

异构集成代理辅助的区间多模态粒子群优化算法

季新芳¹ 张勇¹ 巩敦卫¹ 郭一楠^{1,2} 孙晓燕¹

摘要 现实生活中的很多黑盒优化问题可归为高计算代价的多模态优化问题 (Multimodal optimization problem, MMOP), 即昂贵多模态优化问题 (Expensive MMOP, EMMOP). 在处理该类问题时, 决策者希望以尽量少的计算代价 (即尽量少的真实函数评价次数) 找到多个高质量的最优解. 然而, 已有代理辅助的进化优化算法 (Surrogate-assisted evolutionary algorithm, SAEA) 很少考虑问题的多模态属性, 运行一次仅可获得问题的一个最优解. 鉴于此, 研究一种异构集成代理辅助的区间多模态粒子群优化 (Interval multimodal particle swarm optimization algorithm assisted by heterogeneous ensemble surrogate, IMPSO-HES) 算法. 首先, 借助异构集成的思想构建一个由多个基础代理模型组成的模型池; 随后, 依据待评价粒子与已发现模态之间的匹配关系, 从模型池中自主选择部分基础代理模型进行集成, 并使用集成后的代理模型预测该粒子的适应值. 进一步, 为节约代理模型管理的代价, 设计一种增量式的代理模型管理策略; 为减少代理模型预测误差对算法性能的影响, 首次将区间排序关系引入到进化过程中. 将所提算法与当前流行的 5 种代理辅助进化优化算法和 7 种最先进的多模态优化算法进行对比, 在 20 个测试函数和 1 个建筑节能实际问题上的实验结果表明, 所提算法可以在较少计算代价下获得问题的多个高竞争最优解.

关键词 粒子群优化, 多模态优化, 高昂计算代价, 代理辅助

引用格式 季新芳, 张勇, 巩敦卫, 郭一楠, 孙晓燕. 异构集成代理辅助的区间多模态粒子群优化算法. 自动化学报, 2024, 50(9): 1831-1853

DOI 10.16383/j.aas.c210223

Interval Multimodal Particle Swarm Optimization Algorithm Assisted by Heterogeneous Ensemble Surrogate

Ji Xin-Fang¹ ZHANG Yong¹ GONG Dun-Wei¹ GUO Yi-Nan^{1,2} SUN Xiao-Yan¹

Abstract Many real-world black-box optimization problems can be classified as multimodal optimization problems (MMOPs) with high computational cost, that is, expensive multimodal optimization problems (EMMOPs). When dealing with such problems, decision-makers hope to find multiple high-quality solutions with less computational cost (i.e., the least number of real function evaluations). However, existing surrogate-assisted evolutionary algorithms (SAEAs) seldom consider the multimodal properties of problem, and they can only obtain one optimal solution of the problem at a time. In view of this, this paper studies an interval multimodal particle swarm optimization (PSO) algorithm assisted by heterogeneous ensemble surrogate (IMPISO-HES). Firstly, a model pool composed of multiple basic surrogate models is constructed with the idea of heterogeneous ensemble. Then, according to the matching relationship between the particle to be evaluated and the discovered modalities, some basic surrogate models will be selected from the model pool for integration, and the integrated surrogate model is utilized to predict the fitness value of the particle. Furthermore, in order to save the cost of model management, an incremental surrogate model management strategy is designed. In order to reduce the influence of prediction error of surrogate model on the algorithm's performance, the interval ordering relation is introduced into the evolutionary process for the first time. The proposed algorithm is compared with five SAEAs and seven state-of-the-art multimodal algorithms, experimental results on 20 benchmark functions and the building energy conservation problem show that the proposed algorithm can obtain multiple highly-competitive optimal solutions at a low computational cost.

Key words Particle swarm optimization (PSO), multimodal optimization, expensive computational cost, surrogate-assisted

Citation Ji Xin-Fang, Zhang Yong, Gong Dun-Wei, Guo Yi-Nan, Sun Xiao-Yan. Interval multimodal particle swarm optimization algorithm assisted by heterogeneous ensemble surrogate. *Acta Automatica Sinica*, 2024, 50(9): 1831-1853

收稿日期 2021-03-19 录用日期 2021-09-05
Manuscript received March 19, 2021; accepted September 5, 2021

国家自然科学基金 (62273348, 62133015), 北方民族大学青年人才培养项目 (2024QNPY04) 资助

Supported by National Natural Science Foundation of China (62273348, 62133015) and Young Talent Cultivation Project of North Minzu University (2024QNPY04)

本文责任编辑 袁勇

Recommended by Associate Editor YUAN Yong

1. 中国矿业大学信息与控制工程学院 徐州 221116 2. 中国矿业大学机械电子与信息工程学院 (北京) 北京 100083

1. School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116 2. School of Mechanical Electronic and Information Engineering, China University of Mining and Technology (Beijing), Beijing 100083

实际生产生活中的许多问题都具有多模态特性,即存在多个全局或局部最优解,如药物分子设计^[1]、作业车间调度^[2]和蛋白质结构设计^[3]等.这类问题称为多模态优化问题(Multimodal optimization problem, MMOP)^[4-5].在很多情况下,该类问题又同时存在目标函数评价代价昂贵的现象.以基于EnergyPlus仿真软件的建筑节能设计问题为例,在普通电脑上运行一次进化优化算法通常需要几十分钟,甚至数个小时^[6].将同时具有多模态特性和高昂计算代价的优化问题称为昂贵多模态优化问题(Expensive MMOP, MMOP)^[7].

在求解昂贵多模态优化问题时,决策者往往希望可以同时找到它的多个最优解,主要原因如下^[7-8]: 1) 对于很多的实际工程优化问题,设计师通常希望从差异较大的若干优秀解决方案中自主选择满足不同需求的方案; 2) 当采用代理模型辅助的进化优化算法处理该类问题时,受训练数据不足的影响,代理模型拟合出的目标函数的全局最优点未必是实际问题的全局最优点.这些伪最优点会诱导种群舍弃真实最优点,进而陷入局部收敛.此时,同时定位问题的多个优化解,势必可以提高算法找到真实全局最优解的可能性.

针对多模态优化问题,国内外学者已经提出诸多的进化优化算法.依据模态处理方法的不同,这些算法大致可分为3类^[9].第1类为小生境技术,其基本思想是采用某种策略产生若干个多样性好的子种群,使它们分别朝着不同优化解的方向搜索^[10-14].近年来,为克服小生境半径对算法性能的影响,学者们也提出多种无参数的改进小生境技术,如Li^[15]提出的物种形成策略、王湘中等^[16]提出的山谷探索法等.类似地,张贵军等^[17]也设计了一种两阶段退火策略来动态调整小生境半径.特别地,文献^[18-19]提出的适应值共享策略,其思想也是根据个体相似性将种群分为不同的子种群.第2类是种群多样性增强技术,其基本思想是通过增强种群的多样性来确保算法搜索方向的多样性.部分代表性方法如Qu等^[20]提出的基于局部信息交互的粒子群优化算法、Biswas等^[21]提出的局部信息共享的差分进化算法等.第3类是多目标转换技术,其主要思想是将单目标多模态问题转化成一般的多目标优化问题进行求解.变换后问题通常包括两个目标:一个是给定的多模态问题,另一个是衡量解多样性的指标^[9, 22-23].与此同时,这些算法也在诸多实际问题中获得成功应用,如药物分子设计^[1]、特征选择问题^[24]等.但是,当处理昂贵优化问题时,由于需要进行若干次的真实个体评价才能获得问题的最优解,

这些算法依然存在运行代价过高的不足.

目前,求解高昂优化问题的进化算法(Evolutionary algorithm, EA)已有很多,其中最具代表性的是代理辅助的进化优化算法(Surrogate-assisted evolutionary algorithm, SAEA)^[25].该类方法采用计算便宜的代理模型替代昂贵真实函数来评价个体的适应值,可以达到降低算法计算代价的目的.随之,很多优秀的机器学习方法相继用于产生代理模型^[26-29];相应地,诸多典型的SAEA也相继提出^[30-32].从代理模型的使用规模上,这些算法可以分为单代理模型辅助的EA和集成代理模型辅助的EA.部分单代理模型辅助的EA包括多项式回归(Polynomial regression, PR)辅助的EA^[29]、支持向量机辅助的EA^[33]、径向基函数网络(Radial basis function network, RBFN)辅助的EA^[28, 34]和Kriging辅助的EA^[35]或高斯过程模型辅助的EA^[36-38]等.通常不同的代理模型具有不同的特性,适合于不同类型的优化问题.在没有问题先验知识的情况下,决策者很难为当前问题提供一个合适的代理模型.由于没有其他代理模型可以弥补当前代理模型的缺陷,单代理辅助EA的鲁棒性相对较差.

通过合理组合多个不同特性的代理模型,集成代理模型通常可以获得比单一代理模型更为优越的泛化性能.现有代理模型集成策略可分为异构集成和同质集成两种.异构集成由多个不同类型的单个代理模型组成,而同质集成由多个相同类型的单个代理模型组成.为便于区分,下文将集成代理模型中的单个代理模型统称为基础代理模型.在诸多的集成代理模型辅助EA中,最常用的同质集成代理模型是基于RBFN的集成代理.文献^[39-41]采用RBFN同时构建问题的全局和局部代理模型;Wang等^[32]采用Bagging算法构建大量的RBFN代理模型,并在优化过程中自适应集成这些代理模型.在研究异构集成代理模型时,学者们最常用的3种基础代理模型是Kriging、RBFN和PR.学者们采用不同的策略组合这些基础代理模型,分别提出不同的集成代理模型^[42-48].实验表明,与同质集成代理模型相比,基于异构集成代理模型的进化优化算法可以显著提高算法的搜索性能^[25].然而,这些方法都是面向求解问题的一个全局最优解而设计的.在很多情况下,它们只需要保证代理模型在全局最优峰处的精度即可.除在单目标优化问题中采用异构集成模型,陈万芬等^[49]在处理昂贵多目标优化问题时,也给出一种基于Kriging和RBFN的异构加权集成代理模型.然而,由于其处理对象为包含多个冲突目标的多目标优化问题,他们提出的代理模型构

建和更新方法并不适合昂贵多模态优化问题. 当面对包含多个全局最优解的多模态问题时, 如何同时保证代理模型在多个最优峰/谷处的精度, 需要提出更为高效的代理模型集成和管理策略.

此外, 代理模型预测的不确定性在 SAEA 中也扮演着重要的角色. 它会给 SAEA 带来负面影响, 导致 SAEAs 搜索停顿或收敛于假最优, 因此, 需要合理使用这些不确定信息. 在 SAEA 中代理模型不确定性的表示有多种形式, 如采用 Kriging 模型提供的预测不确定性、利用解与其邻域内训练数据的距离来估计不确定性^[50]等. 此外, 在多代理辅助的 EA 中, 可以采用多个代理模型的预测差值刻画预测值的不确定性^[47, 51]. 目前, 大部分 SAEA 仅将不确定信息引入到代理模型管理, 即用于挑选不确定性较大的解^[44, 47, 50-52]进行真实函数评价. 这些方法均从代理模型的角度出发, 期望提高代理模型的精度. 与之不同, 本文从进化角度出发, 将不确定信息融入到个体的适应值评价中, 在减少计算代价的同时避免代理模型不精确导致的进化方向错误等问题.

可以看出, 虽然用于求解多模态优化问题或高昂优化问题的进化算法已有很多, 但是, 少有算法将两者结合在一起. 2011 年 Yahyaie 等^[53]首次给出一种代理辅助的多模态进化优化算法. 首先, 采用自适应网格技术获得目标函数曲面的初始估计; 随后, 利用生成的网格定位问题的局部最优点, 并在局部最优点附近建立多个局部代理模型. 由于仅使用了单一类型的代理模型, 这极大限制了该方法处理复杂多模态优化问题的性能. 为此, 2021 年 Ji 等^[7]提出了一种双代理辅助的双种群粒子群优化算法 (Dual-surrogate assisted cooperative particle swarm optimization, DSCPSO). 该算法设计了一种双种群协同粒子群优化机制, 用来同时探索和开发问题的多个模态; 构建了一种多模态引导的双层协同代理模型, 用于协助种群快速发现新模态和开发已有模态. 尽管 DSCPSO 显著提升了 PSO 处理多模态问题的能力, 但是, 它需要决策者设置多个重要的控制参数, 这在一定程度上限制了它的实际应用效果.

如前所述, 如果直接使用已有的多模态进化算法求解高昂多模态优化问题, 存在计算耗时的问题. 如果将代理模型辅助的进化算法用来处理该类问题, 至少存在如下 3 个困难亟待解决:

1) 选择何种代理模型集成策略来拟合所要求解的高昂多模态优化问题. 现有集成策略只需要保证代理模型在全局最优峰/谷处的精度即可. 当面对包含多个全局最优解的多模态问题时, 如何同时保证集成代理模型在多个最优峰/谷处的精度, 需

要提出更为高效的模型集成策略.

2) 代理模型的管理, 即需要真实函数评价的个体的选择问题和代理模型的更新问题. 选取的个体既能体现出问题的全部模态信息, 又要保证代理模型在已有模态处的精度. 当问题的模态数和模态位置未知时, 还需要设计的代理模型更新策略能够及时响应模态数量和位置的变化.

3) 当采用代理模型代替真实目标函数来预测种群中个体的适应值时, 预测误差会影响种群的搜索性能. 在不增加模型管理代价的前提下, 如何避免预测误差对算法性能的影响, 也是一个需要解决的挑战性问题.

鉴于此, 研究一种异构集成代理辅助的区间多模态粒子群优化算法 (Interval multimodal particle swarm optimization algorithm assisted by heterogeneous ensemble surrogate, IMPSO-HES), 重点解决多模态特性下的代理模型选择与管理问题. 相对已有进化优化算法, 主要贡献如下:

1) 提出一种基于异构集成的代理模型自主构建策略. 针对算法识别得到的每个模态, 从代理模型池中自主选择合适的异构基础代理模型进行集成. 通过对每个模态建立合适的集成代理模型, 该策略可以同时保证代理模型在多个最优峰/谷处的精度.

2) 给出一种增量式的代理模型管理和更新策略. 在算法迭代过程中, 不断识别可能发现的新模态, 并利用这些模态对应的最优粒子/个体重新构建新的基础代理模型; 同时, 利用这些最优粒子作为填充样本, 重新训练模型池中部分相关的旧基础代理模型. 该策略不仅可以保障模型池中异构基础代理模型的质量, 而且能够节约模型池的管理代价.

3) 首次引入区间数来表示模型预测得到的粒子适应值, 并给出一种基于区间排序的粒子位置更新策略. 相对传统基于精确值的粒子比较策略, 该策略同时考虑了粒子预测适应值的大小和不确定程度, 其得到的粒子排序结果更为可靠.

相对 DSCPSO (Dual-surrogate assisted cooperative PSO)^[7]等已有方法, IMPSO-HES 具有如下优势或改进:

1) DSCPSO 采用两个种群分别实现问题新模态的搜索和旧模态的开采, 而 IMPSO-HES 只需单个种群即可实现上述目标, 其所需控制参数更少.

2) DSCPSO 仅利用了 RBFN 来构建双层代理模型, 而 IMPSO-HES 同时采用了两类代理模型, 即 RBFN 和 PR. 一方面, 由于 RBFN 和 PR 的组合可以处理更多类型的优化问题, IMPSO-HES 的应用范围更广; 另一方面, 由于 IMPSO-HES 可以

从模型池中为每个粒子自主选择合适的基础代理模型,其得到的粒子评估结果更为准确.

3) 在 DSCPSO 中,当数据库发生变化时需要重新构建所有的代理模型;与之不同,IMPSON-HES 采用增量式方法有选择性地更新部分基础代理模型,可以减少代理模型的更新代价.

4) IMPSON-HES 考虑到了代理模型预测的不确定性,其得到的粒子排序结果相对更为可靠.

本文结构安排如下:第 1 节主要介绍相关工作;第 2 节详细介绍所提 IMPSON-HES 的框架及其改进策略;第 3 节和第 4 节则在数值问题和建筑能源实例上验证所提算法的有效性;第 5 节总结全文并给出研究展望.

1 相关工作

1.1 粒子群优化算法

考虑如下单目标优化问题:

$$\begin{aligned} \min \quad & f(\mathbf{X}) \\ \text{s.t.} \quad & \mathbf{X} \in S \subseteq \mathbf{R}^D \end{aligned} \quad (1)$$

其中, \mathbf{X} 为 D 维决策变量, S 为决策变量 \mathbf{X} 的取值空间, \mathbf{R}^D 是 D 维空间, $f(\cdot)$ 为目标函数. 具体到昂贵多模态优化问题时, $f(\cdot)$ 将同时具有多个性能相同的最优解,并且具有高昂的计算代价.

PSO 是由 Eberhart 等^[54] 模拟鸟类觅食等群体行为提出的一类启发式搜索算法. 在 PSO 中,每个粒子代表优化问题的一个“潜在可行解”,并以一定速度在搜索空间中不断更新自身的位置. 以常用的带惯性权重 PSO 算法^[55] 为例,粒子更新公式为

$$\begin{aligned} v_{i,d}(t+1) &= wv_{i,d}(t) + c_1r_1(Pbest_{i,d}(t) - x_{i,d}(t)) + \\ & \quad c_2r_2(Gbest_d(t) - x_{i,d}(t)) \\ x_{i,d}(t+1) &= x_{i,d}(t) + v_{i,d}(t+1) \end{aligned} \quad (2)$$

其中, t 为迭代次数, $\mathbf{X}_i(t) = (x_{i,1}(t), \dots, x_{i,D}(t))$ 和 $\mathbf{V}_i(t) = (v_{i,1}(t), \dots, v_{i,D}(t))$ 分别表示第 i 个粒子的位置和速度; $Pbest_i(t)$ 为第 i 个粒子目前为止所发现的最优位置,简称个体引导者; $Gbest(t)$ 为整个粒子群目前为止所发现的最优位置,简称全局引导者. w 为惯性权重; c_1 和 c_2 是两个学习因子; r_1 和 r_2 为 $[0, 1]$ 中的随机数.

相对遗传算法等传统进化优化技术,PSO 具有收敛速度快和易于实现等优点,目前已广泛应用于各种优化问题^[30, 40, 49, 56-57]. 为了提升 PSO 处理多模态优化问题的能力,学者们也设计了多种高效的多模态粒子群优化算法. 这其中,比较经典的算法是 Qu 等^[20] 提出的基于局部信息交互的粒子群优化算

法 (Locally informed particle swarm, LIPS). 该算法以每个粒子为中心形成一个小生境,采用小生境内最优信息来引导该粒子的飞行,可以很好地均衡种群的全局探索和局部开发能力. 鉴于此,本文使用 LIPS 来指导粒子群的进化. LIPS 提出的粒子更新公式为

$$\begin{cases} v_{i,d}(t+1) = w(v_{i,d}(t) + \varphi(Pb_{i,d}(t) - x_{i,d}(t))) \\ Pb_i(t) = \frac{\sum_{j=1}^{n_{\text{size}}} \varphi_j nbest_j(t)}{\varphi} \\ x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1) \end{cases} \quad (3)$$

其中, n_{size} 为邻居规模,其值随进化过程动态变化; $nbest_j(t)$ 为个体引导者 $Pbest_i$ 的第 j 个最优邻居; φ_j 为 $[0, 4.1/n_{\text{size}}]$ 范围内均匀分布的随机数, $\varphi = \sum_{j=1}^{n_{\text{size}}} \varphi_j$.

1.2 代理模型

RBFN 和 PR 是 SAEA 中常用的两种代理模型^[42-43]. 已有研究表明,PR 适合解决低阶非线性问题,而 RBFN 模型适合解决高维高阶非线性问题^[58]. 鉴于它们之间的互补性,本文使用它们共同构建模型池中的基础代理模型.

RBFN 是一种含有单隐层的前馈神经网络,它使用径向基函数作为隐层神经元的激活函数. 输入层到隐含层的转换是非线性的,但输出层是对隐层神经元输出的线性组合. 假设网络输入为 D 维向量 \mathbf{X} ,则 RBFN 的输出可表示为

$$\hat{f}(\mathbf{X}) = \sum_{i=1}^q w_i \phi(\mathbf{X}, c_i) \quad (4)$$

其中, q 为隐层神经元的个数, c_i 是第 i 个隐层神经元所对应基函数的中心; w_i 为第 i 个隐层神经元的权重; $\phi(\mathbf{X}, c_i)$ 是径向基函数,其形式有多种,如高斯和多重二次曲面 (multi-quadric, MQ)^[59]. 本文采用 MQ 基函数,其表达式为

$$\phi(\mathbf{X}, c_i) = \sqrt{\|\mathbf{X} - c_i\|^2 + b_i^2} \quad (5)$$

其中, b_i 为第 i 个隐层神经元所对应的偏置.

在工程设计中 PR 是最常使用的一种代理模型. 它的一个二阶多项式模型可以表示为

$$\hat{y} = \beta_0 + \sum_{i=1}^D \beta_i x_i + \sum_{i=1}^D \beta_{ii} x_i^2 + \sum_{i=1}^D \sum_{j<i}^D \beta_{ij} x_i x_j \quad (6)$$

其中, β_0 和 β_i 是需要估计的系数, D 为输入变量的维数;上式中总共有 $(D+1)(D+2)/2$ 项. 通常采用

最小二乘法估计模型的未知系数.

1.3 区间优化

定义 1. 对于实数轴上的区间数 $\bar{a} = [a^L, a^U]$, 可由中点 M_a 和半宽 W_a 表示为

$$\bar{a} = \langle M_a, W_a \rangle \quad (7)$$

其中, $M_a = \frac{1}{2}(a^L + a^U)$, $W_a = \frac{1}{2}(a^U - a^L)$; a^L 和 a^U 为区间数 a 的下界和上界.

对于两个区间数, Zhou 等^[60] 定义了一种基于可能度的比较关系.

定义 2. 假设 $\bar{a}_1 = \langle M_{a_1}, W_{a_1} \rangle$ 和 $\bar{a}_2 = \langle M_{a_2}, W_{a_2} \rangle$ 为两个区间数, 那么, $\bar{a}_1 \geq \bar{a}_2$ 的可能度为

$$P(\bar{a}_1 \geq \bar{a}_2) = \max \left\{ 1 - \frac{1}{2} \max \left(\frac{M_{a_2} - M_{a_1}}{W_{a_2} + W_{a_1}} + 1, 0 \right), 0 \right\} \quad (8)$$

类似地, 有

$$P(\bar{a}_1 \leq \bar{a}_2) = \max \left\{ 1 - \frac{1}{2} \max \left(\frac{M_{a_1} - M_{a_2}}{W_{a_2} + W_{a_1}} + 1, 0 \right), 0 \right\} \quad (9)$$

其中, $P(\bar{a}_1 \leq \bar{a}_2) + P(\bar{a}_1 \geq \bar{a}_2) = 1$. $P(\bar{a}_1 \geq \bar{a}_2) > 0.5$ 意味着 \bar{a}_1 大于等于 \bar{a}_2 , 而 $P(\bar{a}_1 \leq \bar{a}_2) > 0.5$ 意味着 \bar{a}_1 小于等于 \bar{a}_2 .

对于一组区间数, 一种常用的排序方法是基于可能度矩阵的行求和法^[61]. 假如需要对一组区间数 $\bar{a}_1, \bar{a}_2, \dots, \bar{a}_n$ 进行排序, 首先采用式 (8) 或式 (9) 对区间数进行两两比较, 求得相应的可能度 p_{ij} , $i, j = 1, 2, \dots, n$, 进而建立可能度矩阵 P , 即

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}$$

然后, 基于所得矩阵 P , 构造布尔矩阵 $Q = [q_{ij}]_{n \times n}$, 其中

$$q_{ij} = \begin{cases} 1, & p_{ij} \geq 0.5 \\ 0, & p_{ij} < 0.5 \end{cases}$$

随后, 求取每行布尔值之和, 即 $r_i = \sum_{j=1}^n q_{ij}$, 得到 n 个区间数的排序向量 $\mathbf{r} = \{r_1, r_2, \dots, r_n\}$; 最后, 根据 r_i 的大小对这些区间数进行排序.

2 所提 IMPSO-HES 算法

所提代理辅助的区间多模态粒子群优化算法 (IMPSO-HES) 主要包括三个创新工作, 即异构集

成代理模型的构建、基于区间适应值的粒子更新、增量式的代理模型更新. 首先, 给出 IMPSO-HES 的基本框架.

2.1 算法框架

设计 IMPSO-HES 算法的目的是, 在计算资源有限的情况下为决策者同时提供多个高质量的最优解. 图 1 给出了算法的基本框架图. 该框架主要包括 4 个模块, 即模型池的构建、基础代理模型选择与集成、基于区间 PSO 更新种群以及代理模型管理与更新. 其中, “模型池的构建”模块的主要功能是, 为被优化问题提供多个多样性好的基础代理模型; “基础代理模型选择与集成”模块的作用是, 针对每个粒子从模型池中选出最适合的 Q 个基础代理模型, 集成这些代理模型并用其预测该粒子的区间适应值; “基于区间 PSO 更新种群”模块的目的是, 采用适于区间适应值的进化搜索策略不断更新种群中粒子的位置, 以发现更多高质量的最优解, 具体地, 本文采用一种融合小生境技术的区间 PSO 来不断更新粒子群; “代理模型管理与更新”模块主要用于从当前种群中选择少量的代表粒子 (称为种子) 作为新增样本, 并采用一种增量式的学习方法更新模型池中的基础代理模型.

进一步, 算法 1 给出了 IMPSO-HES 的伪代码. 首先, 采用拉丁超立方方法 (Latin hypercube sampling, LHS) 产生规模为 $3D$ 的初始训练数据库 (Database, DB); 其次, 分别以 50% 的概率选择 RBFN 和 PR 作为代理模型, 采用异构集成方式产生 K_{ini} 个基础代理模型, 并由其构建初始代理模型池; 接着, 采用动态 K-means 聚类方法从 DB 中找出 N_K 个初始种子, 并将它们放入种子集合 $Seeds$ 中; 最后, 随机生成规模为 N_p 的初始种群, 并循环执行所提代理模型辅助的区间粒子群优化算法, 直到达到终止条件.

算法 1. IMPSO-HES 算法伪代码

输入. 真实目标函数 f , 最大真实评价次数 TE_{max} , 集成规模 Q , 模型更新间隔代数 g_{max} .

输出. 最优解集, 即种子集合 $Seeds$.

- 1) 采用 LHS 产生 $3D$ 个初始样本, 并将其存入 DB;
- 2) 基于 DB 中的样本, 构建基础代理模型池 (详见第 2.2 节);
- 3) 利用动态 K-means 聚类技术从 DB 中找出 N_K 个初始种子, 并存入种子集合 $Seeds$;
- 4) 生成规模为 N_p 的初始种群;
- 5) **While** 真实评价次数 $< TE_{\text{max}}$
- 6) 执行基础代理模型选择与集成策略, 为每个模态

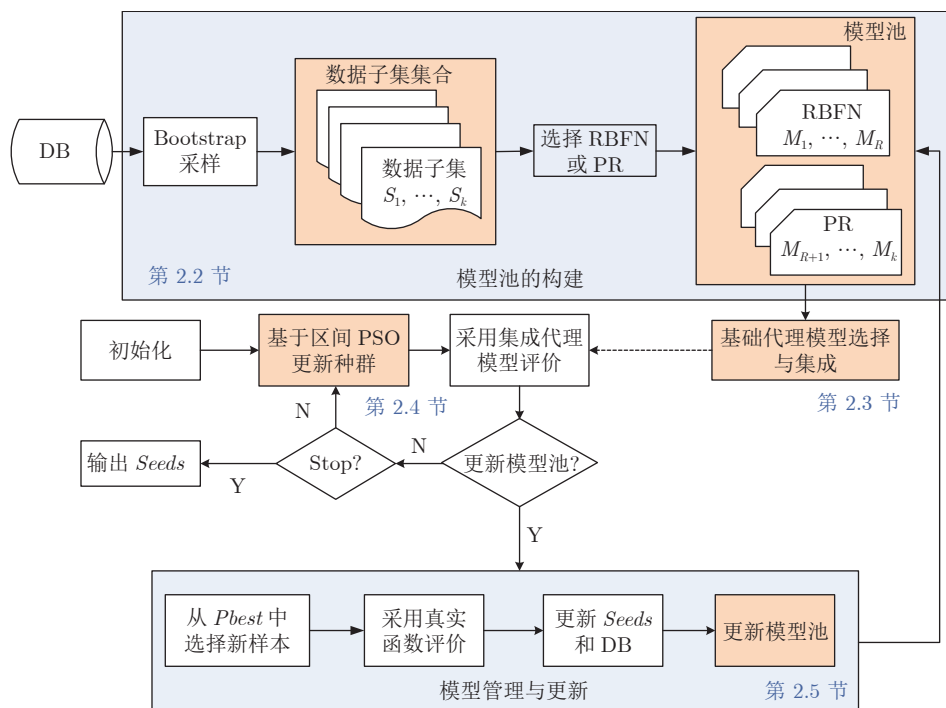


图 1 IMPSO-HES 的框架图

Fig.1 General framework of IMPSO-HES

- 分配 Q 个基础代理模型 (详见第 2.3 节);
- 7) 连续执行 g_{\max} 次的区间 PSO 更新策略, 输出得到的个体引导者集合 P_{best} (详见第 2.4 节);
 - 8) 从 P_{best} 中选出代表新模态的最优解, 存入临时集合 $Sample$ (详见第 2.5 节);
 - 9) 利用真实函数 f 评价集合 $Sample$ 中解, 并将 $Sample$ 中解存入 $Seeds$;
 - 10) 基于 $Sample$ 和 DB , 采用增量式方法更新基础代理模型池 (详见算法 4);
 - 11) 重新识别优化问题的模态信息, 即从 $Seeds$ 中选出精英种子;
 - 12) 利用 $Sample$ 中的解更新数据库 DB ;
 - 13) **End while.**

区间粒子群优化算法 (算法 1 的 5 ~ 13 行) 又包含如下步骤: a) 基于当前识别出的模态, 即 $Seeds$ 中的精英种子, 为每个模态分配 Q 个基础代理模型. b) 循环执行区间 PSO 更新策略 g_{\max} 次, 并输出得到个体引导者集合 P_{best} . 在此期间, 采用集成代理模型评价种群新产生的粒子. 对于任意一个粒子, 具体评价方法如下: 首先, 找到与当前粒子距离最近的种子 (即模态); 接着, 集成该种子所分配的 Q 个基础代理模型, 并使用集成代理模型预测该粒子的区间适应值. c) 从 P_{best} 中选出多样性好的最优解存入 $Sample$ 中, 利用真实函数评价 $Sample$ 中所有解. d) 基于 $Sample$ 和 DB 中样本, 采

用增量式方法更新模型池中的基础代理模型. e) 更新种子集合 $Seeds$ 和数据集 DB . 整个算法迭代结束后, 种子集合 $Seeds$ 中的精英种子即为问题的求解结果.

2.2 初始基础代理模型池的构建

使用改进的 Bootstrap 采样技术创建训练数据子集, 并选择 RBFN 和 PR 两种代理模型来构建所需要的异构基础代理模型. 已有研究表明, PR 适合解决低阶非线性问题, 而 RBFN 模型适合解决高维高阶非线性问题^[57]. 为了权衡模型集成的精度和构建模型池的代价, 首先需要确定初始模型池的规模 K_{ini} . 文献 [32] 和文献 [62] 分别将模型池的规模设置为 2 000 和 $\max((D+1)(D+2)/2, 100)$. 本节根据 Bootstrap 的采样原理^[63] 设置最小 K_{ini} 值.

假设数据集 DB 的规模为 m , 当使用 Bootstrap 进行 m 次采样时, 一个样本不被采集到的概率为

$$p_m = \left(1 - \frac{1}{m}\right)^m \quad (10)$$

求式 (10) 中 m 为 ∞ 时的极限, 可得 $p_m = 1/e \approx 0.368$. 因此, 每次采集到的不重复数据约占数据总量的 $2/3$ ^[63]. 同时, DB 中具有 $2m/3$ 个不重复数据的组合共有 $C_m^{[2m/3]}$ 种. 为了保证代理模型集成的精度, 将 K_{ini} 的最小值设置为 $C_m^{[2m/3]}$. 进一步, 为减

小建模的代价, 设置基础代理模型数量的上限为 $10D$, 其中, D 为优化问题变量的维数. 通常被优化问题的变量维数越高, 其曲线越难拟合, 其建立的基础代理模型也应越多. 基于此, 初始模型池的规模设为 $K_{ini} = \min(C_m^{\lfloor 2m/3 \rfloor}, 10D)$.

Bootstrap 是一种有放回式数据随机采样方法, 在其采集到的数据子集中很可能会有相同的样本. 为避免同一个数据子集中出现过多相同的样本, 文献 [32] 采用了无放回式概率采集方法. 当需要产生一个新的数据子集时, 该方法以 $p_s = 0.5$ 的概率依次判断 DB 中样本是否需要加入到该子集. 换句话说, 对于 DB 中的任意样本, 若产生的随机数大于 p_s , 则将该样本加入新数据子集, 否则丢弃. 不同于文献 [32], 为使数据子集的数据量与 Bootstrap 中采集到的不重复数据量类似, 本节对其选择概率进行重新估算. 当以概率 p_s 采集样本时, 采样得到的样本数量占总样本 $2/3$ 的概率为

$$P\left(\frac{2m}{3} | m\right) = C_m^{\frac{2m}{3}} p_s^{\frac{2m}{3}} (1-p_s)^{\frac{m}{3}} \quad (11)$$

其中, P 表示从规模为 m 的数据中选中 $2m/3$ 个不同数据的概率. P 值越大, 则以概率 p_s 进行无放回采样时的结果越逼近 Bootstrap 方法的采样结果. 采用 MATLAB 中的 Fmincon Solver 求解不同 m 值时的 p_s 值, 所得结果表明, 当 $p_s = 2/3$ 时 P 值最大. 因此, 设置采样概率 $p_s = 2/3$.

算法 2 展示了初始基础代理模型池构建策略的伪代码. 该策略以相同概率 ($p_m = 0.5$) 产生类型为 RBFN 的基础代理模型和类型为 PR 的基础代理模型, 其中, p_m 为选择 RBFN 产生基础代理模型的概率. 在后续的模型更新过程中, 根据基础代理模型的使用情况自适应调整 p_m 值, 具体见算法 3 的 9、10 和 12 行. 某类基础代理模型使用越频繁, 说明该类基础代理模型越适用于当前问题, 因此, 在下次模型更新时其被选择的概率越高. 每个 RBFN 模型在隐含层包含 D 个神经元.

算法 2. 初始基础代理模型池构建的伪代码

输入. 训练数据集 DB; 初始模型池的规模 K_{ini} ; 模型类的选择概率 p_m .

输出. 数据子集 ($S_1, \dots, S_{K_{ini}}$) 和初始模型池 ($M_1, \dots, M_{K_{ini}}$).

- 1) **For** $i = 1 : K_{ini}$ % 依次构建 K_{ini} 个基础代理模型
- 2) $S_i = \emptyset$;
- 3) **For** DB 中的任意样本;
- 4) **If** $\text{rand} < p_s$, 将该样本存入 S_i , **end if**;
- 5) **End for**
- 6) **If** $\text{rand} < p_m$

- 7) 基于样本集 S_i 构建 RBFN 类型的基础代理模型, 记为 M_i ;
- 8) **else**
- 9) 基于样本集 S_i 构建 PR 类型的基础代理模型, 记为 M_i ;
- 10) **End if**
- 11) **End for**.

2.3 基础代理模型的选择与集成

针对不同的待评价粒子选择合适的基础代理模型, 用来构建集成代理模型. 对于传统的昂贵优化问题, 学者们已经给出了多种基础代理模型的选择策略, 如文献 [32] 给出的基于排序的集成代理模型选择策略. 然而, 在很多情况下, 这些方法只能保证代理模型在全局最优点处的精度. 因为在处理多模态优化问题时需要同时获取问题的多个最优解, 所以, 希望种群中属于不同模态的粒子可以采用更适合自己的集成代理模型. 鉴于此, 本节提出一种基于模态匹配的基础代理模型选择策略.

在进行模型集成之前, 首先需要识别当前种群的模态信息. 由第 2.1 节可知, 在很大程度上当前种群的模态信息会隐藏于种子集合 *Seeds* 中. 考虑到 *Seeds* 会同时包含多个属于相同模态的种子, 需要对 *Seeds* 进行动态聚类, 仅选取每一类中的最优种子 (即精英种子) 来表示问题当前的模态信息. 在获取问题的模态信息之后, 接着, 针对每个精英种子, 计算 K 个基础代理模型的预测值与其真实适应值之间的差值 (K 为当前模型池中基础代理模型的数量), 并按差值大小对所有基础代理模型进行排序; 选择前 Q 个预测误差值最小的基础代理模型, 组成当前精英种子所对应模态的基础代理模型集合. 算法 3 给出了所提基础代理模型选择策略的伪代码.

算法 3. 基于模态匹配的基础代理模型选择策略的伪代码

输入. 种子集合 *Seeds*; 模型池 (M_1, \dots, M_K); 集成规模 Q .

输出. 每个模态的 Q 个基础代理模型; 使用 RBFN 更新模型池的概率 p_m .

- 1) 初始化精英种子集合, $ElitSeeds = \emptyset$;
- 2) 对 *Seeds* 进行 K-means 动态聚类, 选择每类中最优种子放入 *ElitSeeds*;
- 3) 初始化 RBFN 和 PR 模型的使用次数, $L_{nbf} = 0$ 和 $L_{pr} = 0$;
- 4) **For** $i = 1 : |ElitSeeds|$ % 依次确定每个精英种子的基础代理模型集合

- 5) 依次使用 K 个基础代理模型 ($M_k, k = 1, 2, \dots, K$) 预测 $ElitSeeds(i)$ 的适应值;
- 6) 分别计算 K 个预测值与 $ElitSeeds(i)$ 真实适应值之间的差值, 记为 (e_1, \dots, e_K) ;
- 7) 按照 (e_1, \dots, e_K) 值大小对 K 个基础代理模型进行排序;
- 8) 选取前 Q 个误差小的基础代理模型, 组成 $ElitSeeds(i)$ 的基础代理模型集合;
- 9) 统计 Q 个基础代理模型中属于 RBFN 和 PR 模型的个数, 分别记为 nbf 和 pr ;
- 10) 更新 $Lnbf \leftarrow Lnbf + nbf, Lpr \leftarrow Lpr + pr$;
- 11) **End for**
- 12) $p_m = Lnbf / (Lnbf + Lpr)$.

得到每个模态的 Q 个基础代理模型后, 采用简单常用的线性集成策略获得其最终的集成代理模型. 具体方法可参见第 2.4 节中的式 (12).

2.4 区间 PSO 更新策略

2.4.1 粒子的区间适应值评价

采用第 2.3 节的方法可以为每个已知模态建立它的集成代理模型. 对于属于当前模态的任意一个粒子, 使用它的集成代理模型即可获得该粒子的预测适应值. 尽管上面的集成代理模型可以提高粒子适应值的预测精度, 但是, 受训练样本规模的限制, 其得到的预测值仍存在不确定性. 正如众所周知, 在 PSO 算法中每个粒子主要通过学习两个最优位置 (即全局和局部引导者) 来更新自身位置. 不准确的适应值会产生质量较差的全局和局部引导者, 进而误导种群向较差区域进化. 为进一步减少预测不确定性对算法搜索能力的影响, 引入区间数来表示粒子的适应值.

针对粒子 \mathbf{X}_i , 其区间适应值的预测方法如下: 首先, 判断当前粒子所处的模态, 计算当前粒子与 $Seeds$ 中精英种子的距离, 并选出距离最近的精英种子; 其次, 设置该精英种子对应的模态作为当前粒子所追随的模态; 接着, 线性集成该模态所对应的 Q 个基础代理模型; 最后, 集成模型的输出即为当前粒子的区间适应值的中值, Q 个基础代理模型预测结果的标准差即为该粒子区间适应值的半宽. 具体地, \mathbf{X}_i 的区间适应值为 $(\hat{f}_m(\mathbf{X}_i), \hat{f}_w(\mathbf{X}_i))$:

$$\begin{cases} \hat{f}_m(\mathbf{X}_i) = \sum_{j=1}^Q \frac{\hat{f}_j(\mathbf{X}_i)}{Q} \\ \hat{f}_w(\mathbf{X}_i) = \sqrt{\sum_{j=1}^Q \frac{(\hat{f}_j(\mathbf{X}_i) - \hat{f}_m(\mathbf{X}_i))^2}{Q}} \end{cases} \quad (12)$$

其中, $\hat{f}_m(\mathbf{X}_i)$ 为 Q 个基础代理模型的预测均值; $\hat{f}_w(\mathbf{X}_i)$ 为预测结果的标准差; $\hat{f}_j(\mathbf{X}_i)$ 为第 j 个基础代理模型预测得到的适应值.

当得到种群中每个粒子的区间适应值后, 即可采用第 1.3 节介绍的间数排序方法^[61] 比较粒子的优劣, 进而完成粒子全局引导者和个体引导者的更新或者选择.

2.4.2 粒子位置的更新

在更新种群时, 为发现更多的新模态, 希望保持种群的多样性; 同时, 为提高解的质量, 还希望对已知模态进行深度开采. 为此, 本节采用文献 [20] 给出的小生境机制来更新粒子的位置, 如式 (3) 所示. 该方法可以有效保证种群的多样性. 但是, 分析式 (3) 可知, 当一个粒子的个体引导者为其邻域内最优解时其集成得到的参考点 Pb_i 相对较差. 此时, 如果继续使用式 (3), 则会浪费种群计算资源. 鉴于此, 本节引入一种时变扰动机制, 用来提高算法的搜索能力. 具体地, 对于粒子 \mathbf{X}_i , 当其个体引导者 $Pbest_i$ 在其邻域内非最优时, 使用式 (3) 更新其位置; 否则, 则用式 (13) 更新其位置:

$$x_{i,d}(t+1) = Pbest_{i,d}(t) + Gaussian(0, \sigma) \quad (13)$$

$$\sigma = 0.5 \left(1 - \frac{C_{FE}}{TE_{max}} \right) + 0.01 \quad (14)$$

其中, $Gaussian(0, \sigma)$ 是均值为 0、标准差为 σ 的高斯随机数; C_{FE} 为进化过程中消耗的真实评价次数; TE_{max} 为进化过程中总的真实评价次数. 随着 C_{FE} 的增大, 种群进化代数随之增大, 种群中的粒子也逐渐向各自的模态中心靠拢, 此时应缩小式 (13) 的搜索范围, 以增大找到各模态中最优点的概率. 因此, σ 值应该随 C_{FE} 的增大而逐渐减小, 如式 (14) 所示.

2.5 模型的管理与更新

模型管理也称为填充采样, 其作用是从当前种群中选出一些代表解进行真实评价, 并使用这些解构建新样本, 用于更新代理模型. 如前所述, 设计 IMPSO-HES 算法的目的是, 同时找到问题的多个最优解. 这就要求在有限计算代价下选出的代表解要尽可能接近真实问题的每个模态的最优点. 采用动态 K-means 聚类对当前种群的 $Pbest$ 集合进行聚类, 选择每一类中的最优解作为新样本, 记新样本集为 $Sample$. 使用真实目标函数评价所选出的新样本后, 利用它们更新基础代理模型池. 需要说明的是, 为了减少频繁更新代理模型带来的计算代价问题, 并保证种群能够找到当前代理模型下问题的

最优解, 每隔 g_{\max} 次迭代进行一次模型管理. 算法 4 给出了所提模型管理和更新的伪代码.

算法 4. 模型管理与更新策略的伪代码

输入. 算法 1 中第 7 行的输出结果 $Pbest$; 数据子集 (S_1, \dots, S_K); 模型池 (M_1, \dots, M_K); 样本集合 DB.

输出. 更新后的模型池 (M_1, \dots, M_K).

- 1) 对 $Pbest$ 执行 K-means 动态聚类, 选取每个类中的最优解放入新样本集合 $Sample$;
- 2) 采用真实函数评价 $Sample$ 中保存的 L 个新样本;
- 3) 随机从 S_1, \dots, S_K 中选择 L 个数据子集; % 第 1 阶段, 更新 L 个旧的基础代理模型;
- 4) 将 $Sample$ 中的 L 个样本分别加入到选中的 L 个数据子集中;
- 5) 基于更新后的 L 个数据子集, 重新训练这些子集对应的基础代理模型;
- 6) For $i = 1:L$ % 第 2 阶段, 依次产生 L 个新的基础代理模型;
- 7) 从 DB 中选取 $2|DB|/3$ 个离 $Sample(i)$ 最近的数据, 组成新的样本子集 SL_i ;
- 8) 选择 RBFN 或 PR 作为模型, 利用 SL_i 训练一个新的基础代理模型;
- 9) 将所得新的基础代理模型加入模型池;
- 10) End for.

为在减少模型构建代价的同时增加集成模型的精度, 本节给出一种增量式的模型更新策略. 该策略分为两个阶段: 第 1 阶段, 选取当前模型池中已有基础代理模型进行重新训练; 第 2 阶段, 基于新增样本, 产生少量新的基础代理模型.

假设新增样本的数量为 L , 具体的模型更新策略如下: 1) 第 1 阶段 (算法 4 中第 3 ~ 5 行), 先从模型池中随机选出 L 个基础代理模型; 接着, 将 L 个新样本分别加入到 L 个基础代理模型的数据子集中, 形成 L 个新的数据子集; 最后, 利用新数据子集重新训练这 L 个基础代理模型. 2) 第 2 阶段 (算法 4 中第 6 ~ 10 行), 首先以 $Sample$ 中的每个新样本 $Sample(i)$ 为中心, 从 DB 中选取距离该中心最近的 $2|DB|/3$ 个数据 ($|DB|$ 表示数据库 DB 的规模), 组成一个新的数据子集. 重复上述方法, 可以得到 L 个新数据子集. 需要说明的是, 新数据子集中有 $2|DB|/3$ 个数据, 与第 2.2 节中理论样本采集数量相符. 接着, 基于这些数据子集, 选择 RBFN 或 PR 进行训练, 产生 L 个新的基础代理模型. 类似算法 2, 当 $rand < p_m$ 时, 用 RBFN 产生新的基础代理模型; 否则, 使用 PR 产生新的基础代理模型.

上述模型更新策略具有如下特点: a) 两个阶段

分别采用不同方式使用新样本, 可以有效避免新数据子集之间、新数据子集与旧数据子集之间重复. b) 两个阶段的模型更新策略具有互补性. 第 1 阶段使用新样本更新已有的基础代理模型, 可以提高这些代理模型的预测精度; 第 2 阶段使用这些代表最新模态信息的新样本构建新的基础代理模型, 能够提高集成代理模型在新模态处的预测精度. c) 此外, 无需对模型池的规模设置上限. 随着模型池的不断更新, 模型池中基础代理模型的数量会逐渐增多. 由样本产生方法可知, 每次新增的基础代理模型数与新增的样本数相同. 这表明, 新增基础代理模型的数量与进化过程中真实函数评价的次数相关, 其最大值为 TE_{\max} . 由于本文设置的 TE_{\max} 值相对较小, 故此, 无需对模型池的规模设置上限.

2.6 复杂度分析

参照文献 [25] 的方法分析 IMPSO-HES 的计算复杂度. IMPSO-HES 的计算复杂度主要由以下因素决定: 真实函数的评价、初始基础代理模型池的构建、区间 PSO 更新策略、基础代理模型选择与集成, 以及模型的管理与更新. 具体的计算复杂度表达式如下:

$$T = TE_{\max} \cdot F + T_{\text{inipool}} + \sum_{i=1}^{\text{Iter}} T_{\text{PSO}}^i + \sum_{i=1}^{\text{cycle}} (T_{\text{uppool}}^i + T_{\text{selpool}}^i + T_{\text{all}}^i) \quad (15)$$

其中, TE_{\max} 为真实函数评价的次数, F 为真实评价一个候选解的计算代价, Iter 为种群总的迭代代数, cycle 为模型更新的次数.

假设新增样本数量为 L , 种群规模为 N_p , 模型池的初始规模为 K_{ini} , 模型池的当前规模为 K , 优化问题的模态数为 N_m . 构建初始模型池的计算代价 T_{inipool} 最大为 $O(K_{\text{ini}} \times |DB|^3)$; PSO 更新一次种群的计算代价 T_{PSO} 最大为 $O(D \times N_p^2)$; 进化过程中更新一次模型池的计算代价 T_{uppool} 最大为 $O(L \times (|DB| + L)^3)$; 执行基础代理模型选择与集成策略的最大计算代价为 $T_{\text{selpool}} = O(N_m \times K^2)$. 此外, T_{all} 为所有其他附加操作的计算代价.

需要说明的是, 单个解的真实评价非常耗时, 式 (15) 中除了第 1 项外, 其余部分的运行耗时相对较小, 甚至远小于真实评价的代价. 此时, IMPSO-HES 的计算复杂度可近似为 $O(TE_{\max} \times F)$. 第 3.5 节和第 4.2 节分别提供了真实函数评价不耗时和耗时情况下 IMPSO-HES 与对比算法的运行时长, 其结果进一步证明了上述结论.

3 实验分析

为了验证所提 IMPSO-HES 算法的有效性, 将实验分为如下 4 个部分: 1) IMPSO-HES 中关键参数的取值分析; 2) IMPSO-HES 中关键策略的有效性分析; 3) 与 5 种 SAEA 进行对比, 验证所提异构集成代理模型的有效性; 4) 与 7 种传统的多模态进化优化算法进行对比, 验证所提算法处理多模态优化问题的能力. 采用 20 个广泛使用的基准问题进行测试, 并假设其为计算耗时问题. 表 1 给出了这些问题的具体特性. 问题 F1 是一个单模态优化问题^[30], 只包含一个全局最优点. 问题 F2 ~ F5 包含了一个全局和多个局部最优点^[30], 称为 1/many 问题. 该类问题用于验证算法跳出局部优化的能力. 问题 F6 ~ F20 的目标是找到尽可能多的全局/局部最优解. 其中, F6 ~ F17 为 CEC2013 基准测试函数集^[64], F18 ~ F20 为常用的几种测试函数^[19, 65]. 所有算法均采用 MATLAB R2014b 编程实现, 运行环境为 Intel Core i5, CPU 1.6 GHz, 实验结果为 30 次独立运行的统计值.

3.1 评价指标

本文采用如下 2 个指标评价算法的性能.

- 1) 全局最优解 (Global optimal solution, GS);
- 2) 有效谷的比例 (Valley ratio, VR), 用于计算算法检测到的谷点数与所有谷点数的比值.

$$VR = \frac{\sum_{i=1}^R N F V_i}{N V \times R} \quad (16)$$

其中, NV 为问题总的谷点数; $N F V_i$ 表示算法第 i 次运行找到的谷点数, R 为算法的运行次数. 针对测试问题 F6 ~ F20, 表 2 给出了计算有效谷点数时需要的幅值精度 R_v 和距离精度 R_d ^[66]. 由于各问题的复杂程度不同, 所以针对不同函数设置了不同的幅值精度 R_v 和距离精度 R_d .

3.2 参数设置

针对 IMPSO-HES, 其种群规模 N_p 设为 100; 模型更新频率 g_{\max} 设置为 6; 集成时所选基础代理模型的规模设置为 $Q = \text{ceil}(K/4)$. 设置最大真实评价次数作为算法的终止条件. 对于 F1 ~ F5, 最大真实评价次数设为 $8D$, 动态 K-means 中的聚类数目随机取自 $\{2, 3, 4\}$; 对于 F6 ~ F20, 最大真实评价次数设为 $3D + 100$, 动态 K-means 中的聚类数目随机取自 $\{2, 3, 4, 5, 6, 7\}$. 依据文献 [20] 的建议, 根据进化代数依次设置邻里规模 n_{size} 为 $\{2, 3, 4,$

表 1 基准问题
Table 1 Benchmark problems

问题	测试函数	维数	变量空间	全局/局部解个数	全局最优解的目标值
F1	Ellipsoid	10/20	$\mathbf{X} \in [-1, 1]^D$	1/0	0
F2	Ackley	10/20	$\mathbf{X} \in [-30, 30]^D$	1/many	0
F3	Rastrigin	10/20	$\mathbf{X} \in [-5.12, 5.12]^D$	1/many	0
F4	Rosenbrock	10/20	$\mathbf{X} \in [-2.048, 2.048]^D$	1/many	0
F5	Griewank	10/20	$\mathbf{X} \in [-600, 600]^D$	1/many	0
F6	Reverse five-uneven-peak trap	1	$\mathbf{X} \in [0, 30]$	2/3	-200
F7	Reverse equal maxima	1	$\mathbf{X} \in [0, 1]$	5/0	-1
F8	Reverse uneven decreasing maxima	1	$\mathbf{X} \in [0, 1]$	1/4	-1
F9	Reverse himmelblau	2	$\mathbf{X} \in [-6, 6]^D$	4/0	-200
F10	Six-hump camel	2	$x_1 \in [-1.9, 1.9], x_2 \in [-1.1, 1.1]$	2/2	-1.031 6
F11	Reverse shubert	2	$\mathbf{X} \in [-10, 10]^D$	18/many	-186.73
F12	Reverse vincent	2	$\mathbf{X} \in [0.25, 10]^D$	36/0	-1
F13	Reverse modified rastrigin	2	$\mathbf{X} \in [0, 1]^D$	12/0	2
F14	Reverse CF1	2	$\mathbf{X} \in [-5, 5]^D$	6/0	0
F15	Reverse CF2	2	$\mathbf{X} \in [-5, 5]^D$	8/0	0
F16	Reverse CF3	2	$\mathbf{X} \in [-5, 5]^D$	6/0	0
F17	Reverse CF4	3	$\mathbf{X} \in [-5, 5]^D$	8/0	0
F18	UrsemF4 back	2	$\mathbf{X} \in [-2, 2]^D$	2/0	-0.267 9
F19	Branin RCOS	2	$x_1 \in [-5, 10], x_2 \in [0, 15]$	3/0	0.397 8
F20	Waves	2	$x_1 \in [-0.9, 1.2], x_2 \in [-1.2, 1.2]$	1/9	-7.776

表 2 F6 ~ F20 的幅值精度和距离精度
Table 2 Amplitude accuracy and distance accuracy for F6 ~ F20

	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
R_v	1	0.05	0.1	0.5	0.05	10	0.1	0.5	1	1	1	1	0.1	0.1	0.5
R_d	1	0.05	0.5	0.5	0.2	2	0.5	0.5	1	1	1	1	0.5	1	0.2

5}; 设置式 (3) 中 $w = 0.7298$.

3.3 关键参数分析

选择问题 F5、F9、F10 和 F12 作为代表, 分析模型更新频率 g_{max} 和集成规模 Q 取值对算法性能的影响. 采用 Mann-Whitney 非参数检验, 验证 IMPSO-HES 与对比算法之间性能差异的显著性, 显著性水平取 0.05. 表 3 ~ 12 中, “+”和“-”分别表示 IMPSO-HES 明显优于和劣于对比算法, “=”表示两者无明显差别.

3.3.1 参数 g_{max}

依次设置模型更新频率 g_{max} 为 {3, 6, 9}, 表 3 给出了不同 g_{max} 取值下 IMPSO-HES 所得结果. 实验以 $g_{max} = 6$ 时 IMPSO-HES 的结果作为基础, 计算三种取值下 IMPSO-HES 所得结果之间的显著差异性.

表 3 不同 g_{max} 取值下 IMPSO-HES 所得的性能指标值
Table 3 Performance values obtained by IMPSO-HES under different g_{max} values

问题	g_{max}	GS 均值 (标准差)	VR 均值	耗时 (s)
F5 ($D = 10$)	3	3.800 7 (3.5E+00)+	—	64
	6	1.174 5 (3.7E-02)	—	85
	9	1.108 3 (2.5E-02) =	—	116
F5 ($D = 20$)	3	8.198 0 (9.8E+00) +	—	776
	6	1.075 7 (1.6E-02)	—	1 400
	9	0.807 9 (2.8E-01) -	—	2 045
F9	3	-199.93 (3.1E-03) =	0.68	11
	6	-199.99 (1.0E-04)	0.70	19
	9	-200.00 (1.4E-03) =	0.63	36
F10	3	-1.031 6 (1.7E-06) =	1.00	19
	6	-1.031 6 (9.8E-07)	1.00	28
	9	-1.031 6 (9.8E-07) =	1.00	38
F12	3	-0.999 0 (7.1E-06) =	0.13	10
	6	-0.999 9 (1.0E-06)	0.13	14
	9	-0.999 9 (2.2E-06) =	0.11	25

注: 加粗字体表示各组的最优结果值.

可以看出: 1) 对于所有 4 个测试问题, 当 $g_{max} = 9$ 时 IMPSO-HES 获得的 GS 值最好; 2) 对于 F9、F10 和 F12, 当 $g_{max} = 3$ 或 $g_{max} = 6$ 时, 算法获得的 VR 值无明显差异, 但要优于 $g_{max} = 9$ 时的结果;

3) g_{max} 取值越大, 算法的运行时间越长. 综上可见, 设置 $g_{max} = 6$ 较为合适.

3.3.2 参数 Q

依次设置集成规模 Q 为 { $K/5, K/4, K/3, K/2$ }, 表 4 给出了不同 Q 值下 IMPSO-HES 所得结果. 实验以 $Q = K/4$ 时 IMPSO-HES 的结果作为基础, 计算 4 种取值下 IMPSO-HES 所得结果之间的显著差异性.

表 4 不同 Q 取值下 IMPSO-HES 所得的性能指标值
Table 4 Performance values obtained by IMPSO-HES under different Q values

问题	Q	GS 均值 (标准差)	VR 均值	耗时 (s)
F5 ($D = 10$)	$K/5$	1.658 1 (2.2E-01) +	—	64
	$K/4$	1.174 5 (3.7E-02)	—	85
	$K/3$	1.382 1 (1.5E-01) +	—	108
	$K/2$	1.269 6 (5.1E-02) +	—	160
F5 ($D = 20$)	$K/5$	1.980 0 (1.0E+00) +	—	1137
	$K/4$	1.075 7 (1.6E-02)	—	1400
	$K/3$	1.832 1 (1.1E+00) +	—	1920
	$K/2$	1.835 2 (1.7E+00) +	—	2700
F9	$K/5$	-199.98 (7.2E-04) =	0.53	17
	$K/4$	-199.99 (1.0E-04)	0.70	19
	$K/3$	-199.98 (4.6E-04) =	0.55	24
	$K/2$	-199.14 (6.8E+00) +	0.33	34
F10	$K/5$	-1.031 6 (1.1E-09) =	1.00	28
	$K/4$	-1.031 6 (9.8E-07)	1.00	28
	$K/3$	-1.031 6 (9.8E-07) =	1.00	30
	$K/2$	-1.030 0 (1.4E-03) +	0.85	48
F12	$K/5$	-0.999 1 (2.3E-06) +	0.12	12
	$K/4$	-0.999 9 (1.0E-06)	0.13	14
	$K/3$	-0.999 6 (8.5E-07) +	0.10	18
	$K/2$	-0.994 9 (9.2E-05) +	0.10	24

注: 加粗字体表示各组的最优结果值.

可以看出: 1) 对于所有 4 个测试问题, 当 $Q = K/4$ 时 IMPSO-HES 获得的 GS 值优于其他取值情况; 2) 对于问题 F9、F10 和 F12, 当 $Q = K/4$ 时 IMPSO-HES 获得的 VR 值不劣于其余取值时的结果; 3) Q 取值越大, 算法的运行时间越长. 综上可见, 设置 $Q = K/4$ 较合适.

3.4 所提关键策略分析

3.4.1 异构集成策略的有效性分析

1) 异构集成与同质集成的比较

与同质集成学习策略对比, 验证所提异构集成策略的优越性. 表 5 给出了异构集成策略和同质集成策略下本文算法所得结果的各项性能指标. 表 5 中, “IMPSON-PR”表示本文算法采用 PR 作为基础代理模型的情况, “IMPSON-RBFN”表示本文算法采用 RBFN 作为基础代理模型的情况. 可以看出: a) 对于测试问题 F5 和 F12, 异构集成下 IMPSON-HES 的 GS 值显著优于“IMPSON-PR”和“IMPSON-RBFN”; 对于 F5, “IMPSON-PR”获得的 GS 值显著优于“IMPSON-RBFN”; 对于 F9 和 F10, 异构集成下 IMPSON-HES 获得的 GS 值显著优于“IMPSON-PR”的结果, 但与“IMPSON-RBFN”的结果无显著差异. b) 对于测试问题 F9、F10 和 F12, 相对其他两种方法, IMPSON-HES 获得了最优的 VR 值. c) 对于测试问题 F5, “IMPSON-PR”策略的运行时间最长, “IMPSON-RBFN”策略的运行时间最短; 对于其余问题, 三种策略的运行时间差异不大. 综上所述, 相对上述同构集成策略, 异构集成可以有效结合各类基础模型的优点.

表 5 异构集成与同质集成下 IMPSON-HES 所得结果
Table 5 Performance values obtained by IMPSON-HES under heterogeneous and homogeneous ensemble

问题	算法	GS 均值 (标准差)	VR 均值	耗时 (s)
F5 ($D = 10$)	IMPSON-PR	1.631 0 (7.1E-01) +	—	86
	IMPSON-RBFN	45.27 2 (8.9E+02) +	—	39
	IMPSON-HES	1.174 5 (3.7E-02)	—	85
F5 ($D = 20$)	IMPSON-PR	2.003 7 (2.9E+00) +	—	1 478
	IMPSON-RBFN	116.7 8 (9.5E+02) +	—	180
	IMPSON-HES	1.075 7 (1.6E-02)	—	1 400
F9	IMPSON-PR	-196.81 (9.5E+00) +	0.05	16
	IMPSON-RBFN	-199.99 (4.7E-07) =	0.65	22
	IMPSON-HES	-199.99 (1.0E-04)	0.70	19
F10	IMPSON-PR	-0.962 0 (2.5E-03) +	0.2	17
	IMPSON-RBFN	-1.031 6 (9.8E-09) =	1.00	20
	IMPSON-HES	-1.031 6 (9.8E-07)	1.00	28
F12	IMPSON-PR	-0.988 6 (1.5E-04) +	0.06	11
	IMPSON-RBFN	-0.999 5 (9.4E-07) +	0.09	19
	IMPSON-HES	-0.999 9 (1.0E-06)	0.13	14

注: 加粗字体表示各组的最佳结果值.

2) 自适应更新概率 p_m 的有效性分析

为使模型池中的基础代理模型更适于被优化问题, 算法 3 引入了一个自适应更新概率 p_m , 来自

主调整选择 RBFN 或 PR 的概率. 通过与固定 p_m 值进行对比, 验证所提自适应更新概率的有效性. 表 6 给出了不同更新概率 p_m 下 IMPSON-HES 所得优化结果. 表 6 中, “固定”表示采用 RBFN 或 PR 产生基础代理模型的概率值始终不变, 即 $p_m = 0.5$; “自适应”代表算法 3 给出的自适应更新概率. 可以看出: a) 除了问题 F10 外, 在“自适应”策略的帮助下, IMPSON-HES 均获得了显著优于“固定”策略的 GS 和 VR 值; b) 两种策略下 IMPSON-HES 的运行时间差别不大. 综上所述, 在一定程度上, 采用自适应方法更新代理模型可以提高集成模型的精度.

表 6 不同更新概率 p_m 下 IMPSON-HES 所得结果
Table 6 Performance values obtained by IMPSON-HES under different p_m values

问题	p_m	GS 均值 (标准差)	VR 均值	耗时 (s)
F5 ($D = 10$)	固定	1.439 3 (3.8E-01) +	—	84
	自适应	1.174 5 (3.7E-02)	—	85
F5 ($D = 20$)	固定	1.750 3 (1.7E+00) +	—	1313
	自适应	1.075 7 (1.6E-02)	—	1400
F9	固定	-199.91 (2.6E-02) +	0.40	19
	自适应	-199.99 (1.0E-04)	0.70	19
F10	固定	-1.031 6 (4.7E-08) =	1.00	26
	自适应	-1.031 6 (9.8E-07)	1.00	28
F12	固定	-0.996 9 (4.8E-05) +	0.12	14
	自适应	-0.999 9 (1.0E-06)	0.13	14

注: 加粗字体表示各组的最佳结果值.

3) 所提基础代理模型选择策略的有效性分析

为使每个粒子选择更适合自己的基础代理模型进行集成, 第 2.3 节中提出了基于模态匹配的基础代理模型选择集成策略. 与使用全部基础代理模型进行集成的策略 (记为 All-S) 进行对比, 验证所提代理模型选择策略 (记为 Mod-S) 的有效性. 表 7 给出了分别使用 All-S 和 Mod-S 时 IMPSON-HES 所得的结果. 可以看出: a) 对于所有问题, Mod-S 策略获得的 GS 和 VR 值均优于 All-S 获得的结果; b) All-S 策略的运行时间远大于 Mod-S 策略. 可见, 根据问题的模态位置, Mod-S 可以为每个粒子分配合适的基础代理模型.

3.4.2 区间 PSO 更新策略的有效性分析

1) 区间适应值评价策略分析

为进一步减少预测不确定性对算法搜索能力的影响, 第 2.4 节引入了区间数来表示粒子的适应值. 通过与传统的精确适应值评价策略进行对比, 验证区间适应值评价策略的有效性. 图 2 和图 3 分别给出了两种适应值评价策略下 IMPSON-HES 所得到

表 7 使用 All-S 和 Mod-S 时 IMPSO-HES 所得结果
Table 7 Performance values obtained by IMPSO-HES with All-S and Mod-S

问题	集成策略	GS 均值 (标准差)	VR 均值	耗时 (s)
F5 ($D = 10$)	All-S	3.878 5 (3.8E+00) +	—	243
	Mod-S	1.174 5 (3.7E-02)	—	85
F5 ($D = 20$)	All-S	8.838 7 (8.1E+00) +	—	3 362
	Mod-S	1.075 7 (1.6E-02)	—	1 400
F9	All-S	-187.33 (2.0E+2) +	0.05	80
	Mod-S	-199.99 (1.0E-04)	0.70	19
F10	All-S	-0.9751 (1.4E-02) +	0.70	57
	Mod-S	-1.031 6 (9.8E-07)	1.00	28
F12	All-S	-0.973 7 (1.9E-02) +	0.08	42
	Mod-S	-0.999 9 (1.0E-06)	0.13	14

注: 加粗字体表示各组的最优结果值。

的优化结果. 图中, “Precision”表示精确适应值评价策略; “Interval”表示区间适应值评价策略. 图 2(a)中前两个箱图为 F5 ($D = 10$) 时的 GS 值, 后两个箱图为 F5 ($D = 20$) 时的 GS 值. 可以看出, 对于问题 F5 和 F10, 采用“Interval”时 IMPSO-HES 获得的 GS 值要优于采用“Precision”时获得的结果; 对于 F9 和 F12, 两种策略获得的 GS 值无显著差异. 从图 3 可以看出, 对于 F9 和 F12, 采用“Interval”时 IMPSO-HES 获得的 VR 值均高于采用“Precision”时的结果. 可见, 采用区间适应值评价策略可以在一定程度上提高算法的搜索性能.

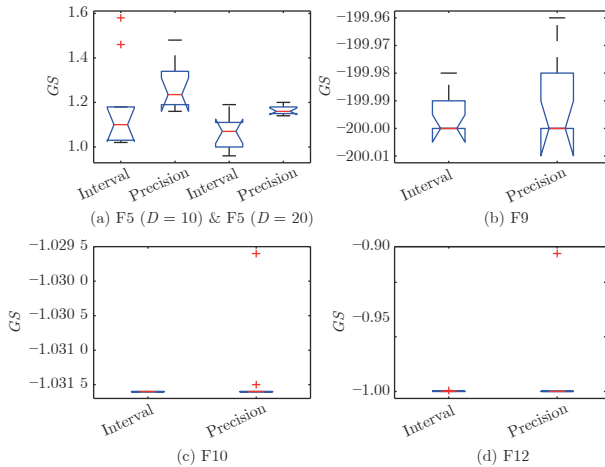


图 2 精确评价和区间评价策略下 IMPSO-HES 所得 GS 值

Fig.2 GS values obtained by IMPSO-HES under precision and interval evaluation

2) PSO 自适应位置更新策略的有效性分析

为进一步提高算法的搜索能力, 第 2.4 节引入了一种时变扰动机制. 通过与不带时变扰动机制的

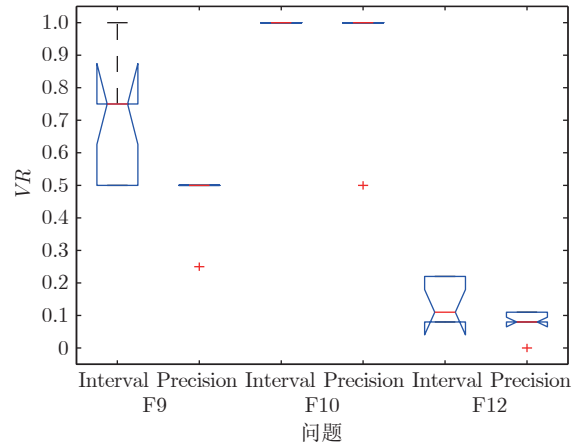


图 3 精确评价和区间评价策略下 IMPSO-HES 所得 VR 值

Fig.3 VR values obtained by IMPSO-HES under precision and interval evaluation

IMPSO-HES 进行对比, 验证该扰动机制的有效性. 便于表述, 记不带时变扰动机制的 IMPSO-HES 为 IMPSO-HES/D. 图 4 和图 5 给出了 IMPSO-HES/D 和 IMPSO-HES 得到的优化结果. 图中, 横坐标“1”代表 IMPSO-HES/D, “2”代表 IMPSO-HES. 从图 4 可以看出, 对于 F5、F9 和 F12, 在所提扰动机制的帮助下, IMPSO-HES 获得了显著优于 IMPSO-HES/D 的 GS 值; 对于 F10, 两种算法获得的 GS 值无显著差异. 从图 5 可以看出, 对于 F9、F10 和 F12, IMPSO-HES 所得 VR 值皆不劣于 IMPSO-HES/D 的结果. 这些结果都证明了本文所提时变扰动机制的有效性.

3.4.3 增量式模型更新策略的有效性分析

为节约模型的构建时间, 第 2.5 节中引入了一

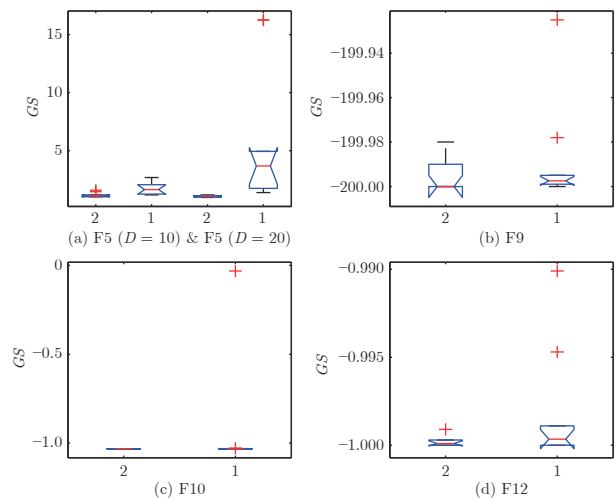


图 4 IMPSO-HES/D 和 IMPSO-HES 得到的 GS 值

Fig.4 GS values obtained by IMPSO-HES/D and IMPSO-HES

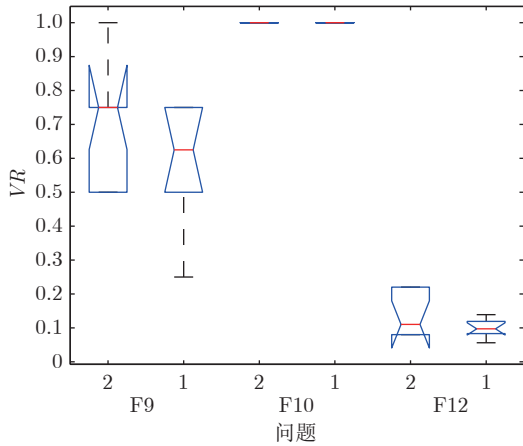


图 5 IMPSO-HES/D 和 IMPSO-HES 得到的 VR 值

Fig.5 VR values obtained by IMPSO-HES/D and IMPSO-HES

种增量式模型更新策略. 与更新所有基础代理模型的策略 (记为 All-up) 进行对比, 验证所提增量式模型更新策略 (记为 Inc-up) 的有效性. 表 8 给出了不同模型更新策略下 IMPSO-HES 所得结果的各项性能指标. 可以看出: 1) 对于测试问题 F5, Inc-up 策略获得的 GS 值显著优于 All-up 策略的结果; 对于其余 3 个问题, 两种策略获得的 GS 值无显著差异. 2) 对于 F9、F10 和 F12, Inc-up 策略获得的 VR 值略优于 All-up 策略的结果. 3) All-up 策略的运行时间长于 Inc-up 策略. Inc-up 优于 All-up 的原因可能在于, Inc-up 可以及时为新发现的模态构建基础代理模型, 从而提高了集成代理模型对各模态的拟合精度.

3.5 与 5 种代理辅助进化优化算法的对比

为了验证 IMPSO-HES 获得全局最优解的能

表 8 不同模型更新策略下 IMPSO-HES 所得结果

Table 8 Performance values obtained by IMPSO-HES under different model update strategies

问题	更新策略	GS 均值 (标准差)	VR 均值	耗时 (s)
F5 ($D = 10$)	All-up	1.500 9 (3.9E-02) +	—	97
	Inc-up	1.174 5 (3.7E-02)	—	85
F5 ($D = 20$)	All-up	32.184 (2.4E+04) +	—	1 509
	Inc-up	1.075 7 (1.6E-02)	—	1 400
F9	All-up	-200.00 (3.6E-10) =	0.63	30
	Inc-up	-199.99 (1.0E-04)	0.70	19
F10	All-up	-1.031 6 (1.2E-04) =	0.95	30
	Inc-up	-1.031 6 (9.8E-07)	1.00	28
F12	All-up	-0.999 8 (2.7E-07) =	0.11	16
	Inc-up	-0.999 9 (1.0E-06)	0.13	14

注: 加粗字体表示各组的最佳结果值.

力, 将其与 5 种典型的 SAEA 进行比较. 这 5 种对比算法分别如下.

1) CAL-SAPSO^[30], 该算法混合使用 Kriging、RBF 和 PR 预测粒子的适应值;

2) SA-COSO^[56], 该算法混合使用 RBFN 和适应值继承策略预测粒子的适应值;

3) Gr-based SAPSO^[57], 该算法混合使用 GP 和适应值继承策略预测粒子的适应值;

4) PESPSO 和 ESPSO^[40], 这两种算法同时使用全局和局部代理模型预测粒子的适应值.

为保证公平性, 所有算法采用相同的真实函数评价次数. CAL-SAPSO 的初始 DB 规模与本文算法保持一致, 其余参数采用原文设置^[30]. 其余 4 种算法均基于初始种群建立初始代理模型, 其中, PESPSO 和 ESPSO 的参数均采用原文设置. 受最大真实函数评价次数的限制, 当 $D \leq 10$ 时, Gr-based SAPSO 的种群规模设置为 60; 当 $D \leq 10$ 和 $D > 10$ 时, SA-COSO 中 SL-PSO 的种群规模分别设置为 30 和 80, 其余参数采用相应的原文设置.

表 9 展示了 IMPSO-HES 与 5 种对比算法所得的 GS 值. 进一步, 表 10 给出了表 9 数据的统计分析结果. 表 10 中的“好/平/差”统计了 IMPSO-HES 与对比算法在 F2 ~ F20 测试问题上的对比结果. 其中, “好”、“平”和“差”分别表示 IMPSO-HES 优于、类似和劣于对比算法的函数个数; “Rank”表示各算法的排序值. 同时, 使用显著性水平为 0.05 的 Friedman 非参数检验方法^[67], 其中 IMPSO-HES 为控制方法. “Adjusted p -value”表示 IMPSO-HES 与对比算法之间的显著性差异水平^[68].

问题 F1 为单模态优化问题. 从表 9 可以看出, 在处理该问题时, IMPSO-HES 的性能要略低于 5 种对比算法. 主要原因是, 本文算法采取的多模态处理方式会引导种群进行分散搜索, 降低了算法对该问题唯一全局最优解的开发能力. 这在一定程度上也表明了本文所提搜索策略同时处理多模态的潜力.

问题 F2 ~ F5 为 1/many 问题. 从表 10 中的“好/平/差”可以看出, IMPSO-HES 分别在 5、3、4、4 和 4 个测试函数上占优了对比算法 SA-COSO、CAL-SAPSO、Gr-based SAPSO、PESPSO 和 ESPSO; 同时, 其劣于对比算法的测试函数数量分别为 1、1、2、2 和 2. 从“Rank”值可以看出, IMPSO-HES 的排序最优. 从“Adjusted p -value”可以看出, IMPSO-HES 显著优于 SA-COSO, 且与其余 4 种对比算法无显著差异. 综上可见, 对于 1/many 类问题, IMPSO-HES 可以获得与典型 SAEAs 相竞争的优化结果.

表 9 IMPSO-HES 与 5 种 SAEA 所得 GS 值 (均值 (方差))
Table 9 GS values obtained by IMPSO-HES and 5 SAEAs (mean (variance))

问题	D	IMPSON-HES	SA-COSO	CAL-SAPSO	Gr-based SAPSO	PESPSO	ESPSO
F1	10	3.660 0 (4.2E+00)	3.160 0- (6.5E-02)	0.115 3- (4.9E-02)	0.147 6- (1.1E-03)	0.296 2- (1.3E-03)	0.664 5- (5.0E-02)
	20	21.398 (6.1E+01)	11.017- (1.2E+01)	0.229 2- (1.9E-02)	0.027 9- (8.2E-06)	1.377 0- (1.2E-01)	1.866 4- (2.4E-01)
F2	10	17.990 (1.1E+00)	17.248= (4.1E-02)	18.606+ (4.8E-01)	15.910- (6.4E-01)	11.820- (4.3E+00)	13.786- (2.0E+00)
	20	18.866 (9.0E-01)	18.025- (4.4E-01)	18.421= (2.4E+00)	14.717- (1.1E+00)	12.584- (2.3E+01)	15.958- (1.6E+01)
F3	10	78.266 (1.3E+02)	97.683+ (5.8E+02)	79.727= (1.6E+03)	94.349+ (7.3E+01)	82.325= (1.2E+02)	89.952= (2.0E+02)
	20	173.97 (2.4E+02)	177.43= (6.6E+02)	128.71- (4.0E+03)	168.14= (1.6E+02)	173.99= (1.7E+02)	175.65= (1.1E+02)
F4	10	37.310 (1.1E+02)	537.31+ (2.4E+04)	39.003= (2.0E+02)	173.66+ (3.3E+02)	90.531+ (6.7E+02)	66.581+ (1.0E+02)
	20	41.469 (5.7E+02)	891.97+ (1.7E+04)	42.758= (2.0E+02)	330.37+ (3.9E+03)	97.508+ (6.8E+02)	195.90+ (1.9E+03)
F5	10	1.174 5 (3.7E-02)	66.556+ (1.8E+02)	1.736 4+ (1.4E-01)	1.310 6+ (1.7E-02)	2.798 7+ (2.4E+00)	2.317 2+ (3.9E-01)
	20	1.075 7 (1.6E-02)	43.897+ (1.9E+02)	2.255 3+ (3.2E-01)	1.057 2= (2.0E-05)	6.701 8+ (7.4E+00)	10.373+ (6.2E+00)
F6	1	-199.15 (4.6E+00)	-200.00- (2.1E-10)	-200.00- (1.6E-09)	-190.91+ (3.2E+01)	-200.00- (1.2E-13)	-200.00- (1.0E-11)
F7	1	-0.999 9 (3.1E-06)	-1.00= (0.0E+00)	-0.505 2+ (1.2E-01)	-0.999 1+ (1.1E-07)	-0.999 9= (2.7E-05)	-0.999 8= (3.8E-06)
F8	1	-0.985 4 (1.3E-05)	-0.980 8= (1.0E-10)	-0.511 4+ (8.0E-02)	-0.944 7+ (7.4E-04)	-0.948 6+ (5.1E-04)	-0.948 6+ (5.1E-04)
F9	2	-199.99 (1.0E-04)	-196.14+ (3.8E+01)	-157.69+ (8.6E+02)	-199.93+ (5.1E-04)	-199.98= (2.7E-04)	-199.74+ (6.4E-03)
F10	2	-1.031 6 (9.8E-07)	-0.995 6+ (1.6E-03)	-0.464 6+ (1.3E-01)	-1.030 6+ (1.9E-06)	-1.030 3+ (1.7E-07)	-1.029 2+ (5.3E-07)
F11	2	-158.32 (1.9E+03)	-89.368+ (2.4E+03)	-52.464+ (2.6E+03)	-113.85+ (3.5E+04)	-130.53+ (2.5E+03)	-94.463+ (1.5E+03)
F12	2	-0.999 9 (1.0E-06)	-0.979 8+ (5.6E-04)	-0.719 4+ (9.0E-02)	-0.984 5+ (1.9E-04)	-0.995 4+ (2.0E-06)	-0.980 0+ (5.5E-05)
F13	2	2.232 9 (2.3E-01)	2.890 3+ (6.4E-02)	7.846 7+ (3.0E+01)	2.298 5= (1.0E-01)	2.022 8- (4.6E-03)	2.060 9- (3.1E-03)
F14	2	0.087 9 (5.0E-01)	40.011+ (2.6E+02)	197.39+ (9.2E+03)	23.774+ (6.3E+03)	7.588 4+ (1.1E+02)	9.961 7+ (3.0E+02)
F15	2	36.423 (3.7E+03)	89.091+ (2.7E+02)	183.14+ (3.6E+03)	80.557+ (1.1E+03)	26.116= (7.6E+02)	57.889+ (2.8E+03)
F16	2	0.242 3 (1.3E-01)	90.430+ (1.2E+04)	350.88+ (4.8E+04)	60.296+ (3.2E+03)	1.162 1+ (2.5E+00)	18.280+ (1.2E+03)
F17	3	32.566 (2.0E+04)	88.270+ (5.3E+02)	173.56+ (2.6E+04)	57.380+ (2.1E+03)	26.079= (6.2E+02)	37.233= (6.0E+02)

表 9 IMPSO-HES 与 5 种 SAEA 所得 GS 值 (均值 (方差)) (续表)Table 9 GS values obtained by IMPSO-HES and 5 SAEAs (mean (variance)) (continued table)

问题	D	IMPSON-HES	SA-COSO	CAL-SAPSO	Gr-based SAPSO	PESPSO	ESPSO
F18	2	-0.267 9 (1.6E-06)	-0.245 7+ (3.6E-04)	-0.130 4+ (5.6E-03)	-0.267 1+ (6.8E-08)	-0.267 8= (1.6E-06)	-0.267 8= (5.4E-09)
F19	2	0.399 9 (2.4E-05)	1.148 8+ (8.6E-01)	2.260 3+ (6.2E+00)	0.425 9+ (1.3E-03)	0.424 9+ (1.2E-03)	0.513 6+ (5.3E-02)
F20	2	-7.429 9 (1.7E-02)	-7.776 0- (0.0E+00)	-7.775 3- (4.2E-06)	-6.340 8+ (8.4E-01)	-7.294 3+ (2.2E-01)	-7.451 1= (2.7E-01)

注: 加粗字体表示各行 GS 值的最优结果值。

表 10 基于表 9 的统计结果

Table 10 Statistical results based on Table 9

问题		IMPSON-HES	SA-COSO	CAL-SAPSO	Gr-based SAPSO	PESPSO	ESPSO
F2 ~ F5	好/平/差	—	5/2/1	3/4/1	4/2/2	4/2/2	4/2/2
	Rank	2.500 0	5.500 0	3.000 0	3.125 0	3.125 0	3.750 0
	Adjusted p -value	—	0.006 6	0.689 2	0.689 2	0.689 2	0.393 8
F6 ~ F20	好/平/差	—	11/2/2	13/0/2	14/1/0	8/5/2	9/4/2
	Rank	1.833 3	4.166 6	5.433 3	4.000 0	2.266 6	3.300 0
	Adjusted p -value	—	0.001 6	0.000 0	0.002 5	0.525 8	0.039 5

注: 加粗字体表示各组的最佳结果值。

问题 F6 ~ F20 为具有多个全局最优解的多模态优化问题。从“好/平/差”值可以看出, 在 15 个测试函数中, IMPSO-HES 占优对比算法 SA-COSO、CAL-SAPSO、Gr-based SAPSO、PESPSO 和 ESPSO 的测试函数数量分别为 11、13、14、8 和 9, 而其劣于对比算法的问题数量分别为 2、2、0、2 和 2。由“Rank”值可知, IMPSO-HES 的排序值为 1.833 3, 小于其他 5 种对比算法。从“Adjusted p -value”可以看出, 除了 PESPSO 之外, IMPSO-HES 显著优于其余 4 种对比算法。综上所述, 对于具有多个全局最优解的多模态优化问题, IMPSO-HES 在获取多个全局最优解的同时, 也可以保证所得全局最优解的质量。其原因可能是, 这些 SAEA 没有处理多模态的机制。受多个全局最优解的影响, 在进化过程中种群会频繁切换搜索区域, 进而影响算法的搜索效率。

图 6 给出了 IMPSO-HES 与对比算法的运行时间, 图 6(a) 为各算法优化 F1 ~ F5 所需时间, 图 6(b) 为各算法优化 F6 ~ F20 所需时间。可以看出: 1) 维数越高, IMPSO-HES 的运行时间越长; 2) 对于 F1 ~ F5, IMPSO-HES 和 CAL-SAPSO 的运行时间明显大于其余 4 种算法; 3) 对于 F6 ~ F20, CAL-SAPSO 运行时间最长, IMPSO-HES 的运行时间略长于其他 4 种对比算法。IMPSO-HES 运行时间较长的原因可能在于, 它需要建立大量的基础代理模型。虽

然 IMPSO-HES 运行时间长于部分对比算法, 但是, 与算法花费在昂贵真实函数评价上的代价相比 (如第 4.2 节所示), 其运行代价仍然相对较小, 甚至可以忽略。

3.6 与 7 种多模态进化优化算法的对比

为了验证 IMPSO-HES 同时定位多个优化解的能力, 将其与 7 种经典多模态进化算法进行比较。这些比较算法包括 3 种基于 PSO 的多模态优化算法 (即 R3PSO、FERPSO^[15] 和 LIPS^[20])、3 种基于差分进化的多模态优化算法 (即 NCDE、NSDE^[66] 和 ANDE^[11]) 和 1 种多目标进化算法 (即 EMO-MMO^[9])。为了保证对比的公平性, 所有算法采用相同的最大真实函数评价次数 TE_{max} 。由于这些对比算法均采用真实函数评价所有的个体, 因此, 为确保每个算法都可以执行多次迭代, 实验调小了它们的种群规模。对于 7 种对比算法, 将它们的种群规模分别设为 20 和 50。实验结果发现, 当种群规模为 20 时这些算法的综合效果更佳。由于篇幅限制, 此处不提供对比算法的调参过程。主要原因可能是: 在种群规模为 20 时这些算法的进化代数相对较大, 种群有较为充分的时间收敛。因此, 所有对比算法的种群规模均设置为 20。对于这些对比算法的其他参数, 均按照原文的建议进行设置。

表 11 展示了 IMPSO-HES 与 7 种多模态优化

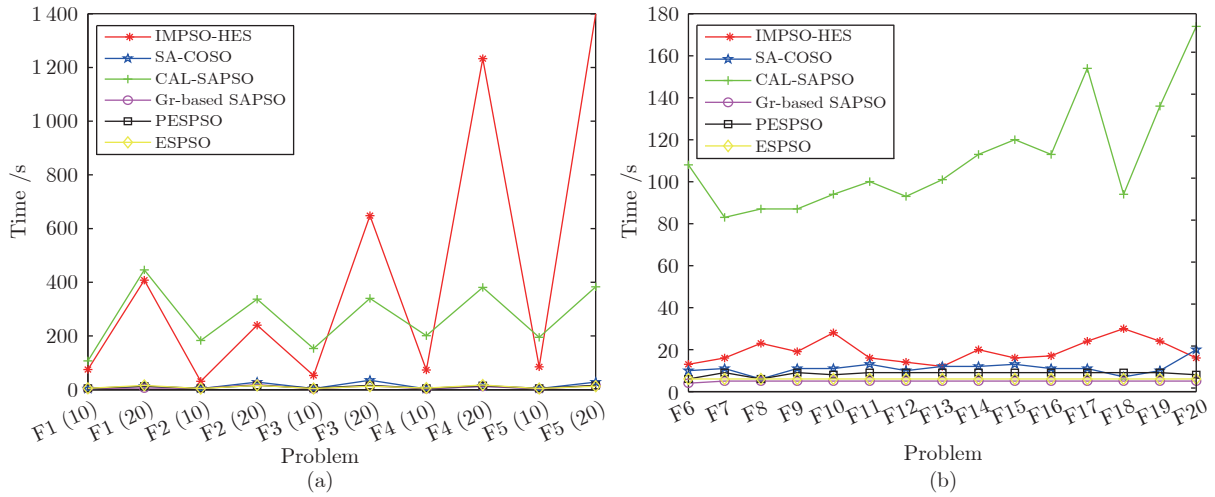


图 6 IMPSO-HES 与 5 种 SAEA 的运行耗时
Fig.6 Running times of IMPSO-HES and the five SAEAs

表 11 处理 F1 ~ F5 时 IMPSO-HES 与 7 种多模态进化算法所得 GS 值 (均值 (方差))
Table 11 GS values obtained by IMPSO-HES and the 7 multimodal EAs on F1 ~ F5 (mean (variance))

问题	D	IMPSO-HES	LIPS	EMO-MMO	R3PSO	FERPSO	NCDE	NSDE	ANDE
F1	10	3.6600 (4.2E+00)	3.3110- (7.8E-01)	5.0580+ (1.3E+00)	5.9282+ (2.3E+00)	4.3713+ (1.2E+00)	5.7227+ (6.4E+00)	5.8277+ (1.6E+00)	5.2888+ (2.6E+00)
	20	21.398 (6.1E+01)	19.528= (9.8E+00)	26.709+ (2.2E+01)	31.059+ (2.2E+01)	18.792- (1.2E+01)	28.868+ (5.8E+01)	29.060+ (1.5E+01)	32.311+ (5.5E+01)
F2	10	17.990 (1.1E+00)	18.046= (8.1E-01)	18.022= (7.0E-01)	19.159+ (3.9E-01)	18.073= (1.06E+00)	19.411+ (1.3E+00)	19.432+ (3.0E-01)	19.523+ (1.5E-01)
	20	18.866 (9.0E-01)	18.924= (3.6E+01)	18.922= (1.7E-01)	19.663+ (6.5E-02)	19.313+ (2.5E-01)	19.895+ (9.9E-02)	20.108+ (4.9E-02)	19.950+ (8.2E-06)
F3	10	78.266 (1.3E+02)	95.069+ (6.3E+01)	89.325= (1.2E+02)	108.58+ (2.2E+02)	100.83+ (8.2E+01)	110.95+ (5.5E+02)	101.33+ (1.3E+02)	106.90+ (1.3E+02)
	20	173.97 (2.4E+02)	212.48+ (2.6E+02)	207.09+ (2.8E+02)	258.90+ (3.3E+02)	225.25+ (5.1E+02)	251.77+ (3.2E+02)	262.26+ (6.5E+02)	268.57+ (1.1E+02)
F4	10	37.310 (1.1E+02)	343.96+ (4.2E+05)	257.96+ (3.6E+05)	670.32+ (1.3E+05)	451.41+ (2.8E+04)	812.90+ (1.0E+05)	982.18+ (1.1E+05)	523.1+ (2.7E+05)
	20	41.469 (5.7E+02)	1431.9+ (1.1E+05)	1399.6+ (1.5E+05)	2853.3+ (3.6E+05)	1722.6+ (5.1E+04)	3031.2+ (6.9E+05)	2737.0+ (7.1E+05)	2416.1+ (1.6E+05)
F5	10	1.1745 (3.7E-02)	66.246+ (3.1E+02)	65.750+ (6.7E+02)	94.936+ (4.7E+02)	71.342+ (4.8E+02)	129.69+ (3.5E+02)	115.66+ (8.8E+02)	109.05+ (6.6E+01)
	20	1.0757 (1.6E-02)	160.00+ (4.8E+02)	156.27+ (1.1E+03)	305.74+ (7.0E+02)	194.22+ (1.5E+03)	298.18+ (3.7E+03)	300.28+ (2.1E+03)	300.13+ (2.2E+03)

注: 加粗字体表示各组的最优结果值。

算法处理问题 F1 ~ F5 时得到的 GS 值, 表 12 列出了各算法求解问题 F6 ~ F20 时所得的 GS 和 VR 值; 进一步, 表 13 对表 11 和表 12 的实验结果进行了统计分析. 对于测试函数 F1 ~ F5, 从表 13 中的“好/平/差”可以看出, IMPSO-HES 占优对比算法 LIPS、EMO-MMO、R3PSO、FERPSO、NCDE、NSDE

和 ANDE 的测试数量分别为 6、7、10、8、10、10 和 10, 而其劣于对比算法的测试函数数量分别为 1、0、0、0、1、0、0、0. 从“Rank”值可以看出, IMPSO-HES 的排序为 1.3000, 均小于对比算法的序值. 从“Adjusted p-value”可以看出, IMPSO-HES 显著优于 R3PSO、NCDE、NSDE 和 ANDE, 但与其余 3 种

表 12 处理 F6 ~ F20 时 IMPSO-HES 与 7 种多模态进化算法所得结果
Table 12 Results of IMPSO-HES and the 7 multimodal EAs on F6 ~ F20

问题	<i>D</i>	IMPSO-HES	LIPS	EMO-MMO	R3PSO	FERPSO	NCDE	NSDE	ANDE
F6	<i>GS</i> 均值	-199.15	-185.64+	-196.52+	-190.93+	-186.31+	-191.25+	-197.86+	-195.52+
	(标准差)	(4.6E+00)	(8.8E+01)	(1.0E+02)	(6.1E+01)	(1.0E+02)	(3.4E+02)	(4.5E+01)	(5.0E+02)
	<i>VR</i> 均值	0.80	0.20+	0.40+	0.10+	0.00+	0.65+	0.75=	0.40+
F7	<i>GS</i> 均值	-0.999 9	-0.999 4+	-0.999 5+	-0.999 1+	-0.998 6+	-0.998 7+	-0.998 4+	-0.998 0+
	(标准差)	(3.1E-06)	(7.3E-07)	(2.5E-07)	(7.2E-07)	(1.0E-06)	(8.5E-07)	(5.6E-06)	(4.6E-06)
	<i>VR</i> 均值	0.78	0.78=	0.76=	0.70=	0.66+	0.74=	0.78=	0.67+
F8	<i>GS</i> 均值	-0.985 4	-0.969 3+	-0.993 7-	-0.993 1-	-0.975 8+	-0.966 0+	-0.948 3+	-0.968 3+
	(标准差)	(1.3E-04)	(6.8E-04)	(2.5E-04)	(6.7E-05)	(4.1E-04)	(8.9E-04)	(5.1E-03)	(3.1E-03)
	<i>VR</i> 均值	1.00	0.80+	0.90+	1.00=	1.00=	0.90+	0.60+	0.80+
F9	<i>GS</i> 均值	-199.99	-197.58+	-197.79+	-196.99+	-196.92+	-197.04+	-196.10+	-197.22+
	(标准差)	(1.0E-04)	(1.7E+00)	(9.9E+00)	(1.3E+01)	(8.6E+00)	(5.2E+00)	(1.6E+01)	(1.3E+01)
	<i>VR</i> 均值	0.70	0.02+	0.05+	0.07+	0.07+	0.10+	0.05+	0.05+
F10	<i>GS</i> 均值	-1.031 6	-1.004 7+	-1.001 6+	-1.003 2+	-0.994 9+	-0.987 8+	-0.973 0+	-1.002 0+
	(标准差)	(9.8E-07)	(3.6E-04)	(2.8E-03)	(2.8E-03)	(8.8E-04)	(8.7E-03)	(5.0E-03)	(3.4E-02)
	<i>VR</i> 均值	1.00	0.55+	0.10+	0.45+	0.30+	0.40+	0.35+	0.5+
F11	<i>GS</i> 均值	-158.32	-105.20+	-134.50=	-90.154+	-114.099+	-123.777+	-111.92+	-132.37=
	(标准差)	(1.9E+03)	(1.3E+03)	(1.7E+03)	(5.4E+02)	(1.3E+03)	(1.0E+03)	(2.3E+03)	(1.6E+03)
	<i>VR</i> 均值	0.02	0.01=	0.01=	0.00+	0.00+	0.00+	0.00+	0.01=
F12	<i>GS</i> 均值	-0.999 9	-0.973 3+	-0.975 3+	-0.972 7+	-0.976 4+	-0.976 4+	-0.989 0+	-0.988 7+
	(标准差)	(1.0E-06)	(3.2E-04)	(4.9E-04)	(4.6E-04)	(5.8E-04)	(5.2E-04)	(3.0E-04)	(4.6E-03)
	<i>VR</i> 均值	0.13	0.08+	0.05+	0.07+	0.07+	0.08+	0.10+	0.09+
F13	<i>GS</i> 均值	2.232 9	2.714 6+	2.560 4+	2.438 4+	2.590 3+	2.481 7+	2.344 6=	2.579 2+
	(标准差)	(2.3E-01)	(3.2E-01)	(2.3E+00)	(2.1E-01)	(2.4E-01)	(7.3E-01)	(8.7E-01)	(2.2E+00)
	<i>VR</i> 均值	0.09	0.08=	0.08=	0.07=	0.08=	0.13+	0.09=	0.08=
F14	<i>GS</i> 均值	0.087 9	44.360+	45.829+	43.836+	38.669+	40.250+	38.149+	41.010+
	(标准差)	(5.0E-01)	(4.0E+03)	(4.8E+03)	(4.5E+03)	(4.5E+03)	(4.3E+03)	(1.6E+03)	(1.2E+02)
	<i>VR</i> 均值	0.24	0.01+	0.01+	0.00+	0.01+	0.01+	0.00+	0.00+
F15	<i>GS</i> 均值	36.423	103.12+	85.620+	108.46+	82.451+	67.647+	75.308+	89.100+
	(标准差)	(3.7E+03)	(1.4E+03)	(6.8E+03)	(3.2E+03)	(2.7E+03)	(1.7E+03)	(6.6E+03)	(1.8E+03)
	<i>VR</i> 均值	0.03	0.00+	0.01=	0.00+	0.01=	0.00+	0.00+	0.00+
F16	<i>GS</i> 均值	0.242 3	74.272+	52.296+	132.800+	52.555+	81.104+	114.04+	67.231+
	(标准差)	(1.3E-01)	(8.2E+03)	(8.1E+03)	(6.6E+03)	(3.2E+03)	(9.0E+03)	(1.6E+03)	(1.6E+03)
	<i>VR</i> 均值	0.15	0.00+	0.02+	0.00+	0.00+	0.00+	0.00+	0.00+
F17	<i>GS</i> 均值	32.566	127.50+	141.05+	165.93+	148.05+	192.72+	162.20+	100.12+
	(标准差)	(2.0E+04)	(2.3E+03)	(2.5E+04)	(5.7E+03)	(2.0E+03)	(8.5E+03)	(5.2E+03)	(3.2E+03)
	<i>VR</i> 均值	0.13	0.00+	0.00+	0.00+	0.00+	0.00+	0.00+	0.00+
F18	<i>GS</i> 均值	-0.267 9	-0.264 2+	-0.260 3+	-0.257 9+	-0.262 5+	-0.254 8+	-0.260 4+	-0.263 0+
	(标准差)	(1.6E-06)	(6.0E-05)	(6.8E-05)	(9.1E-05)	(4.9E-05)	(1.6E-04)	(1.8E-05)	(1.5E-04)
	<i>VR</i> 均值	1.00	1.00=	0.95=	0.95=	1.00=	0.80+	0.80+	0.85+
F19	<i>GS</i> 均值	0.399 9	0.529 2+	0.882 5+	0.797 9+	0.776 3+	0.789 5+	1.337 5+	0.885 8+
	(标准差)	(2.4E-05)	(1.9E-01)	(1.8E-01)	(1.8E-01)	(1.8E-01)	(1.7E+00)	(8.6E-02)	(2.0E-01)
	<i>VR</i> 均值	0.60	0.03+	0.03+	0.10+	0.06+	0.16+	0.06+	0.2+
F20	<i>GS</i> 均值	-7.429 9	-6.619 2+	-6.649 6+	-6.664 4+	-6.728 0+	-6.679 1+	-6.420 4+	-6.981 8+
	(标准差)	(1.7E-02)	(2.9E-01)	(8.6E-01)	(4.0E-01)	(3.3E-01)	(4.5E-01)	(3.8E-01)	(4.6E-01)
	<i>VR</i> 均值	0.40	0.26+	0.26+	0.26+	0.20+	0.29+	0.27+	0.28+

注: 加粗字体表示各组的最佳结果值。

表 13 IMPSO-HES 与 7 种多模态进化算法的统计对比结果
Table 13 Statistical comparison results of IMPSO-HES and the 7 multimodal EAs

问题		IMPSO-HES	LIPS	EMO-MMO	R3PSO	FERPSO	NCDE	NSDE	ANDE	
F1 ~ F5	好/平/差	<i>GS</i>	—	6/3/1	7/3/0	10/0/0	8/1/1	10/0/0	10/0/0	10/0/0
	Rank		1.300 0	2.800 0	3.000 0	6.100 0	3.300 0	6.300 0	6.800 0	6.700 0
	Adjusted <i>p</i> -value		—	0.315 3	0.116 0	0.000 2	0.0937	0.000 1	0.000 0	0.000 0
F6 ~ F20	好/平/差	<i>GS</i>	—	15/0/0	13/1/1	14/0/1	15/0/0	15/0/0	14/1/0	14/1/0
	Rank		1.258 6	4.827 5	4.268 9	5.551 7	4.603 4	6.103 4	4.862 0	4.224 1
	Adjusted <i>p</i> -value		—	0.000 0	0.000 0	0.000 0	0.000 0	0.000 0	0.000 0	0.000 0

对比算法无显著差异。

对于 F6 ~ F20, 由表 13 可以看出: 1) IMPSO-HES 得到的 *GS* 值均优于 LIPS、FERPSO 和 NCDE. 2) 对于指标 *VR* (有效解比例), IMPSO-HES 占优对比算法 LIPS、EMO-MMO、R3PSO、FERPSO、NCDE、NSDE 和 ANDE 的测试函数数量分别为 11、10、11、11、14、12 和 13; 而其劣于对比算法的测试函数数量均为 0. 3) 由“Rank”值可知, IMPSO-HES 在所有算法中排序第 1, 且排序值远小于其他对比算法. 4) 从 *p* 值可知, IMPSO-HES 均显著优于 7 种对比算法. 可见, 针对包含多个最优解的昂贵多模态优化问题, 相对 7 种对比算法, IMPSO-HES 能够在保证解质量同时找到问题更多的最优解。

4 建筑节能设计实例

为了检验 IMPSO-HES 处理实际问题的性能, 将其应用于建筑节能设计问题^[6]. 该问题以建筑能耗作为优化目标. 一方面, 对于一组设计参数, 计算其对应的建筑能耗需要调用高耗时的 EnergyPlus 仿真软件; 另一方面, 不同的参数组合可能会产生相近的建筑能耗, 因此, 该问题是一类典型的高昂多模态优化问题。

4.1 问题描述

实验以北京一类常见的办公建筑为例. 图 7 给出了这类建筑的基础外形. 该模型的长宽高分别为 8.8 m、3.6 m 和 3.9 m; 窗的初始长度为 1.7 m, 初始高度为 1.6 m. 参照 EnergyPlus 软件手册中的“输入输出参考”. 表 14 给出了优化模型的 12 个决策变量。

4.2 实验结果

在第 3.6 节中 EMO-MMO 的综合排序仅次于 IMPSO-HES, 因此, 实验选择 EMO-MMO 作为对比算法. 为了对比的公平性, 两种算法设置相同的

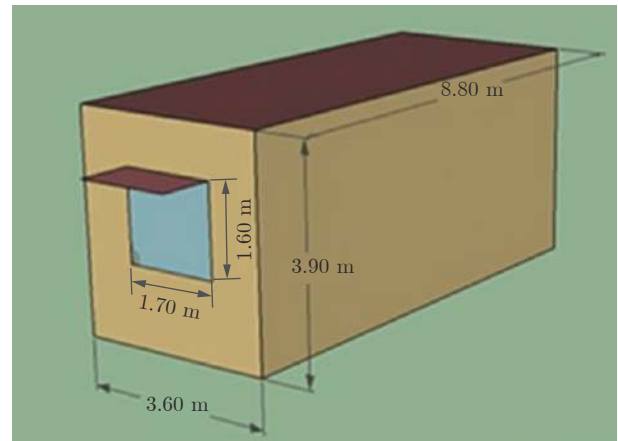


图 7 单居室居住建筑的外形图

Fig. 7 Outline of the single-room building

表 14 问题的决策变量信息

Table 14 Decision variable information of the problem

决策变量	单位	范围
房屋方向	(°)	[0, 360]
窗户的长	m	(0, 3.6)
窗户的高	m	(0, 3.9)
窗户的传热系数	W/(m ² · K)	[2, 6]
窗户的日射热取得率	—	(0, 0.7)
墙体外保温层厚度	m	(0, 0.1]
墙体日射吸收率	—	[0.1, 1]
人员密度	人/m ²	[0.1, 1)
照明功率密度	W/m ²	[6, 12]
设备功率密度	W/m ²	[10, 18]
空调供热设置温度	°C	[18, 23]
空调制冷设置问题	°C	[24, 28]

种群规模和最大进化代数, 其中, 种群规模均设为 100, 最大进化代数设为 50, 其余参数均保持原文设置. 两种算法的运行环境为 Intel Core i5, CPU 1.6 GHz, MATLAB R2014b 编程, 并调用 EnergyPlus 软件计算建筑能耗. 表 15 给出了实验结果, 其中, “Op-

表 15 处理建筑节能设计问题时两种算法所得的实验结果
Table 15 Results of the two algorithms on building energy conservation

	<i>GS</i>	Optimal solutions	时间 (s)
IMPSO-HES	5.02	$X = 71.8, 1.06, 1.85, 3.64, 0.0382, 0.0905, 0.2212, 0.1033, 6.5, 14.0, 22.3, 26.4, f = 5.1$	450
		$X = 297.3, 2.53, 1.63, 4.0065, 0.0556, 0.0402, 0.5983, 0.1027, 6.0, 17.2, 19.6, 24.0, f = 5.1$	
		$X = 351.7, 3.50, 0.38, 2.266, 0.1604, 0.0567, 0.8882, 0.1062, 6.1, 17.3, 22.6, 24.6, f = 5.11$	
EMO-MMO	4.96	$X = 183.2, 1.19, 2.36, 2.32, 0.3439, 0.0489, 0.9743, 0.1085, 6.18, 12.3, 21.1, 26.3, f = 5.01$	42 357
		$X = 215.1, 2.41, 2.09, 5.38, 0.2847, 0.0532, 0.4720, 0.1015, 6.44, 11.8, 19.3, 27.1, f = 5.02$	
		$X = 134.7, 1.07, 2.87, 3.73, 0.3129, 0.0418, 0.9553, 0.1015, 6.02, 12.8, 20.4, 25.3, f = 5.02$	

timal solutions”为某次运行中算法得到的最优解集. 需要说明的是, IMPSO-HES 的运行代价中包括了建立数据库的时间.

从表 15 中可以看出: 1) IMPSO-HES 得到的平均 *GS* 值略大 EMO-MMO; 2) EMO-MMO 的运行耗时远远大于 IMPSO-HES. 造成上述两种现象的原因在于, 在相同的种群规模和进化代数下, EMO-MMO 需要采用真实函数评价所有个体, 而 IMPSO-HES 只需要采用真实函数评价少部分代表个体; 3) 两种算法都得到了一个以上的最优解或次优解, 决策者可以根据自己的需要选择不同的解决方案. 综上所述, IMPSO-HES 能够以较小的计算代价找到问题的多个最优或次优解.

5 结束语

针对昂贵多模态优化问题, 本文提出一种异构集成代理辅助的多模态粒子群优化算法, 即 IMPSO-HES. 在更新模型池时, 通过种群反馈信息自适应调整 RBFN 和 PR 被选择的概率, 可以使构建的模型池更适合处理当前优化问题. 设计的基于模态的基础代理模型选择策略, 可以让不同粒子选择更适合自己的基础代理模型进行集成, 提高集成代理模型的预测精度. 提出的两阶段增量式模型更新策略, 在提高集成代理模型的质量同时减少了模型的训练代价. 此外, 引入的区间适应值评价策略和 PSO 自适应更新策略, 在提高算法探索能力的同时提升了最优解的精度. 在 20 个基准测试函数和 1 个建筑能源优化问题上的实验结果表明, 相比 12 种典型对比算法, 在有限计算资源内, 所提算法在获得较好全局最优解的同时, 也能找到较多的全局最优解或局部最优解.

尽管在处理昂贵多模态优化问题时 IMPSO-HES 表现出较好的性能, 但是目前这方面的研究工作相对较少, 仍有很多问题需要进一步研究. 首先, 本文仅采用常用的 2 类代理模型. 代理模型的种类多样, 集成其他代理模型是否会获得更好的结果需

要进一步研究; 另外, 含约束昂贵多模态优化问题和昂贵多目标多模态优化问题也是未来需要研究的课题.

References

- Kruisselbrink J W, Aleman A, Emmerich M T M, Ijzerman A P, Bender A, Baeck T, et al. Enhancing search space diversity in multi-objective evolutionary drug molecule design using niching. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation. Montreal, Canada: ACM, 2009. 217-224
- Pérez E, Herrera F, Hernández C. Finding multiple solutions in job shop scheduling by niching genetic algorithms. *Journal of Intelligent Manufacturing*, 2003, **14**(3-4): 323-339
- Li Zhang-Wei, Xiao Lu-Qian, Hao Xiao-Hu, Zhou Xiao-Gen, Zhang Gui-Jun. Multimodal optimization algorithm for protein conformation space. *Computer Science*, 2020, **47**(7): 161-165 (李章维, 肖璐倩, 郝小虎, 周晓根, 张贵军. 蛋白质构象空间的多模态优化算法. *计算机科学*, 2020, **47**(7): 161-165)
- Wang Z J, Zhan Z H, Lin Y, Yu W J, Wang H, Kwong S, et al. Automatic niching differential evolution with contour prediction approach for multimodal optimization problems. *IEEE Transactions on Evolutionary Computation*, 2020, **24**(1): 114-128
- Li Hang, Li Min-Qiang, Kou Ji-Song. Dynamical behavior of genetic algorithms on multi-modal optimization. *Acta Automatica Sinica*, 2008, **34**(2): 180-187 (李航, 李敏强, 寇纪淞. 遗传算法求解多模态优化问题的动力性. *自动化学报*, 2008, **34**(2): 180-187)
- Zhang Y, Yuan L J, Zhang Q, Sun X Y. Multi-objective optimization of building energy performance using a particle swarm optimizer with less control parameters. *Journal of Building Engineering*, 2020, **32**: Article No. 101505
- Ji X F, Zhang Y, Gong D W, Sun X Y. Dual-surrogate-assisted cooperative particle swarm optimization for expensive multimodal problems. *IEEE Transactions on Evolutionary Computation*, 2021, **25**(4): 794-808
- Li X D. Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference. Seattle, USA: Springer, 2004. 105-116
- Cheng R, Li M Q, Li K, Yao X. Evolutionary multiobjective optimization-based multimodal optimization: Fitness landscape approximation and peak detection. *IEEE Transactions on Evolutionary Computation*, 2018, **22**(5): 692-706
- Li X D, Epitropakis M G, Deb K, Engelbrecht A. Seeking multiple solutions: An updated survey on niching methods and their applications. *IEEE Transactions on Evolutionary Computation*,

- 2017, **21**(4): 518–538
- 11 Zhang Shu-Mei, Wang Fu-Li, Tan Shuai, Wang Shu. A fully automatic offline mode identification method for multi-mode processes. *Acta Automatica Sinica*, 2016, **42**(1): 60–80
(张淑美, 王福利, 谭帅, 王姝. 多模态过程的全自动离线模式识别方法. *自动化学报*, 2016, **42**(1): 60–80)
- 12 Mahfoud S W. Crowding and preselection revisited. In: Proceedings of the Parallel Problem Solving from Nature 2, PPSN-II. Brussels, Belgium: Elsevier, 1992. 27–36
- 13 Zhang Gui-Jun, He Yang-Jun, Guo Hai-Feng, Feng Yuan-Jing, Xu Jian-Ming. Differential evolution algorithm for multimodal optimization based on abstract convex underestimation. *Journal of Software*, 2013, **24**(6): 1177–1195
(张贵军, 何洋军, 郭海峰, 冯远静, 徐建明. 基于广义凸下界估计的多模态差分进化算法. *软件学报*, 2013, **24**(6): 1177–1195)
- 14 Petrowski A. A clearing procedure as a niching method for genetic algorithms. In: Proceedings of the IEEE International Conference on Evolutionary Computation. Nagoya, Japan: IEEE, 1996. 798–803
- 15 Li X D. Niching without niching parameters: Particle swarm optimization using a ring topology. *IEEE Transactions on Evolutionary Computation*, 2010, **14**(1): 150–169
- 16 Wang Xiang-Zhong, Yu Shou-Yi. Multi-population evolution strategies for multi-modal function optimization. *Control and Decision*, 2006, **21**(3): 285–288
(王湘中, 喻寿益. 多模态函数优化的多种群进化策略. *控制与决策*, 2006, **21**(3): 285–288)
- 17 Zhang Gui-Jun, Chen Ming, Zhou Xiao-Gen. Two-stage differential evolution algorithm using dynamic niche radius for multimodal optimization. *Control and Decision*, 2016, **31**(7): 1185–1191
(张贵军, 陈铭, 周晓根. 动态小生境半径两阶段多模态差分进化算法. *控制与决策*, 2016, **31**(7): 1185–1191)
- 18 Holland J H. *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan, 1975.
- 19 Li Min-Qiang, Kou Ji-Song. Coordinate multi-population genetic algorithms for multi-modal function optimization. *Acta Automatica Sinica*, 2002, **28**(4): 497–504
(李敏强, 寇纪淞. 多模态函数优化的协同多群体遗传算法. *自动化学报*, 2002, **28**(4): 497–504)
- 20 Qu B Y, Suganthan P N, Das S. A distance-based locally informed particle swarm model for multimodal optimization. *IEEE Transactions on Evolutionary Computation*, 2013, **17**(3): 387–402
- 21 Biswas S, Kundu S, Das S. Inducing niching behavior in differential evolution through local information sharing. *IEEE Transactions on Evolutionary Computation*, 2015, **19**(2): 246–263
- 22 Deb K, Saha A. Multimodal optimization using a Bi-objective evolutionary algorithm. *Evolutionary Computation*, 2012, **20**(1): 27–62
- 23 Bandaru S, Deb K. A parameterless-niching-assisted bi-objective approach to multimodal optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation. Cancún, Mexico: IEEE, 2013. 95–102
- 24 Yue C T, Liang J J, Qu B Y, Yu K J, Song H. Multimodal multiobjective optimization in feature selection. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC). Wellington, New Zealand: IEEE, 2019. 302–309
- 25 Lim D, Jin Y C, Ong Y S, Sendhoff B. Generalizing surrogate-assisted evolutionary computation. *IEEE Transactions on Evolutionary Computation*, 2010, **14**(3): 329–355
- 26 Liu B, Zhang Q F, Gielen G G E. A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Transactions on Evolutionary Computation*, 2014, **18**(2): 180–192
- 27 Clarke S M, Griebisch J H, Simpson T W. Analysis of support vector regression for approximation of complex engineering analyses. *Journal of Mechanical Design*, 2005, **127**(6): 1077–1087
- 28 Regis R G. Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions. *IEEE Transactions on Evolutionary Computation*, 2014, **18**(3): 326–347
- 29 Roux W J, Stander N, Haftka R T. Response surface approximations for structural optimization. *Numerical Methods in Engineering*, 1998, **42**(3): 517–534
- 30 Wang H D, Jin Y C, Doherty J. Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Transactions on Cybernetics*, 2017, **47**(9): 2664–2677
- 31 Min A T W, Ong Y S, Gupta A, Goh C K. Multiproblem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems. *IEEE Transactions on Evolutionary Computation*, 2019, **23**(1): 15–28
- 32 Wang H D, Jin Y C, Sun C L, Doherty J. Offline data-driven evolutionary optimization using selective surrogate ensembles. *IEEE Transactions on Evolutionary Computation*, 2019, **23**(2): 203–216
- 33 Loshchilov I, Schoenauer M, Sebag M. Comparison-based optimizers need comparison-based surrogates. In: Proceedings of the 11th International Conference on Parallel Problem Solving From Nature. Krakow, Poland: Springer, 2010. 364–373
- 34 Long Teng, Guo Xiao-Song, Peng Lei, Liu Li. Optimization strategy using dynamic radial basis function metamodel based on trust region. *Journal of Mechanical Engineering*, 2014, **50**(7): 184–190
(龙腾, 郭晓松, 彭磊, 刘莉. 基于信赖域的动态径向基函数代理模型优化策略. *机械工程学报*, 2014, **50**(7): 184–190)
- 35 Chugh T, Jin Y C, Miettinen K, Hakanen J, Sindhya K. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2018, **22**(1): 129–142
- 36 Sun Chao-Li, Li Zhen, Jin Yao-Chu. Surrogate-assisted expensive evolutionary many-objective optimization. *Acta Automatica Sinica*, 2022, **48**(4): 1119–1128
(孙超利, 李贞, 金耀初. 模型辅助的计算费时进化高维多目标优化. *自动化学报*, 2022, **48**(4): 1119–1128)
- 37 Tian Jie, Sun Chao-Li, Tan Ying, Zeng Jian-Chao. Similarity-based multipoint infill criterion for surrogate-assisted social learning particle swarm optimization. *Control and Decision*, 2020, **35**(1): 131–138
(田杰, 孙超利, 谭瑛, 曾建潮. 基于多点加准则的代理模型辅助社会学习微粒群算法. *控制与决策*, 2020, **35**(1): 131–138)
- 38 Tian Jie, Tan Ying, Sun Chao-Li, Zeng Jian-Chao. Surrogate-assisted evolutionary optimization for high-dimensional expensive optimization. *Machinery Design and Manufacture*, 2018, **12**: 269–272
(田杰, 谭瑛, 孙超利, 曾建潮. 代理模型辅助进化算法在高维优化问题中的应用. *机械设计与制造*, 2018, **12**: 269–272)
- 39 Sun C L, Jin Y C, Zeng J C, Yu Y. A two-layer surrogate-assisted particle swarm optimization algorithm. *Soft Computing*, 2015, **19**(6): 1461–1475
- 40 Cai X W, Qiu H B, Gao L, Jiang C, Shao X Y. An efficient sur-

- rogate-assisted particle swarm optimization algorithm for high-dimensional expensive problems. *Knowledge-Based Systems*, 2019, **184**: Article No. 104901
- 41 Liao P, Sun C L, Zhang G C, Jin Y C. Multi-surrogate multi-tasking optimization of expensive problems. *Knowledge-Based Systems*, 2020, **205**: Article No. 106262
- 42 Tang Y F, Chen J Q, Wei J H. A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions. *Engineering Optimization*, 2013, **45**(5): 557–576
- 43 Li F, Cai X W, Gao L. Ensemble of surrogates assisted particle swarm optimization of medium scale expensive problems. *Applied Soft Computing*, 2019, **74**: 291–305
- 44 Liu Q F, Wu X F, Lin Q Z, Ji J K, Wong K C. A novel surrogate-assisted evolutionary algorithm with an uncertainty grouping based infill criterion. *Swarm and Evolutionary Computation*, 2021, **60**: Article No. 100787
- 45 Dong H C, Li C S, Song B W, Wang P. Multi-surrogate-based differential evolution with multi-start exploration (MDEME) for computationally expensive optimization. *Advances in Engineering Software*, 2018, **123**: 62–76
- 46 Gu J C, Li W Q, Cai Y Z. A hybrid meta-model based global optimization method for expensive problems. *Computers and Industrial Engineering*, 2019, **136**: 421–428
- 47 Guo D, Jin Y C, Ding J L, Chai T Y. Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems. *IEEE Transactions on Cybernetics*, 2019, **49**(3): 1012–1025
- 48 Tenne Y. Online ensemble topology selection in expensive optimization problems. *International Journal of Control, Automation and Systems*, 2020, **18**(4): 955–965
- 49 Chen Wan-Fen, Wang Yu-Jia, Lin Wei-Xing. Heterogeneous ensemble surrogate assisted multi-objective particle swarm optimization algorithm. *Computer Engineering and Applications*, 2021, **57**(23): 71–80
(陈万芬, 王宇嘉, 林炜星. 异构集成代理辅助多目标粒子群优化算法. *计算机工程与应用*, 2021, **57**(23): 71–80)
- 50 Branke J, Schmidt C. Faster convergence by means of fitness estimation. *Soft Computing*, 2005, **9**(1): 13–20
- 51 Wang H D, Jin Y C, Yang C E, Jiao L C. Transfer stacking from low-to high-fidelity: A surrogate-assisted bi-fidelity evolutionary algorithm. *Applied Soft Computing*, 2020, **92**: Article No. 106276
- 52 Emmerich M, Giotis A, Özdemir M, Bäeck T, Giannakoglou K. Metamodel-assisted evolution strategies. In: Proceedings of the 7th International Conference on Parallel Problem Solving From Nature. Granada, Spain: Springer, 2002. 361–370
- 53 Yahyaie F, Filizadeh S. A surrogate-model based multi-modal optimization algorithm. *Engineering Optimization*, 2011, **43**(7): 779–799
- 54 Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In: Proceedings of the 6th International Symposium on Micro Machine and Human Science. Nagoya, Japan: IEEE, 1995. 39–43
- 55 Shi Y, Eberhart R. A modified particle swarm optimizer. In: Proceedings of the IEEE International Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence (Cat. No.98TH8360). Anchorage, USA: IEEE, 1998. 69–73
- 56 Sun C L, Jin Y C, Cheng R, Ding J L, Zeng J C. Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Transactions on Evolutionary Computation*, 2017, **21**(4): 644–660
- 57 Tian J, Sun C L, Tan Y, Zeng J C. Granularity-based surrogate-assisted particle swarm optimization for high-dimensional expensive optimization. *Knowledge-Based Systems*, 2020, **187**: Article No. 104815
- 58 Goel T, Haftka R T, Shyy W, Queipo N V. Ensemble of surrogates. *Structural and Multidisciplinary Optimization*, 2007, **33**(3): 199–216
- 59 Gutmann H M. A radial basis function method for global optimization. *Journal of Global Optimization*, 2001, **19**(3): 201–227
- 60 Zhou L G, Chen H Y, Merigó J M, Gil-Lafuente A M. Uncertain generalized aggregation operators. *Expert Systems With Applications*, 2012, **39**(1): 1105–1117
- 61 Li De-Qing, Han Guo-Zhu, Zeng Wen-Yi, Yu Xian-Chuan. Ranking method of interval numbers based on Boolean matrix. *Control and Decision*, 2016, **31**(4): 629–634
(李德清, 韩国柱, 曾文艺, 余先川. 基于布尔矩阵的区间数排序方法. *控制与决策*, 2016, **31**(4): 629–634)
- 62 Fan C D, Hou B, Zheng J H, Xiao L Y, Yi L Z. A surrogate-assisted particle swarm optimization using ensemble learning for expensive problems with small sample datasets. *Applied Soft Computing*, 2020, **91**: Article No. 106242
- 63 Efron B, Tibshirani R J. *An Introduction to the Bootstrap*. New York: Chapman & Hall, 1993.
- 64 Li X D, Engelbrecht A, Epitropakis M G. Benchmark Functions for CEC' 2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization, Technical Report, RMIT University, Melbourne, Australia, 2013.
- 65 Stoean C, Preuss M, Stoean R, Dumitrescu D. Multimodal optimization by means of a topological species conservation algorithm. *IEEE Transactions on Evolutionary Computation*, 2010, **14**(6): 842–864
- 66 Qu B Y, Suganthan P N, Liang J J. Differential evolution with neighborhood mutation for multimodal optimization. *IEEE Transactions on Evolutionary Computation*, 2012, **16**(5): 601–614
- 67 Friedman M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 1937, **32**(200): 675–701
- 68 Finner H. On a monotonicity problem in step-down multiple test procedures. *Journal of the American Statistical Association*, 1993, **88**(423): 920–923



季新芳 中国矿业大学信息与控制工程学院博士研究生. 2013 年获得中国矿业大学硕士学位. 主要研究方向为代理辅助进化优化, 多模态优化.

E-mail: mimosa_615615@126.com

(JI Xin-Fang Ph.D. candidate at the School of Information and Control Engineering, China University of Mining and Technology. She received her master degree from China University of Mining and Technology in 2013. Her research interest covers surrogate-assisted evolutionary optimization and multimodal optimization.)



张 勇 中国矿业大学信息与控制工程学院教授. 2009 年获中国矿业大学控制理论与控制工程专业博士学位. 主要研究方向为智能优化, 数据挖掘. 本文通信作者.

E-mail: yongzh401@126.com

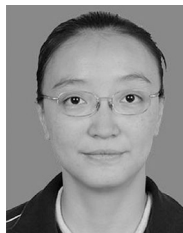
(ZHANG Yong Professor at the School of Information and Control Engineering, China University of Mining and Technology. He received his Ph.D. degree in control theory and control engineering from China University of Mining and Technology in 2009. His research interest covers intelligence optimization, data mining. Corresponding author of this paper.)



巩敦卫 中国矿业大学信息与控制工程学院教授. 1999 年获得中国矿业大学博士学位. 主要研究方向为进化计算与应用.

E-mail: dwgong@vip.163.com

(GONG Dun-Wei Professor at the School of Information and Control Engineering, China University of Mining and Technology. He received his Ph.D. degree from China University of Mining and Technology in 1999. His research interest covers evolutionary computation and its applications.)



郭一楠 中国矿业大学信息与控制工程学院教授. 主要研究方向为智能优化算法与控制, 数据挖掘.

E-mail: nanly@126.com

(GUO Yi-Nan Professor at the School of Information and Control Engineering, China University of Mining and Technology. Her research interest covers intelligence optimization, control, and data mining.)



孙晓燕 中国矿业大学信息与控制工程学院教授. 2009 年获中国矿业大学控制理论与控制工程专业博士学位. 主要研究方向为进化计算, 机器学习.

E-mail: xysun78@126.com

(SUN Xiao-Yan Professor at the School of Information and Control Engineering, China University of Mining and Technology. She received her Ph.D. degree in control theory and control engineering from China University of Mining and Technology in 2009. Her research interest covers evolutionary computation and machine learning.)