题、部分 (加法) 可分问题或不可分问题的子问题。

2 代理模型辅助的大规模优化算法

在实际工程应用中, 计算目标函数通常非常耗时。例如, 对船舶冲击动力响应的仿真计算, 一次计算可能需要数小时到几十小时不等, 故也将这种目标函数计算非常耗时的优化问题称为昂贵的优化问题。目前, LBOP 问题相关算法所需目标函数计算次数仍高达 10^6 数量级, 这种计算成本是不可接受的。因此, 有学者引入代理模型来有效解决时间成本高昂的问题。对于元启发式算法而言, 代理模型几乎可以在元启发式算法的所有阶段使用, 如图 5 所示。图中, 灰色框表示代理模型与元启发式算法交互的阶段。

代理模型可用于指导初始解的生成、辅助评价解的目标函数值、指导新生成解的采样策略等。对于昂贵的优化问题, 最常用方法是采用代理模型替代昂贵的耗时计算。对于低维昂贵的优化问题, 目前的研究已相当成熟, 典型方法可分为代理模型辅助的进化算法 (surrogate-assisted evolutionary algorithm)^[133] 和贝叶斯优化框架 (Baye-

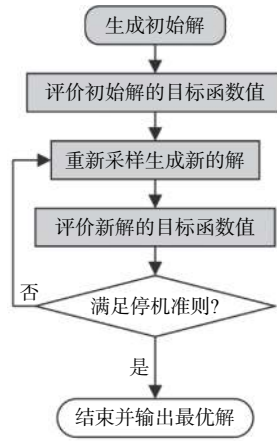


图 5 代理模型与元启发式算法交互示意图

Fig. 5 Diagram of interaction between surrogate model and meta-heuristic algorithm

sian optimization framework)^[134-135]。有关这两类算法可详见文献 [136-137]。但是, 传统上适用于低维昂贵的优化问题的方法却不能用于求解昂贵的 LBOP 问题, 其原因主要在于如下困难: 1) 随着求解问题的维度增加, 优化算法的搜索能力大幅降低, 即使采用本文所述大规模优化算法可提高算法在高维的搜索能力, 但仍难以找到真实的全局最优解; 2) 随着求解问题的维度增加, 代理模型的精度会迅速下降, 构建代理模型所需样本点数也会迅速增加。不仅如此, 代理模型本身的计算复杂度也会随着维度的增加而增加。

目前, 关于代理模型辅助 LBOP 问题的研究并不多见^[138-140]。与上文类似, 这些研究同样可以分为使用和不使用分解策略两大类, 表 3 按此分类给出了求解 LBOP 问题时使用代理模型辅助元启发式算法的典型文献。表中, RBF (radial basis function) 为径向基模型, GP (Gaussian process) 为高斯过程模型。

表 3 代理模型辅助元启发式算法的文献分类

Table 3 Summary of surrogate model assisted meta-heuristic algorithms

| 是否基于分解策略 | 典型文献 | |
|----------|-----------|----------------------|
| | RBF模型 | GP模型 |
| 否 | [141-146] | [138, 140, 145, 147] |
| 是 | [148-151] | [151] |

2.1 不使用分解策略的方法

原则上, 对于不使用分解策略的方法求解低维的昂贵优化问题, 直接使用代理模型拟合目标函数值 (也称适应度近似) 的方法仍可以使用。然而, 由于代理模型在高维空间中的精度会迅速下降, 以及建模成本显著上升, 采用适应度近似

的方法求解 LBOP 问题的效果会严重下降,此时,可以考虑使用其他代理模型。

例如,根据父代适应度拟合子代的适应度值(也称适应度继承)和利用不同精度代理模型迁移个体间的适应度(也称适应度迁移);或使用代理模型指导交叉变异操作、初始种群生成等。另外,还可以采取降维的方式,结合整体与局部代理模型等各种途径来缓解“维数灾难”的问题。其中,部分研究采用的是基于高维空间到低维空间映射的降维方法(例如文献[138,140]中使用的 Sammon 映射方法)。然而,降维也意味着会丢失一定数量的设计变量对目标函数的影响,研究证明此方法仅适用于中等规模的优化问题(50~100个变量)^[138-140]。

为了获取全局最优解,Sun 等^[141]提出了代理模型辅助合作的粒子群优化(surrogate-assisted cooperative swarm optimization, SA-COSO)算法,即通过代理模型辅助的 PSO 算法与代理模型辅助社会学习的粒子群优化算法(social learning-based PSO, SL-PSO)合作来搜索全局最优解。PSO 与 SL-PSO 之间的合作包含两个方面:一是其共享由真实函数评估过的有希望的解;二是 SL-PSO 算法侧重于全局搜索,PSO 算法侧重于局部搜索。Yu 等^[142]提出一种类似于 SA-COSO 的算法以提高 SA-COSO 算法的性能,Sun 等^[139]使用基于适应度近似的竞争群优化算法(competitive swarm optimizer, CSO)处理 500 个决策变量的问题,算法采用适应度继承策略^[55]替代部分昂贵的适应度评估,该适应度继承的方法也可以被视为局部代理模型。研究表明,相比未采用任何适应度近似策略的 CSO 算法,采用代理模型辅助的 CSO 算法对 500 个变量的基准测试函数计算得到的结果更好,或者至少有竞争力。

Werth 等^[147]提出的窗口式优化方法是根据变量相互作用关系对问题进行初步分组,算法每次迭代都在一个滑动窗口(该窗口仅包含所有设计变量的一个子集)内完成。初步研究结果表明,运用这种窗口式优化方法优化 5 个高达 1 000 维度的问题是有效的。Fu 等^[143]提出的采用随机特征选择技术的代理模型辅助进化算法(surrogate-assisted evolutionary algorithm with random feature selection, SAEA-RFS)则是利用随机特征选择技术形成的若干子问题进行序贯优化,来优化 LBOP 问题。

在降低代理模型建模成本方面,陈祺东^[144]提出的代理模型辅助量子粒子群算法(surrogate-assisted quantum-behaved particle swarm optimization,

SAQPSO),也称 SAQPSO-WS 算法,其基于曼哈顿距离(Manhattan distance)方法在优化过程中逐渐丢弃目标函数值不佳的样本点,以此降低代理模型的建模成本。

Wang 等^[145]提出的全局和局部代理模型辅助差分进化算法(evolutionary sampling assisted optimization, ESAO)可以交替进行全局和局部搜索,将 RBF 模型作为全局代理模型,而将 RBF 模型与 Kriging 模型作为局部搜索模型。数值实验表明,RBF 比 Kriging 更适合作为局部代理模型。

Dong 等^[146]提出的代理模型辅助灰狼优化(surrogate-assisted grey wolf optimization, SAGWO)算法在初始阶段基于样本识别出原始狼群和狼首,用于拟合高维空间,在搜索阶段则利用灰狼优化全局搜索,结合多起点局部搜索策略在重点区域局部搜索。SAGWO 算法根据从 RBF 模型中获取的知识在每次循环内辅助生成新的狼首,并根据狼首所在位置改变狼群位置,从而达到平衡开发和探索的目的。

综上所述,目前采用不使用分解策略的算法的研究一般都仅局限于约 200 维度的高维优化问题,而对于上千维度的 LBOP 问题则几乎没有涉及。

2.2 使用分解策略的方法

对于使用分解策略的方法,很自然地会想到能够缓解所谓“维数灾难”问题的代理模型与 CCEA 算法结合的算法。CCEA 算法的特点是,首先将 LBOP 问题分解为一系列低维子问题,有效降低训练代理模型时的维度(也即避免代理模型直接拟合原有高维问题),然后再拟合低维子问题的适应度函数。尽管针对 LBOP 问题的算法研究促进发展出了一些有效的 CCEA 算法,但却鲜有关于 CCEA 算法与代理模型结合的研究,特别是涉及 LBOP 问题时。目前,两者相结合的算法研究仍处于比较初级的阶段,也就是直接在子问题优化过程中使用代理模型来替代原有的适应度函数计算。

Ren 等^[148]提出的 RBF 模型、基于成功历史的自适应差分进化算法(success-history based adaptive differential evolution, SHADE)和代理模型辅助的合作协同进化算法(surrogate model assisted cooperative coevolution, SACC)三者结合的混合算法(RBF-SHADE-SACC)中,RBF 模型作为各子问题的代理模型,SHADE 算法则为各子问题的优化器。对子问题优化过程中,只有代理模型预测值较优的个体才会被用于真实目标函数的计算,其他个体则直接使用 RBF 模型进行适应度计算。

值得指出的是, RBF-SHADE-SACC 算法默认原问题的理想分组情况已知, 但这在实际应用中很难实现。

在 Blanchard 等^[149]提出的 SACC-EAM(surrogate-assisted cooperative co-evolutionary algorithm of Minamo)算法中, 将 RBF 作为代理模型, Jade 算法作为子问题的优化器, 使用随机分组策略。该算法在使用 Jade 算法优化子问题的过程中, RBF 被用于替代耗时的真实函数计算, 最优解的评估采用的是真实函数, 并将其加入到代理模型样本点库。其后, Blanchard 等^[150]又提出了改进 SACC-EAM 的 SACC-EAM-II 算法, 通过使用递归差分分组(recursive differential grouping, RDG)策略代替原有的分组策略, 以达到提高算法性能的目的。

De Falcon 等^[151]提出的基于随机分组策略的 SACC 算法框架将 Jade 算法作为优化器, 在优化子问题的过程中可以使用全局代理模型或局部代理模型。其研究主要在于各参数对 SACC 框架的影响, 结果表明: 在大多数问题中, 降低子问题的维度(2~4 维)能够显著提升收敛性, 每个子问题的最佳进化代数会随着目标函数的不同而不同; 其次, 若由 GP、RBF、支持向量机回归(support vector regression, SVR)作为全局代理模型, 而二次多项式近似(quadratic polynomial approximation, QPA)作为局部代理模型时, 总体上性能表现最好的代理模型是 GP 和 RBF; 在大多数情况下, SVR 模型早期收敛速度更快, 并且对于多模态函数, SACC 算法性能提升不大。

综上所述, 对于不使用分解策略的算法, 由于“维数灾难”问题不能再像低维问题一样采用代理模型直接替换适应度评估, 而是需要采取降维、局部搜索等特殊的改进方法。由结果可见, 这种方法在应用于解决中高维问题(500 维以下)时的效果较好, 超过 500 维度的 LBOP 问题得到的优化解仍不够理想; 采用使用分解策略的算法由于分解机制可以在分解后的低维子问题中直接采用代理模型替换子问题的适应度计算, 因此其研究一般都涉及到了约 1 000 维度的 LBOP 问题。然而, 这种算法只是在子问题中机械地套用了低维问题的方法, 而未考虑分解策略的特点。虽然相比于不使用代理模型的算法这种方法可节省计算资源, 但是针对 LBOP 问题获得的优化解与真实的优化解仍然存在较大差距。

鉴此, 代理模型辅助大规模元启发式算法仍然具有很大的改进空间, 故有比较大的研究价值。

3 结论与展望

近年来, LBOP 问题引起了各领域研究者的广泛兴趣, 本文总结了目前有关 LBOP 问题的元启发式算法的研究进展。总体上, 用于求解 LBOP 问题的元启发式算法可以按照是否使用了分解策略进行分类。不使用分解策略的算法需要通过添加局部搜索、混合不同算法等策略来提高算法在求解 LBOP 问题时的性能; 而使用分解策略的算法则主要考虑分解策略带给算法的新特性, 从这些新特性着手提高算法效率, 例如, 提出高效的分组策略、资源分配策略等。在算法性能方面, 这两类算法无明显的优劣之分。

对于昂贵的 LBOP 问题, 使用代理模型仍然是绝大部分研究的首选方案。由于 LBOP 问题本身的复杂度, 无论算法是否使用了分解策略, 运用代理模型辅助元启发式算法来求解 LBOP 问题仍比较困难, 故有较大的提升空间。总之, 元启发式算法在求解 LBOP 问题时还将面临如下主要挑战:

1) 设计空间体积随着维度的增加呈现指数增加的趋势。

2) 高维空间中目标函数的特性变得复杂, 例如从单模态函数变为多模态函数。

3) 优化算法的搜索能力在高维空间中被削弱。

4) 优化成本高昂。

通过大量的文献分析和归纳总结, 以及基于对实际工程应用的需求思考, 本文认为 LBOP 问题的未来研究方向可以概括为:

1) 进一步提升解决 LBOP 问题的算法性能。目前, 对于 LBOP 问题现有算法仍需要大量的函数计算次数, 并且得到的优化解与真实的优化解之间仍存在较大差距。

2) 求解大规模多目标优化及大规模约束优化问题。目前, 对于无约束单一目标的 LBOP 问题仍不能得到理想的解决方案, 因此解决复杂的大规模约束优化问题更加困难。

3) 昂贵的 LBOP 问题的研究。与目前研究主要使用廉价的测试函数不同, 实际工程应用中的 LBOP 问题通常昂贵, 现有算法所需要的目标函数计算次数仍然巨大。代理模型辅助的元启发式算法似乎是比较有希望的解决方案。

4) 大规模优化算法在实际工程问题中的应用研究。在开展研究时, 可以结合设计对象的专业先验知识指导变量分组, 以提高优化效率。例如: 对于大型船舶复杂横剖面优化设计问题, 可

以将上层建筑的设计变量分为一组,主船体设计变量根据垂向高度分成若干个组;对于船舶舱段优化设计问题,可以根据板架分组(例如甲板、舷侧、船底,平台,纵舱壁等);对于船舶的艏部型线的优化问题,可以按照附体型线(例如球鼻艏型线、艏部主船体型线)分组。

参考文献:

- [1] 卢激. 双体滑翔艇水动力性能优化设计及预报 [D]. 哈尔滨: 哈尔滨工程大学, 2013.
LU W. The optimization design and prediction of hydrodynamic performance of catamaran planing crafts[D]. Harbin: Harbin Engineering University, 2013 (in Chinese).
- [2] 毕晓君, 王朝. 基于 MOEA/D 的船舶水动力性能优化 [J]. 哈尔滨工程大学学报, 2018, 39(10): 1681–1687, 1694.
BI X J, WANG C. Ship hydrodynamic performance optimization based on MOEA/D[J]. Journal of Harbin Engineering University, 2018, 39(10): 1681–1687, 1694 (in Chinese).
- [3] 常琦. 基于描述性计算和直接计算方法的油船结构优化设计 [D]. 哈尔滨: 哈尔滨工程大学, 2019.
CHANG Q. Structural optimization for oil tankers based on descriptive and direct calculation[D]. Harbin: Harbin Engineering University, 2019 (in Chinese).
- [4] 孟松. 基于改进粒子群算法的油船结构优化研究 [D]. 大连: 大连海事大学, 2019.
MENG S. Oil tanker structure optimization research based on improved particle swarm optimization algorithm[D]. Dalian: Dalian Maritime University, 2019 (in Chinese).
- [5] 宋翔, 余培汛, 白俊强, 等. 基于 Hanson 噪声模型的螺旋桨气动与噪声优化设计 [J]. 西北工业大学学报, 2020, 38(4): 685–694.
SONG X, YU P X, BAI J Q, et al. Aerodynamic and aeroacoustic optimization of propeller based on Hanson noise model[J]. Journal of Northwestern Polytechnical University, 2020, 38(4): 685–694 (in Chinese).
- [6] BLUM C, ROLI A. Metaheuristics in combinatorial optimization: overview and conceptual comparison[J]. ACM Computing Surveys, 2003, 35(3): 268–308.
- [7] HOLLAND J H. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence[M]. Cambridge, MA: MIT Press, 1992.
- [8] 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.
- [9] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory[C]//MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Nagoya, Japan: IEEE, 1995: 39–43.
- [10] RECHENBERG I. Evolutionsstrategien[M]//SCHNEIDER B, RANFT U. Simulations method en in Der Medizin Und Biologie. Berlin, Heidelberg: Springer, 1978: 83–114.
- [11] TANG K, LI X D, SUGANTHAN P N, et al. Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization[EB/OL]. (2010-01-08)[2021-03-29]Hefei, China: Nature Inspired Computation and Applications Laboratory, USTC, 2010. <https://titan.csit.rmit.edu.au/~e46507/publications/lsgo-cec10.pdf>.
- [12] ZHANG J Q, SANDERSON A C. Jade: Adaptive differential evolution with optional external archive[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 945–958.
- [13] TAKAHAMA T, SAKAI S. Large scale optimization by differential evolution with landscape modality detection and a diversity archive[C]//2012 IEEE Congress on Evolutionary Computation. Brisbane, QLD, Australia: IEEE, 2012: 1–8.
- [14] BREST J, ZAMUDA A, BOSKOVIC B, et al. High-dimensional real-parameter optimization using self-adaptive differential evolution algorithm with population size reduction[C]//2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). Hong Kong, China: IEEE, 2008: 2032–2039.
- [15] WANG Y, CAI Z X, ZHANG Q F. Enhancing the search ability of differential evolution through orthogonal crossover[J]. Information Sciences, 2012, 185(1): 153–177.
- [16] ZAMUDA A, BREST J, BOSKOVIC B, et al. Large scale global optimization using differential evolution with self-adaptation and cooperative co-evolution[C]//2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). Hong Kong, China: IEEE, 2008: 3718–3725.
- [17] YANG Z Y, TANG K, YAO X. Scalability of generalized adaptive differential evolution for large-scale continuous optimization[J]. Soft Computing, 2011, 15(11): 2141–2155.
- [18] 徐广治. 非线性群智能优化及其应用研究 [D]. 北京: 北京邮电大学, 2019.
XU G Z. Research on nonlinear swarm intelligence optimization and application[D]. Beijing: University of Posts and Telecommunications, 2019 (in Chinese).
- [19] GARCÍA-MARTÍNEZ C, RODRÍGUEZ F J, LOZANO M. Role differentiation and malleable mating for differential evolution: An analysis on large-scale optim-

- isation[J]. *Soft Computing*, 2011, 15(11): 2109–2126.
- [20] ZHAO S Z, SUGANTHAN P N, DAS S. Self-adaptive differential evolution with multi-trajectory search for large-scale optimization[J]. *Soft Computing*, 2011, 15(11): 2175–2185.
- [21] YOU X M. Differential evolution with a new mutation operator for solving high dimensional continuous optimization problems[J]. *Journal of Computational Information Systems*, 2010, 6(9): 3033–3039.
- [22] GOMEZ J, GARCIA A, SILVA C. COFRE: A fuzzy rule coevolutionary approach for multiclass classification problems[C]//2005 IEEE Congress on Evolutionary Computation. Edinburgh, UK: IEEE, 2005: 1637–1644.
- [23] PIOTROWSKI A P, NAPIORKOWSKI J J, KICZKO A. Differential evolution algorithm with separated groups for multi-dimensional optimization problems[J]. *European Journal of Operational Research*, 2012, 216(1): 33–46.
- [24] RAHNAMEYAN S, TIZHOOSH H R, SALAMA M M A. Opposition-based differential evolution[J]. *IEEE Transactions on Evolutionary Computation*, 2008, 12(1): 64–79.
- [25] WEBER M, NERI F, TIRRONEN V. Shuffle or update parallel differential evolution for large-scale optimization[J]. *Soft Computing*, 2011, 15(11): 2089–2107.
- [26] YILDIZ Y E, TOPAL A O. Large scale continuous global optimization based on micro differential evolution with local directional search[J]. *Information Sciences*, 2019, 477: 533–544.
- [27] BREST J, BOŠKOVIĆ B, ZAMUDA A, et al. Self-adaptive differential evolution algorithm with a small and varying population size[C]//2012 IEEE Congress on Evolutionary Computation. Brisbane, QLD, Australia: IEEE, 2012, .
- [28] HIBA H, IBRAHIM A, RAHNAMEYAN S. Large-scale optimization using center-based differential evolution with dynamic mutation scheme[C]//2019 IEEE Congress on Evolutionary Computation (CEC). Wellington, New Zealand: IEEE, 2019: 3189–3196.
- [29] 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*, 2018, 22(4): 578–594.
- [30] HSIEH S T, SUN T Y, LIU C C, et al. Solving large scale global optimization using improved particle swarm optimizer[C]//2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). Hong Kong, China: IEEE, 2008: 1777–1784.
- [31] GARCÍA-NIETO J, ALBA E. Restart particle swarm optimization with velocity modulation: A scalability test[J]. *Soft Computing*, 2011, 15(11): 2221–2232.
- [32] CHENG R, JIN Y C. A competitive swarm optimizer for large scale optimization[J]. *IEEE Transactions on Cybernetics*, 2015, 45(2): 191–204.
- [33] 周建宏. 面向大规模全局优化问题的元启发式算法研究[D]. 无锡: 江南大学, 2017.
ZHOU J H. Study of meta-heuristic algorithm for large scale global optimization problems[D]. Wuxi: Jiangnan University, 2017 (in Chinese).
- [34] CHU W, GAO X G, SOROOSHIAN S. Handling boundary constraints for particle swarm optimization in high-dimensional search space[J]. *Information Sciences*, 2011, 181(20): 4569–4581.
- [35] WANG H, WU Z J, RAHNAMEYAN S, et al. Enhancing particle swarm optimization using generalized opposition-based learning[J]. *Information Sciences*, 2011, 181(20): 4699–4714.
- [36] TIAN Y, ZHENG X T, ZHANG X Y, et al. Efficient large-scale multiobjective optimization based on a competitive swarm optimizer[J]. *IEEE Transactions on Cybernetics*, 2020, 50(8): 3696–3708.
- [37] HUANG H, LV L, YE S J, et al. Particle swarm optimization with convergence speed controller for large-scale numerical optimization[J]. *Soft Computing*, 2019, 23(12): 4421–4437.
- [38] WANG Z J, ZHAN Z H, YU W J, et al. Dynamic group learning distributed particle swarm optimization for large-scale optimization and its application in cloud workflow scheduling[J]. *IEEE Transactions on Cybernetics*, 2020, 50(6): 2715–2729.
- [39] DENG H B, PENG L Z, ZHANG H B, et al. Ranking-based biased learning swarm optimizer for large-scale optimization[J]. *Information Sciences*, 2019, 493: 120–137.
- [40] MOLINA D, LOZANO M, HERRERA F. MA-SW-chains: Memetic algorithm based on local search chains for large scale continuous global optimization[C]//IEEE Congress on Evolutionary Computation. Barcelona, Spain: IEEE, 2010: 1–8.
- [41] MOLINA D, LOZANO M, GARCÍA-MARTÍNEZ C, et al. Memetic algorithms for continuous optimisation based on local search chains[J]. *Evolutionary Computation*, 2010, 18(1): 27–63.
- [42] LATORRE A, MUELAS S, PEÑA J M. Large scale global optimization: Experimental results with mos-based hybrid algorithms[C]//2013 IEEE Congress on Evolutionary Computation (CEC). Cancun, Mexico: IEEE, 2013: 2742–2749.
- [43] BOLUFÉ-RÖHLER A, CHEN S. Extending minimum population search towards large scale global optimization[C]//2014 IEEE Congress on Evolutionary Computation

- tation (CEC). Beijing, China: IEEE, 2014: 845–852.
- [44] ROS R, HANSEN N. A simple modification in CMA-ES achieving linear time and space complexity[C]//Parallel Problem Solving from Nature-PPSN X. Berlin, Heidelberg: Springer, 2008, 5199: 296.
- [45] MOLINA D, LOZANO M, SÁNCHEZ A M, et al. Memetic algorithms based on local search chains for large scale continuous optimisation problems: Ma-ssw-chains[J]. *Soft Computing*, 2011, 15(11): 2201–2220.
- [46] TSENG L Y, CHEN C. Multiple trajectory search for large scale global optimization[C]//2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). Hong Kong, China: IEEE, 2008.
- [47] WANG Y, LI B. A restart univariate estimation of distribution algorithm: Sampling under mixed gaussian and levy probability distribution[C]//2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). Hong Kong, China: IEEE, 2008: 3917–3924.
- [48] BAŞ E, ÜLKER E. Improved social spider algorithm for large scale optimization[J]. *Artificial Intelligence Review*, 2020.
- [49] LI Z H, ZHANG Q F, LIN X, et al. Fast covariance matrix adaptation for large-scale black-box optimization[J]. *IEEE Transactions on Cybernetics*, 2020, 50(5): 2073–2083.
- [50] POTTER M A, DE JONG K A. A cooperative coevolutionary approach to function optimization[M]//DAVIDOR Y, SCHWEFEL H P, MÄNNER R. International Conference on Evolutionary Computation the Third Conference on Parallel Problem Solving from Nature. Jerusalem, Israel: Springer, 1994: 249–257.
- [51] YANG Z Y, TANG K, YAO X. Large scale evolutionary optimization using cooperative coevolution[J]. *Information Sciences*, 2008, 178(15): 2985–2999.
- [52] 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). Hong Kong, China: IEEE, 2008.
- [53] OMIDVAR M N, LI X D, YANG Z Y, et al. Cooperative co-evolution for large scale optimization through more frequent random grouping[C]//IEEE Congress on Evolutionary Computation (CEC). Barcelona, Spain: IEEE, 2010.
- [54] LI X D, YAO X. Tackling high dimensional nonseparable optimization problems by cooperatively coevolving particle swarms[C]//2009 IEEE Congress on Evolutionary Computation. Trondheim, Norway: IEEE, 2009.
- [55] YANG Z Y, ZHANG J Q, TANG K, et al. An adaptive coevolutionary differential evolution algorithm for large-scale optimization[C]//2009 IEEE Congress on Evolutionary Computation. Trondheim, Norway: IEEE, 2009.
- [56] REN Y F, WU Y. An efficient algorithm for high-dimensional function optimization[J]. *Soft Computing*, 2013, 17(6): 995–1004.
- [57] RAY T, YAO X. A cooperative coevolutionary algorithm with correlation based adaptive variable partitioning[C]//2009 IEEE Congress on Evolutionary Computation. Trondheim, Norway: IEEE, 2009.
- [58] SINGH H K, RAY T. Divide and conquer in coevolution: a difficult balancing act[M]//Agent-Based Evolutionary Search. Berlin, Heidelberg, Germany: Springer, 2010: 117–138.
- [59] CHEN W X, WEISE T, YANG Z Y, et al. Large-scale global optimization using cooperative coevolution with variable interaction learning[M]//SCHAEFER R, COTTA C, KOŁODZIEJ J, et al. PPSN'10: Proceedings of the 11th International Conference on Parallel Problem Solving from Nature: Part II. Berlin, Heidelberg: Springer, 2010, 6239: 300–309.
- [60] OMIDVAR M N, LI X D, YAO X. Cooperative coevolution with delta grouping for large scale non-separable function optimization[C]//IEEE Congress on Evolutionary Computation (CEC). Barcelona, Spain: IEEE, 2010.
- [61] OMIDVAR M N, LI X D, MEI Y, et al. Cooperative co-evolution with differential grouping for large scale optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(3): 378–393.
- [62] MEI Y, OMIDVAR M N, LI X D, et al. A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization[J]. *ACM Transactions on Mathematical Software*, 2016, 42(2): 13.
- [63] SUN Y, KIRLEY M, HALGAMUGE S K. Extended differential grouping for large scale global optimization with direct and indirect variable interactions[C]//GECCO'15: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation. New York, NY, USA: ACM, 2015: 313–320.
- [64] 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.
- [65] 刘礼文. 求解大规模全局优化问题的高效算法研究[D]. 西安: 西安电子科技大学, 2019.
- LIU L W. Research on efficient algorithms for solving large scale global optimization problems[D]. Xi'an: Xidian University, 2019 (in Chinese).
- [66] SUN Y, KIRLEY M, HALGAMUGE S K. A recursive decomposition method for large scale continuous

- optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2018, 22(5): 647–661.
- [67] SUN Y, LI X D, ERNST A, et al. Decomposition for large-scale optimization problems with overlapping components[C]//2019 IEEE Congress on Evolutionary Computation (CEC). Wellington, New Zealand: IEEE, 2019: 326–333.
- [68] SUN Y, OMIDVAR M N, KIRLEY M, et al. Adaptive threshold parameter estimation with recursive differential grouping for problem decomposition[C]//Proceedings of the Genetic and Evolutionary Computation Conference. New York, NY, USA: ACM, 2018: 889–896.
- [69] YANG M, ZHOU A M, LI C H, et al. An efficient recursive differential grouping for large-scale continuous problems[J]. *IEEE Transactions on Evolutionary Computation*, 2021, 25(1): 159–171.
- [70] TSUTSUI S, GOLDBERG D E. Simplex crossover and linkage identification: Single-stage evolution vs. Multi-stage evolution[C]//Proceedings of the 2002 Congress on Evolutionary Computation CEC'02 (Cat. No. 02TH8600). Honolulu, HI, USA: IEEE, 2002: 974–979.
- [71] SUN J J, DONG H B. Cooperative co-evolution with correlation identification grouping for large scale function optimization[C]//2013 IEEE Third International Conference on Information Science and Technology (ICIST). Yangzhou, China: IEEE, 2013: 889–893.
- [72] SUN Y, KIRLEY M, HALGAMUGE S K. Quantifying variable interactions in continuous optimization problems[J]. *IEEE Transactions on Evolutionary Computation*, 2017, 21(2): 249–264.
- [73] YU T L, GOLDBERG D E, SASTRY K, et al. Dependency structure matrix, genetic algorithms, and effective recombination[J]. *Evolutionary Computation*, 2009, 17(4): 595–626.
- [74] LI X D, YAO X. Cooperatively coevolving particle swarms for large scale optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2012, 16(2): 210–224.
- [75] LICHBACH M I. *The Cooperator's Dilemma*[M]. Ann Arbor: University of Michigan Press, 1996.
- [76] PANAIT L, LUKE S. Selecting informative actions improves cooperative multiagent learning[C]//Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems. New York, NY, USA: ACM, 2006: 760–766.
- [77] LUKE S, SULLIVAN K, ABIDI F. Large scale empirical analysis of cooperative coevolution[C]//Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation. New York, NY, USA: ACM, 2011: 151–152.
- [78] DE OLIVEIRA F B, ENAYATIFAR R, SADAEL H J, et al. A cooperative coevolutionary algorithm for the multi-depot vehicle routing problem[J]. *Expert Systems with Applications*, 2016, 43: 117–130.
- [79] PANAIT L, LUKE S, HARRISON J F. Archive-based cooperative coevolutionary algorithms[C]//GECCO '06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation. New York, NY, USA: ACM, 2006: 345–352.
- [80] TANK C, YANG Y J, GOH C K. A distributed cooperative coevolutionary algorithm for multiobjective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2006, 10(5): 527–549.
- [81] AU C K, LEUNG H F. Investigating collaboration methods of random immigrant scheme in cooperative coevolution[C]//2009 IEEE Congress on Evolutionary Computation. Trondheim, Norway: IEEE, 2009: 2700–2707.
- [82] GLORIEUX E, SVENSSON B, DANIELSSON F, et al. Improved constructive cooperative coevolutionary differential evolution for large-scale optimisation[C]//2015 IEEE Symposium Series on Computational Intelligence. Cape Town, South Africa: IEEE, 2015: 1703–1710.
- [83] SON Y S, BALDICK R. Hybrid coevolutionary programming for nash equilibrium search in games with local optima[J]. *IEEE Transactions on Evolutionary Computation*, 2004, 8(4): 305–315.
- [84] SEREDYNSKI F. Loosely coupled distributed genetic algorithms[M]//DAVIDOR Y, SCHWEFEL H P, MÄNNER R. International Conference on Evolutionary Computation the Third Conference on Parallel Problem Solving from Nature Jerusalem. Jerusalem, Israel: Springer, 1994: 514–523.
- [85] MA X L, LIU F, QI Y T, et al. A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables[J]. *IEEE Transactions on Evolutionary Computation*, 2016, 20(2): 275–298.
- [86] PANAIT L, LUKE S. A comparative study of two competitive fitness functions[C]// Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002). CA: Morgan Kaufmann, 2002: 503–511.
- [87] PANAIT L, WIEGAND R P, LUKE S. Improving coevolutionary search for optimal multiagent behaviors[C]// Proceedings of the 18th International Joint Conference on Artificial Intelligence. San Francisco, CA, USA: ACM, 2003: 653–660.
- [88] BUCCIA A, POLLACK J B. On identifying global optima in cooperative coevolution[C]//GECCO 05: Genetic and Evolutionary Computation Conference. Washington DC, USA: ACM, 2005: 539–544.

- [89] SHI M. Comparison of sorting algorithms for multi-fitness measurement of cooperative coevolution[C]//Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference. Montréal, Québec, Canada: ACM, 2009: 2583–2588.
- [90] SHI M, GAO S. Reference sharing: A new collaboration model for cooperative coevolution[J]. *Journal of Heuristics*, 2017, 23(1): 1–30.
- [91] SHI M, WU H F. Pareto cooperative coevolutionary genetic algorithm using reference sharing collaboration[C]//GECCO'09: Genetic and Evolutionary Computation Conference. New York, NY, USA: ACM, 2009: 867–874.
- [92] MORIARTY D E, MIIKKULAINEN R. Forming neural networks through efficient and adaptive coevolution [J]. *Evolutionary Computation*, 1997, 5(4): 373–399.
- [93] WIEGAND R P, LILES W C, DE JONG K A. An empirical analysis of collaboration methods in cooperative coevolutionary algorithms[C]//Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation (GECCO). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001: 1235–1245.
- [94] FICICI S G, POLLACK J B. Pareto optimality in coevolutionary learning[M]//KELEMEN J, SOSÍK P. 6th European Conference, ECAL 2001. Prague, Czech Republic: Springer, 2001, 2159: 316–325.
- [95] DE JONG E D. Towards a bounded pareto-coevolution archive[C]//Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753). Portland, OR, USA: IEEE, 2004: 2341–2348.
- [96] HAMEED A, CORNE D, MORGAN D, et al. Large-scale optimization: Are co-operative co-evolution and fitness inheritance additive?[C]//2013 13th UK Workshop on Computational Intelligence (UKCI). Guildford, UK: IEEE, 2013: 104–111.
- [97] HAMEED A H A, KONONOVA A, CORNE D. Engineering fitness inheritance and co-operative evolution into state-of-the-art optimizers[C]//2015 IEEE Symposium Series on Computational Intelligence. Cape Town, South Africa: IEEE, 2015: 1695–1702.
- [98] GOH C K, TAN K C. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(1): 103–127.
- [99] IORIO A W, LI X D. A cooperative coevolutionary multiobjective algorithm using non-dominated sorting[M]//Genetic and Evolutionary Computation Conference-Gecco 2004, Pt 1, Proceedings. Seattle, WA, USA: Springer, 2004, 3102: 537–548.
- [100] TAN K C, LEE T H, YANG Y J, et al. A cooperative coevolutionary algorithm for multiobjective optimization[C]//2004 IEEE International Conference on Systems, Man & Cybernetics (IEEE Cat. No. 04CH37583). Hague, Netherland: IEEE, 2004: 1926–1931.
- [101] WIEGAND R P. Applying diffusion to a cooperative coevolutionary model[C]// International Conference on Parallel Problem Solving from Nature. Springer, Berlin, Heidelberg, 1998: 560–569.
- [102] OMIDVAR M N, LI X D, YAO X. Smart use of computational resources based on contribution for cooperative co-evolutionary algorithms[C]//Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. New York, NY, USA: ACM, 2011: 1115–1122.
- [103] OMIDVAR M N, KAZIMIPOUR B, LI X D, et al. Cbcc3—a contribution-based cooperative co-evolutionary algorithm with improved exploration/exploitation balance[C]//2016 IEEE Congress on Evolutionary Computation (CEC). Vancouver, BC, Canada: IEEE, 2016: 3541–3548.
- [104] 关世伟. 大规模全局优化问题的高效算法研究 [D]. 西安: 西安电子科技大学, 2018.
- GUAN S W. Efficient algorithms for large scale global optimization[D]. Xi'an: Xidian University, 2018 (in Chinese).
- [105] CUI M J, LI L, SHI M J. A selective biogeography-based optimizer considering resource allocation for large-scale global optimization[J]. *Computational Intelligence and Neuroscience*, 2019: 1240162.
- [106] MÜHLENBEIN H, PAAB G. From recombination of genes to the estimation of distributions I. Binary parameters[C]//International Conference on Evolutionary Computation—The 4th International Conference on Parallel Problem Solving from Nature. Berlin, Germany: Springer, 1996: 178–187.
- [107] PELIKAN M, GOLDBERG D E, CANTÚ-PAZ E. BOA: The bayesian optimization algorithm[C]//GECCO'99: Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999: 525–532.
- [108] HARIK G R, LOBO F G, GOLDBERG D E. The compact genetic algorithm[J]. *IEEE Transactions on Evolutionary Computation*, 1999, 3(4): 287–297.
- [109] Baluja S. Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning[R]. Carnegie-Mellon Univ Pittsburgh Pa Dept of Computer Science, 1994.
- [110] 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.
- [111] XU Q, SANYANG M L, KABÁN A. Large scale con-

- tinuous EDA using mutual information[C]//2016 IEEE Congress on Evolutionary Computation (CEC). Vancouver, BC, Canada: IEEE, 2016: 3718–3725.
- [112] YANG P, TANG K, YAO X. Turning high-dimensional optimization into computationally expensive optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2018, 22(1): 143–156.
- [113] YANG P, TANG K, YAO X. A parallel divide-and-conquer-based evolutionary algorithm for large-scale optimization[J]. *IEEE Access*, 2019, 7: 163105–163118.
- [114] MÜHLENBEIN H, MAHNIG T. FDA—a scalable evolutionary algorithm for the optimization of additively decomposed functions[J]. *Evolutionary Computation*, 1999, 7(4): 353–376.
- [115] ZHANG Q F. On stability of fixed points of limit models of univariate marginal distribution algorithm and factorized distribution algorithm[J]. *IEEE Transactions on Evolutionary Computation*, 2004, 8(1): 80–93.
- [116] SCHAFFER J D, MORISHIMA A. An adaptive crossover distribution mechanism for genetic algorithms[C]//Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc, 1987: 36–40.
- [117] LEVENICK J. Metabits: Generic endogenous crossover control[C]//Proceedings of the Sixth International Conference on Genetic Algorithms. San Francisco, CA, United States: Morgan Kaufmann Publishers Inc., 1995.
- [118] SMITH J, FOGARTY T C. Recombination strategy adaptation via evolution of gene linkage[C]//Proceedings of IEEE International Conference on Evolutionary Computation. Nagoya, Japan: IEEE, 1996: 826–831.
- [119] SOLIS F J, WETS R J B. Minimization by random search techniques[J]. *Mathematics of Operations Research*, 1981, 6(1): 19–30.
- [120] HANSEN N, OSTERMEIER A. Completely derandomized self-adaptation in evolution strategies[J]. *Evolutionary Computation*, 2001, 9(2): 159–195.
- [121] OMIDVAR M N, LI X D. A comparative study of CMA-ES on large scale global optimisation[C]//23rd Australasian Joint Conference on Artificial Intelligence. Adelaide, Australia: Springer, 2010.
- [122] MAHDAVI S, SHIRI M E, RAHNAMEYAN S. Metaheuristics in large-scale global continuous optimization: A survey[J]. *Information Sciences*, 2015, 295: 407–428.
- [123] LATORRE A, MUELAS S, PEÑA J M. A comprehensive comparison of large scale global optimizers[J]. *Information Sciences*, 2015, 316: 517–549.
- [124] WEISE T, CHIONG R, TANG K. Evolutionary optimization: Pitfalls and booby traps[J]. *Journal of Computer Science and Technology*, 2012, 27(5): 907–936.
- [125] AUGER A, HANSEN N, MAUNY N, et al. Bio-inspired continuous optimization: the coming of age[C]//CEC 2007. Piscataway, NJ, USA: [s.n.], 2007.
- [126] TOINT P L. Test problems for partially separable optimization and results for the routine pspmin[R]. Belgium: The University of Namur, Department of Mathematics, 1983.
- [127] MA X L, LI X D, ZHANG Q F, et al. A survey on cooperative co-evolutionary algorithms[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(3): 421–441.
- [128] TEZUKA M, MUNETOMO M, AKAMA K. Linkage identification by nonlinearity check for real-coded genetic algorithms[M]//DEB K. Genetic and Evolutionary Computation Conference. Seattle, WA, USA: Springer, 2004, 3103: 222–233.
- [129] PENG X G, JIN Y C, WANG H D. Multimodal optimization enhanced cooperative coevolution for large-scale optimization[J]. *IEEE Transactions on Cybernetics*, 2019, 49(9): 3507–3520.
- [130] SAYED E, ESSAM D, SARKER R. Using hybrid dependency identification with a memetic algorithm for large scale optimization problems[C]//9th International Conference, SEAL 2012. Hanoi, Vietnam: Springer, 2012: 168–177.
- [131] WEICKER K, WEICKER N. On the improvement of coevolutionary optimizers by learning variable interdependencies[C]//Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406). Washington, DC, USA: IEEE, 1999: 1627–1632.
- [132] WU X J, WANG Y P, LIU J H, et al. A new hybrid algorithm for solving large scale global optimization problems[J]. *IEEE Access*, 2019, 7: 103354–103364.
- [133] CAI X W, GAO L, LI X Y. Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems[J]. *IEEE Transactions on Evolutionary Computation*, 2020, 24(2): 365–379.
- [134] YANG Z, QIU H B, GAO L, et al. A surrogate-assisted particle swarm optimization algorithm based on efficient global optimization for expensive black-box problems[J]. *Engineering Optimization*, 2019, 51(4): 549–566.
- [135] BOUHLEL M A, BARTOLI N, REGIS R, et al. Efficient global optimization for high-dimensional constrained problems by using the kriging models combined with the partial least squares method[J]. *Engineering Optimization*, 2018, 50(12): 2038–2053.
- [136] JIN Y C. Surrogate-assisted evolutionary computation: Recent advances and future challenges[J]. *Swarm and Evolutionary Computation*, 2011, 1(2): 61–70.
- [137] ZHAN D W, XING H L. Expected improvement for

- expensive optimization: A review[J]. *Journal of Global Optimization*, 2020, 78(3): 507–544.
- [138] LIU B, ZHANG Q F, GIELEN G G E. A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(2): 180–192.
- [139] SUN C L, DING J L, ZENG J C, et al. A fitness approximation assisted competitive swarm optimizer for large scale expensive optimization problems[J]. *Memetic Computing*, 2018, 10(2): 123–134.
- [140] TENNE Y, IZUI K, NISHIWAKI S. Dimensionality-reduction frameworks for computationally expensive problems[C]//2010 IEEE Congress on Evolutionary Computation. Barcelona, Spain: IEEE, 2010.
- [141] SUN C L, JIN Y C, CHENG R, et al. Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems[J]. *IEEE Transactions on Evolutionary Computation*, 2017, 21(4): 644–660.
- [142] YU H B, TAN Y C, ZENG J C, et al. Surrogate-assisted hierarchical particle swarm optimization[J]. *Information Sciences*, 2018, 454-455: 59–72.
- [143] FU G X, SUN C L, TAN Y, et al. A surrogate-assisted evolutionary algorithm with random feature selection for large-scale expensive problems[M]//BÄCK T, PREUSS M, DEUTZ A, et al. *Parallel Problem Solving from Nature – PPSN XVI*. Cham: Springer, 2020: 125–139.
- [144] 陈祺东. 基于 QPSO 算法求解复杂优化问题的策略研究 [D]. 无锡: 江南大学, 2020.
CHEN Q D. Strategies of solving complex optimization problems based on QPSO algorithm[D]. Wuxi: Jiangnan University, 2020 (in Chinese).
- [145] WANG X J, WANG G G, SONG B W, et al. A novel evolutionary sampling assisted optimization method for high-dimensional expensive problems[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(5): 815–827.
- [146] DONG H C, DONG Z M. Surrogate-assisted grey wolf optimization for high-dimensional, computationally expensive black-box problems[J]. *Swarm and Evolutionary Computation*, 2020, 57: 100713.
- [147] WERTH B, PITZER E, AFFENZELLER M. Enabling high-dimensional surrogate-assisted optimization by using sliding windows[C]//Proceedings of the Genetic and Evolutionary Computation Conference Companion. Berlin, Germany, Deutschland: ACM, 2017: 1630–1637.
- [148] REN Z G, PANG B, WANG M Y, et al. Surrogate model assisted cooperative coevolution for large scale optimization[J]. *Applied Intelligence*, 2019, 49(2): 513–531.
- [149] BLANCHARD J, BEAUTHIER C, CARLETTI T. A surrogate-assisted cooperative co-evolutionary algorithm for solving high dimensional, expensive and black box optimization problems[M]//RODRIGUES H C, HERSKOVITS J, MOTA SOARES C M, et al. *EngOpt 2018 Proceedings of the 6th International Conference on Engineering Optimization*. Cham: Springer, 2019: 41–52.
- [150] BLANCHARD J, BEAUTHIER C, CARLETTI T. A surrogate-assisted cooperative co-evolutionary algorithm using recursive differential grouping as decomposition strategy[C]//2019 IEEE Congress on Evolutionary Computation (CEC). Wellington, New Zealand: IEEE, 2019: 689–696.
- [151] DE FALCO I, CIOPPA A D, TRUNFIO G A. Investigating surrogate-assisted cooperative coevolution for large-scale global optimization[J]. *Information Sciences*, 2019, 482: 1–26.