

模型辅助的计算费时进化高维多目标优化

孙超利¹ 李贞¹ 金耀初²

摘要 代理模型能够辅助进化算法在计算资源有限的情况下加快找到问题的最优解集, 因此建立高效的代理模型辅助多目标进化搜索逐渐受到了重视. 然而随着目标数量的增加, 对每个目标分别建立高斯过程模型时个体整体估值的不确定度会随之增加. 因此通过对模型最优解集的搜索探索原问题潜在的非支配解集, 并基于个体的收敛性, 种群的多样性和估值的不确定度, 提出了一种新的期望提高计算方法, 用于辅助从潜在的非支配解集中选择使用真实目标函数计算的个体, 从而更新代理模型, 能够在有限的计算资源下更有效地辅助优化算法找到好的非支配解集. 在 7 个 DTLZ 基准测试问题上的实验对比结果表明, 该算法在求解计算费时高维多目标优化问题上是有效的, 且具有较强的竞争力.

关键词 高维多目标优化, 代理模型, 计算费时问题, 填充准则

引用格式 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

DOI 10.16383/j.aas.c200969

Surrogate-assisted Expensive Evolutionary Many-objective Optimization

SUN Chao-Li¹ LI Zhen¹ JIN Yao-Chu²

Abstract Surrogate models have attracted increasing attention in assisting evolutionary many-objective optimization when the computational budget is limited since a surrogate model can assist to accelerate the search for a set of Pareto solutions. However, the approximation uncertainty of an individual on the objective approximated values will be increased when the number of objectives increases in the case that a surrogate model is trained for each objective. Therefore, we propose to explore the potential non-dominated solutions of the original optimization problem by searching for the optimal solutions of the surrogate models, and a new expected improvement in this paper, which takes into account on the convergence of the solution, the diversity of the population, and the uncertainty of the approximation, for assisting selecting solutions from the potential non-dominated solutions to be evaluated using the exact expensive objective function. The surrogate model will be updated using the new evaluated solutions and is expected to assist the optimization algorithms to efficiently find a good set of non-dominated solutions within a limited computational budget. The experimental results on seven DTLZ test problems show that our proposed method is efficient and competitive to solve expensive many-objective problems.

Key words Many-objective optimization, surrogate models, computationally expensive problems, infill criterion

Citation Sun Chao-Li, Li Zhen, Jin Yao-Chu. Surrogate-assisted expensive evolutionary many-objective optimization. *Acta Automatica Sinica*, 2022, 48(4): 1119–1128

在复杂的工程优化问题中, 通常有多个目标需要同时优化, 而这些目标之间往往相互冲突和影响, 即一个目标的改善会导致至少一个其他目标的恶化, 这些问题被称为多目标优化问题^[1]. 一般多目标

优化问题^[2]的数学模型可表示为:

$$\begin{aligned} \min \mathbf{F}(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})) \\ \text{s.t. } \mathbf{x} &= (x_1, x_2, \dots, x_D)^T \in \mathbf{R}^D \end{aligned} \quad (1)$$

其中, M 是目标个数, \mathbf{x} 是 D 维决策空间 \mathbf{R}^D 中的一个决策向量. 在优化问题中, 进化算法 (Evolutionary algorithm, EA)^[3] 由于其不需要假设任何目标函数的凹凸性, 可微性或约束性, 且有更多的机会获得全局最优解, 因而获得了工业界和科学界的关注, 并且在实际工程中得到了很多应用. 求解多目标优化问题的进化算法 (Multi-objective evolutionary algorithm, MOEA)^[4] 通常分为 4 大类: 1) 基于支配关系的进化多目标算法: 如快速非支配排序的遗传算法 (Nondominated sorting genetic algorithm II, NSGA-II)^[5–6]; 提升强度的 Pareto 进化

收稿日期 2020-11-22 录用日期 2021-03-19
Manuscript received November 22, 2020; accepted March 19, 2021

国家自然科学基金 (61876123), 山西省自然科学基金 (201901D111262, 201901D111264) 资助

Supported by National Natural Science Foundation of China (61876123), Natural Science Foundation of Shanxi Province (201901D111262, 201901D111264)

本文责任编辑 杨涛

Recommended by Associate Editor YANG Tao

1. 太原科技大学计算机科学与技术学院 太原 030024 中国 2. 萨里大学计算机科学系 吉尔福德 GU2 7XH 英国

1. School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China. 2. Department of Computer Science, University of Surrey, Guildford GU2 7XH, UK

算法^[7]; 2) 基于分解的进化多目标算法:如基于分解的多目标进化算法 (Multiobjective evolutionary algorithm based on decomposition, MOEA/D)^[8]、参考向量引导进化算法 (Reference vector guided evolutionary algorithm, RVEA)^[9]; 3) 基于指标的进化多目标算法:基于指标的进化算法^[10]、基于超体积估计的算法^[11]; 4) 其他算法:如基于分解和支配的高维多目标进化算法 (Many-objective optimization algorithm based on dominance and decomposition, MOEA-DD)^[12]、基于双目标优化的进化算法 (Bi-goal evolution, BiGE)^[13]. 然而, 不管哪一类现有的多目标优化进化算法, 在搜寻最优解集的过程中都需要耗费大量的性能评估次数, 而在许多实际的多目标优化问题中其目标函数的评价非常昂贵, 如: 航空发动机管路卡箍布局优化^[14] 中, 一台典型的航空发动机通常包含上百根管路, 而涉及计算一根管路震动频率的模拟函数评估可能需要大量的时间, 因此很大程度地限制了多目标进化算法在这类问题中的应用. 目前求解昂贵的多目标优化问题常用的方法之一是引入代理模型, 使用模型代替昂贵多目标计算的进化算法通常称为代理模型辅助的进化多目标算法 (Surrogate-assisted evolutionary multi-objective optimization, SAEMO). 常见的求解多目标优化问题的 SAEMO 算法通常分为三类. 第 1 类是在多目标优化过程中直接用代理模型代替费时的目标函数计算来进行环境选择. 如 Akhtar 等^[15] 为每个目标建立一个径向基函数模型, 并提出用多个准则来选择具有代表性的点进行真实的目标函数评价. 如 Zhang 等^[16] 提出了高斯过程随机模型辅助的算法, 该算法对每个目标建立高斯过程模型, 基于分解策略将多目标问题转换成多个单目标优化问题, 根据个体每个目标的高斯过程模型估值计算切比雪夫函数值, 并利用获取函数进行环境选择. Chugh 等^[17] 提出对每个目标函数建立高斯过程模型, 并通过目标函数估值的角度惩罚距离指标值和估值的不确定度来选择真实计算的个体, 称为克里金模型辅助的 RVEA 算法 (Kriging-assisted RVEA, K-RVEA). 为了提高计算费时多目标问题的优化性能, Wang 等^[18] 在为每个目标函数建立代理模型的基础上引入一种自适应获取函数指标, 从而提出了一种新的采样选择标准. Yang 等^[19] 提出了离线数据驱动的多目标优化, 在进化算法中使用了粗代理模型和细代理模型两种模型, 粗代理模型用于引导算法快速地定位到较好的搜索空间, 同时细代理模型主要关注平衡粗代理模型知识迁移过来的好解. 文献 [20] 构建了一个正确模型和多个辅助

模型作为多个优化问题, 然后利用多任务优化方法来求解这些问题, 实现了从辅助模型到正确模型的迁移. Zhao 等^[21] 对多目标问题的每个目标建立了若干代理模型, 并基于目标空间和决策空间个体的距离提出了一种新的不确定度计算方法. 求解多目标优化问题的第 2 类 SAEMO 算法是对多目标问题的聚合函数建立代理模型, 即通过聚合函数将多目标转换为单目标, 对单目标建立代理模型, 从而辅助多目标优化. Knowles^[22] 基于求解单目标问题的有效全局优化算法 (Efficient global optimization, EGO), 提出使用切比雪夫函数将多目标优化问题转换成单目标优化问题, 并对单目标问题建立高斯过程模型, 利用获取函数选择个体进行真实计算, 从而实现了基于 EGO 的 Pareto 面寻优算法 (Pareto optimization with the efficient global optimization, ParEGO). 代理模型辅助的多目标优化算法中第 3 类是根据支配关系训练分类模型, 将代理模型作为分类器辅助多目标优化算法. 如 Pan 等^[23] 引入神经网络来预测参考点与候选解之间的优劣关系来选择好的候选解进行真实计算, 为一种基于分类的代理模型辅助进化算法 (A classification based surrogate-assisted evolutionary algorithm, CSEA). Zhang 等^[24] 提出利用个体间的支配关系训练支持向量机分类模型来预测后代个体的质量, 从而选择好的个体作为下一个父代.

虽然代理模型在单目标计算费时问题的优化中获得了较多关注, 但其在计算费时多目标优化问题中的应用还处于起步阶段, 目前还有很多亟待解决的问题.

1) 模型的选择. 目前常见的代理模型有多项式回归模型^[25], 径向基函数^[26-27], 高斯过程^[28], 神经网络^[29] 和支持向量机^[30] 等. 在进化过程中选择哪一种模型对目标函数进行估值会很大程度影响算法的寻优能力.

2) 模型的用途选择. 通常情况下, 全局代理模型用于辅助提高算法的探索能力, 局部代理模型用于辅助提高算法的开发能力. 而在多目标优化问题中, 由于有多个目标, 确定模型的用途更是进化多目标算法能否快速找到 Pareto 非支配解集的重要因素.

3) 填充标准. 如何选择个体进行真实目标函数计算并且更新模型在代理模型辅助的单目标和多目标进化优化中起着至关重要的作用, 其选择的好坏会直接影响模型更新后的准确度.

在 SAEMO 中, 模型估值的不确定度会影响算法的搜索方向, 从而影响算法的求解性能, 因此在

优化过程中, 估值的不确定度往往和估值同时考虑. 与多项式回归、径向基函数和人工神经网络等模型相比, 高斯过程代理模型不仅能够提供个体估值, 同时还能提供估值的不确定度, 因此本文选择高斯过程模型用来作为原目标函数的估值模型, 并通过对高斯过程模型最优解集的搜索, 探索最优解集可能存在的不同领域, 从而提高算法的开发能力. 另外, 模型搜索获得的最优解集是原优化问题的潜在非支配解, 因此从中选择真实计算的个体能够加快算法对原问题的求解效率. 然而, 由于高斯过程的获取函数是针对单目标优化问题的建模提出来的, 随着目标数的增加, 对每个目标分别建立高斯过程模型时个体估值的不确定度会随之增大. 因此, 针对多目标优化问题, 考虑到个体的收敛性、种群的多样性以及估值的不确定性, 本文对高斯过程模型的期望提高 (Expected improvement, EI) 获取函数进行了改进. 使用角度惩罚距离函数值作为个体的收敛性指标, 所有目标估值的不确定度均值作为个体的估值不确定度, 从而使算法在选择个体进行真实计算时在开发和开采能力上达到平衡.

本文主要贡献包含以下两个方面:

1) 通过对模型最优解集的搜索提高算法的开发能力, 使其能够引导种群向具有较好目标函数值的区域进化, 并从获得的最优解集中选择个体进行真实的目标函数评价, 从而加快收敛速度.

2) 考虑个体的收敛性、种群的多样性以及估值的不确定性, 针对计算费时多目标优化问题提出一种新的填充准则.

1 相关工作

1.1 高斯过程

高斯过程 (Gaussian process, GP) 是基于统计理论提出的机器学习方法^[28], 其性质由均值函数 $\mu(\mathbf{x})$ 和协方差函数 $k(\mathbf{x}_i, \mathbf{x}_j)$ 唯一确定,

$$\mu(\mathbf{x}) = E[f(\mathbf{x})] \quad (2)$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = E[(f(\mathbf{x}_i) - \mu(\mathbf{x}_i))(f(\mathbf{x}_j) - \mu(\mathbf{x}_j))] \quad (3)$$

其中, $\mathbf{x}_i, \mathbf{x}_j$ 代表决策空间 \mathbf{R} 中 2 个任意的 D 维向量, $\mu(\mathbf{x})$ 和 $k(\mathbf{x})$ 分别为均值函数和协方差函数. 因此, 给定数据集 $DS = \{(\mathbf{x}_i, f(\mathbf{x}_i)), i = 1, 2, \dots, n\}$, 假设训练集 $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_n]$, $\mathbf{Y} = f(\mathbf{x}_1); f(\mathbf{x}_2); \dots; f(\mathbf{x}_n)$, 则高斯过程模型可定义如下:

$$\hat{f}(\mathbf{x}) = f(\mathbf{x}) + \varepsilon \quad (4)$$

其中, $\hat{f}(\mathbf{x})$ 是高斯过程回归模型在 \mathbf{x} 上的预测值, ε 是一个随机变量, 服从均值为零, 方差为 σ^2 的高

斯分布. 由此可得

$$\hat{f}(\mathbf{x}) \sim N(0, \mathbf{K} + \sigma^2 \mathbf{I}) \quad (5)$$

其中, \mathbf{K} 为 $n \times n$ 阶对称的正定协方差矩阵, 每个元素 k_{ij} 表示 \mathbf{x}_i 和 \mathbf{x}_j 之间的相关性. 则

$$\begin{bmatrix} \mathbf{Y} \\ \mathbf{Y}^* \end{bmatrix} = N \left(0, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I} & \mathbf{K}(\mathbf{X}, \mathbf{X}^*) \\ \mathbf{K}(\mathbf{X}^*, \mathbf{X}) & \mathbf{K}(\mathbf{X}^*, \mathbf{X}^*) \end{bmatrix} \right) \quad (6)$$

式中, $\mathbf{K}(\mathbf{X}, \mathbf{X}^*)$ 表示测试输出样本 \mathbf{X}^* 和训练输出样本 \mathbf{X} 之间的协方差矩阵, $\mathbf{K}(\mathbf{X}^*, \mathbf{X}^*)$ 为测试输出样本 \mathbf{X}^* 自身的协方差矩阵.

随后, 通过最大似然估计方法寻找最优的超参数, 从而最终确定高斯过程模型. 当给定输入 \mathbf{X}^* , 其通过训练集中的输入 \mathbf{X} 和其观测目标输出值 \mathbf{Y} , 预测出 $\hat{f}(\mathbf{X}^*)$ 概率最大的预测后验分布, 即

$$\hat{f}(\mathbf{X}^* | \mathbf{X}^*, \mathbf{X}, \mathbf{Y}) \sim N(\mu, \Sigma) \quad (7)$$

其中

$$\mu = \mathbf{K}(\mathbf{X}^*, \mathbf{X})(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I})^{-1} \mathbf{Y} \quad (8)$$

$$\Sigma = \mathbf{K}(\mathbf{X}^*, \mathbf{X}^*) -$$

$$\mathbf{K}(\mathbf{X}^*, \mathbf{X})(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I})^{-1} \mathbf{K}(\mathbf{X}, \mathbf{X}^*) \quad (9)$$

1.2 RVEA 算法

RVEA 算法^[9] 是 2016 年 Cheng 等针对高维多目标优化问题提出的基于分解的进化算法. 不同于最初提出的基于分解的多目标进化算法 MOEA/D^[8], RVEA 中使用一组自适应的参考向量, 同时提出了角度惩罚距离 (Angle penalized distance, APD) 作为环境选择策略. 在 RVEA 中, 参考向量根据目标函数值范围的不同调整其分布,

$$\mathbf{V}_{t+1, i} = \frac{\mathbf{V}_{0, i} \circ (\mathbf{Z}_{t+1}^{\max} - \mathbf{Z}_{t+1}^{\min})}{\|\mathbf{V}_{0, i} \circ (\mathbf{Z}_{t+1}^{\max} - \mathbf{Z}_{t+1}^{\min})\|} \quad (10)$$

式中, $\mathbf{V}_{0, i}$ 表示初始化时第 i 个均匀分布的参考向量, \mathbf{Z}_{t+1}^{\max} 和 \mathbf{Z}_{t+1}^{\min} 分别表示迄今为止每个目标上的最大和最小函数值构成的向量.

在 RVEA 中, 角度惩罚距离 APD 的计算方法如下:

$$d_{t, i, j} = (1 + P(\theta_{t, i, j})) \cdot \left\| \mathbf{F}'(\mathbf{x}_i(t)) \right\| \quad (11)$$

其中, $d_{t, i, j}$ 表示第 i 个个体在第 t 代时在第 j 个参考向量上的 APD 值, $\theta_{t, i, j}$ 表示第 t 代个体 i 的目标函数值与第 j 个参考向量之间的夹角. $P(\theta_{t, i, j})$ 为惩罚函数, 其计算公式为

$$P(\theta_{t,i,j}) = M \cdot \left(\frac{t}{t_{\max}}\right)^\alpha \cdot \frac{\theta_{t,i,j}}{\gamma_{v_{t,j}}} \quad (12)$$

式中

$$\gamma_{v_{t,j}} = \min_{i \in \{1, \dots, N\}, i \neq j} \langle \mathbf{V}_{t,i}, \mathbf{V}_{t,j} \rangle \quad (13)$$

M 和 N 分别表示目标数和参考向量数, t_{\max} 为种群最大进化代数, $\gamma_{v_{t,j}}$ 表示参考向量 $\mathbf{V}_{t,i}$ 与其他参考向量之间的最小角度, α 是控制惩罚函数速率的参数. 式 (11) 中, $\mathbf{F}'(\mathbf{x}_i(t))$ 表示第 t 代的第 i 个解归一化之后的目标函数值, 其归一化方法为:

$$\mathbf{F}'(\mathbf{x}_i(t)) = \mathbf{F}(\mathbf{x}_i(t)) - \mathbf{F}^* \quad (14)$$

式中, $\mathbf{F}(\mathbf{x}_i(t))$ 是个体 i 在 t 代的一个目标函数值, \mathbf{F}^* 表示由每个目标最小值组成的向量.

2 代理模型辅助的计算费时进化高维多目标优化

模型的用途以及选择个体真实计算的模型填充准则对于代理模型辅助的进化算法在计算资源有限的情况下寻找计算费时问题的最优解集是非常重要的^[31-33]. 随着目标空间维度的增加, 对计算费时问题的求解算法在搜索效率上有了更高的要求. 由于常见的求解高维多目标的优化算法需要大量的目标函数评价次数, 使其在求解这类费时问题时受到了很大地限制. 使用计算廉价的代理模型代替计算费时的目标函数评价是求解计算费时多目标优化问题的常见方法. 然而, 模型的使用方法会极大地影响算法的搜索效率, 特别是当目标空间维度增加时, 由于各个目标均为估值, 一个目标估值错误将会导致优化算法朝着错误的方向进行搜索, 从而严重影响最优解集的寻找. 另一方面, 在搜索最优解集的过程中, 选择若干个体进行真实评价也是非常重要的. 这些真实计算的个体不仅用于更新模型, 以提高模型的估值准确度, 同时也是潜在的非支配候选解. 鉴于高斯过程模型不仅能够提供估值还能够提供估值不确定度, 本文提出使用高斯过程模型来估计目标函数值, 以辅助计算费时的高维多目标问题的优化 (Surrogate-assisted expensive evolutionary many-objective optimization, SAEExp-EMO). 在该方法中, 为了提高搜索效率, 首先将各个代理模型作为优化目标, 使用对求解高维多目标问题具有较好优化性能的 RVEA 算法对代理模型进行最优解集的搜索, 找到具有较好收敛性能的解, 从而能够提供较好的供真实计算个体选择的候选解集. 算法 1 给出了本文方法的伪代码. 算法 1 分为 3 个部分: 第 1 部分为初始化阶段 (1 ~ 3 行), 主要是用拉丁

超立方抽样方法采样若干个体以供初始代理模型的训练, 同时获得目前的非支配解集. 第 2 部分是训练代理模型并对其进行最优解集的搜索 (5 ~ 6 行), 第 3 部分是通过填充准则策略从搜索到的代理模型最优解集中选择个体进行真实计算. 第 2 部分和第 3 部分交替运行, 直到满足停止条件, 即达到最大评价次数为止.

算法 1. 代理模型辅助的计算费时进化多目标优化 (SAExp-EMO)

输入: 最大评价次数 $MaxOE$; 评价次数 $OE = 0$;

输出: 最优解集 PS

- 1) 使用拉丁超立方体采样 N_s 个解, 并对其使用真实的目标函数进行真实计算. 存放数据库 Arc 中;
- 2) $OE = N_s$;
- 3) 非支配排序 Arc 中的个体, 并将非支配解存入集合 PS 中;
- 4) While $OE \leq MaxOE$;
- 5) 利用 Arc 中的样本信息为每个目标函数训练代理模型;
- 6) 利用 RVEA 对模型进行最优解集的搜索 (详见算法 2);
- 7) 利用填充准则选择个体使用真实目标函数进行真实计算, 更新 PS 并存入 Arc 中 (参见算法 3);
- 8) $OE = OE + 1$;
- 9) End while.

2.1 模型最优解集的搜索

当模型能够很好地拟合原目标函数时, 搜索代理模型得到的最优解集即为原优化问题的最优解集, 并且能够大量地节省求解问题的计算时间. 因此, 为了提高对费时高维多目标优化问题的求解效率, 在 SAEExp-EMO 中, 通过对高斯过程模型进行最优解集的搜索使种群能够落到目标函数值较好的潜在区域, 以供真实计算个体的选择. 任何求解高维多目标优化问题的算法都可以用来实现对代理模型最优解集的搜索. RVEA^[9] 是 Cheng 等在 2016 年提出的基于分解的求解高维多目标优化问题的有效方法, 其提供了角度惩罚距离用于在高维目标空间更好地选择下一代父代种群. 同时, 自适应参考向量可以更均匀的取到最优解集. 因此, 本文选用 RVEA 对高斯过程模型进行非支配最优解集的搜索. $|pop(t)|$ 表示当前 t 代种群大小, 算法 2 给出了搜索模型最优解集的伪代码.

算法 2. 模型最优解集的搜索

输入: 模型的最大评价次数 L ; 数据库 Arc ; 高斯过程模型 $GP = (GP_1, GP_2, \dots, GP_M)$;

输出: 当前种群 $pop(t)$;

- 1) $t = 0$;
- 2) 将 Arc 中所有个体作为 RVEA 的初始种群 $pop(t)$;
- 3) $l = |Arc|$;
- 4) While $l \leq L$;
- 5) 产生参考向量;
- 6) 父代种群 $P(t)$ 通过二进制交叉、多项式变异产生子代种群 $Q(t)$;
- 7) 用 GP 模型对子代个体进行评价;
- 8) 将父代和子代个体分配给其最近的参考向量;
- 9) 利用 APD 指标进行环境选择得到下一代父代种群 $pop(t)$;
- 10) 自适应参考向量;
- 11) $l = l + |pop(t)|$;
- 12) $t = t + 1$;
- 13) End while;
- 14) 输出 $pop(t)$.

2.2 改进的填充准则

模型管理是代理模型辅助的优化算法中最重要的一环, 由于真实计算的个体不仅要用于模型的更新以提高模型的估值准确度, 同时其也是潜在的最优非支配解集中的候选解, 所以填充准则的选择, 将直接影响最终获得的优化结果好坏. 常见的针对高斯过程模型提出的填充准则是针对单目标优化问题的, 不能直接用于多目标优化问题. 考虑到 RVEA 中角度惩罚距离指标可以同时衡量一个个体的收敛性和多样性, 故本文考虑将目标函数估值的角度惩罚距离值作为个体的性能指标. 其角度惩罚距离期望值提高越大, 说明个体的整体性能提高较大, 因此选择这类个体进行真实的目标函数计算有利于加快费时优化问题最优解集的搜索. 另一方面, 若个体估值的总体不确定度较大, 即各个目标估值不确定的累加和较大时, 表明该个体的估值不可信. 因此, 对这类个体进行真实的目标函数计算并用于模型的更新将有利于代理模型准确度的提高. 基于以上分析, 本文针对高维多目标优化问题, 提出一种改进的期望提高获取函数, 以选择具有较高价值的个体进行真实计算. 式 (15) 给出了改进的期望提高获取函数.

$$EI(\mathbf{x}_i) = (d^* - d_{t,i,j})\Phi\left(\frac{d^* - d_{t,i,j}}{\tilde{s}(\mathbf{x}_i)}\right) + \tilde{s}(\mathbf{x}_i)\varphi\left(\frac{d^* - d_{t,i,j}}{\tilde{s}(\mathbf{x}_i)}\right) \quad (15)$$

其中, $d_{t,i,j}$ 为第 i 个个体在 t 代相对于第 j 个参考向量的 APD 值, d^* 表示 Arc 中所有个体具有的最

小的 APD 值, 即

$$d^* = \min_{i=1, 2, \dots, |Arc|} \{d_{t,i,j}\} \quad (16)$$

$s(\mathbf{x}_i)$ 为个体 i 各个目标估值不确定度的平均值, 即

$$\tilde{s}(\mathbf{x}_i) = \frac{\sum_{k=1}^M s_k(\mathbf{x}_i)}{M} \quad (17)$$

其中, $s_k(\mathbf{x}_i)$ 表示第 i 个个体在第 k 个目标上的估值不确定度.

算法 3. 改进的填充准则

输入: 当前模型搜索的最后一代 $pop(t)$; 参考向量 \mathbf{V} ;

输出: 使用真实目标函数计算的解 \mathbf{x}^*

- 1) $pop(t)$ 中每个个体 \mathbf{x}_i 寻找其最近的参考向量 \mathbf{V}_j ;
- 2) 计算个体与最近参考向量的 APD 值 $d_{t,i,j}$;
- 3) 确定个体每个目标上的估值不确定度, 将这些估值不确定度的平均值作为个体估值的不确定度;
- 4) 根据式 (15) 计算 $pop(t)$ 中各个解的 EI 值;
- 5) 选择 $pop(t)$ 中 EI 值最大的个体 \mathbf{x}^* .

算法 3 给出了改进的填充准则的伪代码. 在算法 3 中, 将模型搜索最优解集的最后一代种群个体分配给其最近的参考向量, 并计算相应的角度惩罚距离值. 同时根据各个目标估值的不确定计算个体的整体估值不确定度 (所有目标估值不确定度的平均). 随后根据个体的角度惩罚距离和平均不确定度计算其期望提高值, 从种群中选择期望值最大的个体进行真实计算.

3 实验验证

为验证本文方法的有效性, 本文在 7 个 DTLZ 基准问题^[34] 上进行了测试, 每个问题分别测试了 3、4、6、8、10 个目标. 并和没有代理模型辅助的进化算法 RVEA 以及具有代表性的用于求解计算费时多目标优化问题的代理模型辅助算法, K-RVEA^[17], CSEA^[23] 和 ParEGO^[22] 进行了对比. 其中 K-RVEA 同样为每个目标建立代理模型并搜索模型的最优, 和本文不同的是, 在 K-RVEA 中优化模型的最后一代种群进行了聚类, 并根据和固定参考向量相关联的个体数差异选择 APD 最小或者不确定度最大的若干个体进行真实评价. CSEA 是基于神经网络的求解费时问题的多目标优化问题, 通过对个体的分类选择若干有前途的个体进行真实计算. ParEGO 使用切比雪夫函数将多目标优化问题转换成单目标优化问题, 并对单目标优化问题建立高斯过程模型, 利用获取函数选择个体进行真实计算.

3.1 参数设置

实验中,所有算法的最大目标函数评价次数均设置为 300 次.根据文献 [34] 给出的 DTLZ 测试函数的定义,问题的维度为 $K + M - 1$, M 为目标数,DTLZ1 和 DTLZ7 测试函数 K 的取值分别为 5 和 20,DTLZ2-6 测试函数 K 取值为 10.所有算法都独立运行 20 次,本文对比算法的结果都在 PlatEMO 上运行得到.为了公平比较,除了初始样本大小,对比算法中搜索算法的参数均采用原文给出的参数,即交叉 η_c 和变异 η_m 算子均为 20,交叉概率 p_c 设为 1.0,变异概率 p_m 设为 $1/D$,其中 D 为决策变量的维度.在 K-RVEA、CSEA、ParEGO 中,初始采样大小均为 $11D - 1$,其测试问题维度为固定的 10 维.而本文测试问题维度是不固定的,决策空间大小由目标函数个数决定,因此当目标维度增高,决策空间维度也随之增大.由于 $11D - 1$ 占用大量评价次数,优化代数减少不利于算法的寻优.故本实验中 K-RVEA、CSEA、ParEGO 和 SAExp-EMO 初始样本设置都为 $N_s = 5D - 1$.利用置信度 $\sigma = 0.05$ 的 Wilcoxon 秩和检验方法来判断本文算法和其他算法获得的解集之间的差异性.符号 +、- 和 \approx 分别表示所比较的算法性能比本文 SAExp-EMO 算法好、差和没有明显的差异.

3.2 性能指标

反转世代距离评价指标 (Inverted generational distance, IGD)^[35] 是一个综合性能评价指标,通常被用作衡量求解多目标优化问题方法的性能指标.它主要通过计算每个在真实 Pareto 前沿面上的点 (个体) 到算法获取的非支配面上个体之间的最小欧氏距离和,来评价算法的收敛性能和分布性能.值越小,算法的综合性能越好.IGD 的计算公式如下:

$$IGD(P, Q) = \frac{\sum_{v \in P} \text{dist}(v, Q)}{|P|} \quad (18)$$

其中, P 和 Q 分别为均匀分布在真实 Pareto 面上的点集和算法获得的最优 Pareto 面. $\text{dist}(v, Q)$ 为 P 中个体 v 到 Pareto 面 Q 的最小欧几里得距离.因此,IGD 是通过计算真实 Pareto 面上点集到获取的非支配面的最小欧氏距离的平均值来评价算法的综合性能.当 P 中个体数足够多时,其解就会均匀的覆盖真实 Pareto 面,本文中 $|P|$ 设置为 10000.

3.3 实验结果及分析

3.3.1 搜索模型最优解集的最大评价次数 L

搜索模型最优解集的评价次数会影响算法对计

算费时问题的寻优能力,评价次数过少,算法还没找到模型的最优解集,评价次数过多,搜索可能会偏离真实的问题最优.为此,本文分别使用 $L = 0$, $L = 500 \times M$, $L = 1000 \times M$, $L = 1500 \times M$, $L = 2000 \times M$, $L = 2500 \times M$ 和 $L = 3000 \times M$ 模型评价次数对 DTLZ1 和 DTLZ2 测试问题上进行了算法性能进行了测试,其中 M 为问题的目标数.在实验中,目标函数分别设置为 3、6、8、10 进行了测试.图 1 给出了不同 L 值下获得的 IGD 值.由图 1 可以看出,当搜索模型的最大评价次数为 $L = 1000 \times M$ 时算法在这两个函数上的性能最好.为此,在本文的方法中,搜索模型最优的停止条件为模型评价次数达到 $L = 1000 \times M$.

3.3.2 不同算法中的实验结果

为了验证本文算法的有效性,本文算法和 RVEA, ParEGO, K-RVEA 以及 CSEA 在 3、4、6、8、10 个目标的 DTLZ1~7 测试问题上进行了实验结果对比.需要注意的是 ParEGO 算法是针对目标函数个数不超过 4 个的多目标优化问题提出的,因此本文单独将 ParEGO 和 SAExp-EMO 方法在 3 个和 4 个目标的 DTLZ1 ~ 7 测试函数上进行了对比.表 1 给出了 SAExp-EMO 和 ParEGO 获得的 IGD 平均值的结果,其中最好结果以粗体表示.由表 1 可以看出,本文提出的 SAExp-EMO 方法能够在 3 个目标和 4 个目标的 DTLZ1 ~ 7 测试函数集上获得更好或者一样的 IGD 值,说明 SAExp-EMO 算法在收敛性和多样性上具有更好的性能.

RVEA、K-RVEA 和 CSEA 均是针对高维多目标提出的优化算法,其中 RVEA 无代理模型辅助,而 K-RVEA 和 CSEA 均为代理模型辅助的高维多目标优化方法.表 2 给出了不同算法在 3、4、6、8、10 个目标的 DTLZ 上的测试结果,其中最好结果以粗体表示.由表 2 可以看出,相比于无代理模型辅助的 RVEA,本文的 SAExp-EMO 在所有 DTLZ 测试函数上均获得了性能较好的解,只有在 4 个目标的 DTLZ4 上获得的结果和 RVEA 无差别.相比于代理模型辅助的 K-RVEA,本文方法在 25 个问题上获得了较好解,在测试问题 DTLZ1~7 中,除 DTLZ4 外,本文算法的结果都优于 K-RVEA.这是因为 DTLZ4 的 Pareto 前沿是一条退化的覆盖在目标空间中一个子空间曲线,而 SAExp-EMO 在使用参考向量搜索模型最优解集的过程中,有大量没有分配到解的空参考向量,这使得收敛到 Pareto 前沿的求解过程缓慢,而 CSEA 算法在 DTLZ4 上取得了最好的效果,主要归因于 CSEA 中基于径

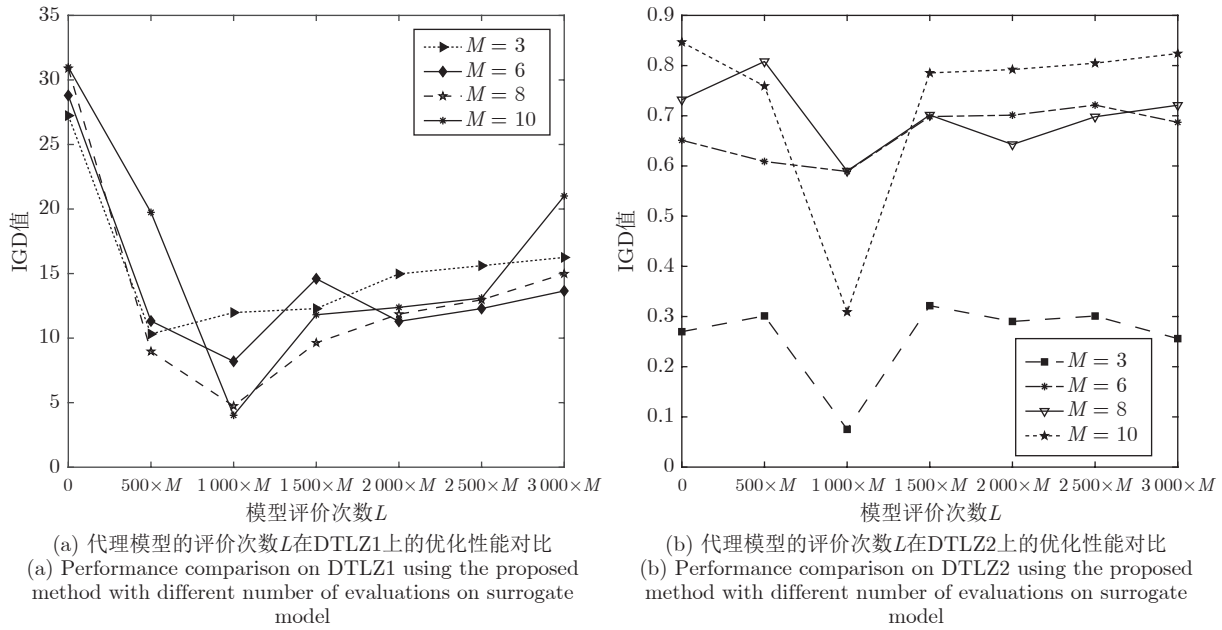


图 1 不同模型评价次数下算法的性能结果对比图

Fig.1 Performance comparison of the proposed method with different number of evaluations on surrogate model

表 1 SAEsp-EMO 和 ParEGO 在 3 个和 4 个目标函数的 DTLZ 测试问题上获得的平均 IGD 统计结果

Table 1 Average IGD statistical results of SAEsp-EMO and ParEGO on DTLZ test problems of 3 and 4 objective functions

测试问题	目标数	ParEGO	SAExp-EMO
DTLZ1	3	$4.84 \times 10^1 (8.51 \times 10^0) -$	$1.09 \times 10^1 (4.16 \times 10^0)$
	4	$5.49 \times 10^1 (1.09 \times 10^1) -$	$1.25 \times 10^1 (5.06 \times 10^1)$
DTLZ2	3	$4.76 \times 10^{-1} (3.63 \times 10^{-2}) -$	$1.72 \times 10^{-1} (4.60 \times 10^{-2})$
	4	$5.77 \times 10^{-1} (2.98 \times 10^{-2}) -$	$3.65 \times 10^{-1} (4.40 \times 10^{-1})$
DTLZ3	3	$4.61 \times 10^0 (5.38 \times 10^1) -$	$1.65 \times 10^2 (6.07 \times 10^1)$
	4	$4.45 \times 10^2 (7.57 \times 10^1) -$	$2.40 \times 10^2 (1.19 \times 10^2)$
DTLZ4	3	$7.80 \times 10^{-1} (7.38 \times 10^{-2}) \approx$	$5.59 \times 10^{-1} (4.80 \times 10^{-2})$
	4	$8.92 \times 10^{-1} (9.00 \times 10^{-1}) -$	$7.12 \times 10^{-1} (1.48 \times 10^{-1})$
DTLZ5	3	$3.74 \times 10^{-1} (7.78 \times 10^{-2}) -$	$4.01 \times 10^{-2} (7.00 \times 10^{-2})$
	4	$4.09 \times 10^{-1} (4.52 \times 10^{-2}) -$	$7.80 \times 10^{-2} (0.00 \times 10^0)$
DTLZ6	3	$8.04 \times 10^0 (2.44 \times 10^{-1}) -$	$3.63 \times 10^0 (2.61 \times 10^0)$
	4	$8.16 \times 10^0 (2.52 \times 10^{-1}) -$	$3.84 \times 10^0 (4.61 \times 10^{-1})$
DTLZ7	3	$7.28 \times 10^0 (2.16 \times 10^0) -$	$7.70 \times 10^{-1} (1.35 \times 10^{-1})$
	4	$1.11 \times 10^1 (7.97 \times 10^{-1}) -$	$1.09 \times 10^0 (3.18 \times 10^{-1})$
+/-/ \approx		0/13/1	

向空间划分的更新参考点的策略. 从表 2 可以看出, SAEsp-EMO 算法在 10 个目标的 DTLZ1、DTLZ2、DTLZ3 和 DTLZ7 的结果明显优于 K-RVEA, 这归因于 K-RVEA 模型最优解集搜索的频率是固定的, 在高维的决策空间中, 会导致种群搜索陷入局部某块区域, 不利于找到有前途的候选解. 与

CSEA 相比, SAEsp-EMO 在 26 个问题上获得了较好解, 只有在 3 个问题上没有比过 CSEA, 表明了本文算法在求解高维多目标优化问题上具有较好的性能.

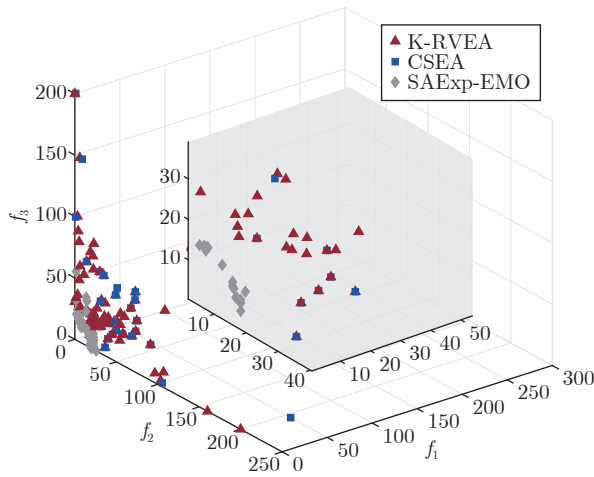
为进一步查看最后非支配解集的分布, 图 2(a) 给出了各个算法在 3 个目标的 DTLZ1 测试问题上

表 2 SAExp-EMO、RVEA、K-RVEA 和 CSEA 得到的平均 IGD 值
Table 2 Average IGD values obtained by SAExp-EMO, RVEA, K-RVEA and CSEA

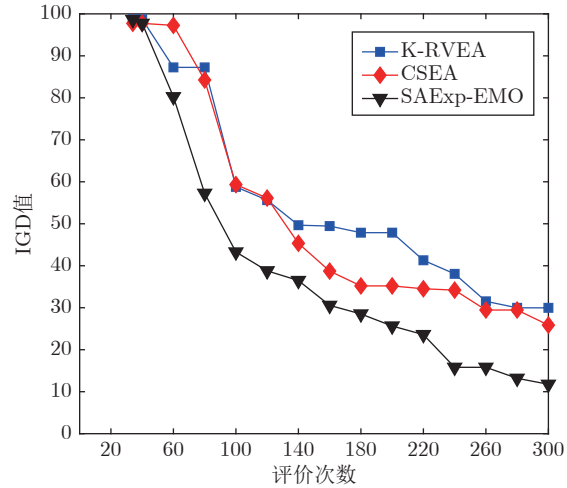
测试问题	目标数	RVEA	K-RVEA	CSEA	SAExp-EMO
DTLZ1	3	$3.65 \times 10^1 (1.10 \times 10^1) -$	$2.48 \times 10^1 (8.56 \times 10^0) -$	$1.97 \times 10^1 (5.82 \times 10^0) -$	$1.33 \times 10^1 (4.53 \times 10^0)$
	4	$3.18 \times 10^1 (1.03 \times 10^1) -$	$3.01 \times 10^1 (1.18 \times 10^1) -$	$1.71 \times 10^1 (5.31 \times 10^0) -$	$1.35 \times 10^1 (5.03 \times 10^0)$
	6	$2.96 \times 10^1 (8.16 \times 10^0) -$	$3.18 \times 10^1 (6.94 \times 10^0) -$	$1.43 \times 10^1 (6.68 \times 10^0) -$	$1.15 \times 10^1 (6.29 \times 10^0)$
	8	$2.00 \times 10^1 (9.31 \times 10^0) -$	$3.22 \times 10^1 (1.12 \times 10^1) \approx$	$1.44 \times 10^1 (6.01 \times 10^0) -$	$1.17 \times 10^1 (4.46 \times 10^0)$
	10	$2.15 \times 10^1 (8.45 \times 10^0) -$	$2.48 \times 10^1 (9.28 \times 10^0) -$	$1.45 \times 10^1 (5.70 \times 10^0) -$	$1.28 \times 10^1 (5.45 \times 10^0)$
DTLZ2	3	$4.09 \times 10^{-1} (3.22 \times 10^{-2}) -$	$2.66 \times 10^{-1} (4.88 \times 10^{-2}) -$	$2.69 \times 10^{-1} (1.13 \times 10^{-1}) -$	$1.38 \times 10^{-1} (6.13 \times 10^{-2})$
	4	$5.16 \times 10^{-1} (3.61 \times 10^{-2}) -$	$3.95 \times 10^{-1} (4.94 \times 10^{-2}) -$	$4.76 \times 10^{-1} (1.04 \times 10^{-1}) -$	$3.27 \times 10^{-1} (5.53 \times 10^{-2})$
	6	$6.97 \times 10^{-1} (6.73 \times 10^{-2}) -$	$5.93 \times 10^{-1} (4.96 \times 10^{-2}) -$	$5.76 \times 10^{-1} (4.01 \times 10^{-2}) \approx$	$6.15 \times 10^{-1} (4.93 \times 10^{-2})$
	8	$7.93 \times 10^{-1} (3.69 \times 10^{-2}) -$	$6.54 \times 10^{-1} (4.95 \times 10^{-2}) -$	$7.57 \times 10^{-1} (3.52 \times 10^{-2}) -$	$5.45 \times 10^{-1} (2.42 \times 10^{-1})$
	10	$9.54 \times 10^{-1} (5.16 \times 10^{-2}) -$	$7.36 \times 10^{-1} (4.59 \times 10^{-2}) -$	$8.44 \times 10^{-1} (5.65 \times 10^{-2}) -$	$6.08 \times 10^{-1} (3.09 \times 10^{-1})$
DTLZ3	3	$4.18 \times 10^2 (6.66 \times 10^1) -$	$3.38 \times 10^2 (7.51 \times 10^1) -$	$2.12 \times 10^2 (4.37 \times 10^1) -$	$1.13 \times 10^2 (2.96 \times 10^1)$
	4	$4.17 \times 10^2 (7.54 \times 10^1) -$	$3.56 \times 10^2 (7.56 \times 10^1) -$	$2.17 \times 10^2 (4.94 \times 10^1) -$	$1.26 \times 10^2 (6.16 \times 10^1)$
	6	$3.85 \times 10^2 (7.05 \times 10^1) -$	$3.45 \times 10^2 (7.90 \times 10^1) -$	$2.09 \times 10^2 (5.44 \times 10^1) -$	$1.46 \times 10^2 (7.89 \times 10^1)$
	8	$3.57 \times 10^2 (7.05 \times 10^1) -$	$3.38 \times 10^2 (5.74 \times 10^1) -$	$2.08 \times 10^2 (5.09 \times 10^1) -$	$1.49 \times 10^2 (7.88 \times 10^1)$
	10	$3.77 \times 10^2 (1.02 \times 10^2) -$	$3.24 \times 10^2 (7.92 \times 10^1) -$	$2.18 \times 10^2 (5.85 \times 10^1) \approx$	$1.10 \times 10^2 (2.87 \times 10^1)$
DTLZ4	3	$5.58 \times 10^{-1} (6.90 \times 10^{-2}) -$	$4.17 \times 10^{-1} (1.12 \times 10^{-1}) \approx$	$7.22 \times 10^{-1} (1.53 \times 10^{-1}) -$	$4.81 \times 10^{-1} (1.40 \times 10^{-1})$
	4	$6.96 \times 10^{-1} (8.80 \times 10^{-2}) \approx$	$5.46 \times 10^{-1} (1.13 \times 10^{-1}) +$	$5.43 \times 10^{-1} (1.02 \times 10^{-1}) +$	$6.64 \times 10^{-1} (1.45 \times 10^{-1})$
	6	$8.53 \times 10^{-1} (8.13 \times 10^{-2}) -$	$6.84 \times 10^{-1} (8.64 \times 10^{-2}) +$	$5.74 \times 10^{-1} (1.01 \times 10^{-1}) \approx$	$8.52 \times 10^{-1} (8.99 \times 10^{-2})$
	8	$9.32 \times 10^{-1} (7.75 \times 10^{-2}) -$	$8.34 \times 10^{-1} (9.14 \times 10^{-2}) +$	$7.39 \times 10^{-1} (3.42 \times 10^{-2}) +$	$8.36 \times 10^{-1} (1.76 \times 10^{-1})$
	10	$1.03 \times 10^0 (7.03 \times 10^{-2}) -$	$8.89 \times 10^{-1} (6.96 \times 10^{-2}) \approx$	$8.12 \times 10^{-1} (4.54 \times 10^{-2}) +$	$8.17 \times 10^{-1} (2.43 \times 10^{-1})$
DTLZ5	3	$3.45 \times 10^{-1} (4.41 \times 10^{-2}) -$	$1.81 \times 10^{-1} (4.44 \times 10^{-2}) -$	$1.46 \times 10^{-1} (4.29 \times 10^{-2}) -$	$3.84 \times 10^{-2} (8.64 \times 10^{-3})$
	4	$3.79 \times 10^{-1} (7.42 \times 10^{-2}) -$	$1.90 \times 10^{-1} (3.12 \times 10^{-2}) -$	$2.00 \times 10^{-1} (4.29 \times 10^{-2}) -$	$6.98 \times 10^{-2} (1.43 \times 10^{-2})$
	6	$4.28 \times 10^{-1} (6.52 \times 10^{-2}) -$	$2.29 \times 10^{-1} (3.40 \times 10^{-1}) \approx$	$2.17 \times 10^{-1} (7.87 \times 10^{-1}) -$	$1.30 \times 10^{-1} (4.04 \times 10^{-2})$
	8	$4.26 \times 10^{-1} (5.83 \times 10^{-2}) -$	$2.19 \times 10^{-1} (4.87 \times 10^{-2}) -$	$2.43 \times 10^{-1} (6.08 \times 10^{-2}) -$	$8.31 \times 10^{-2} (2.32 \times 10^{-2})$
	10	$4.06 \times 10^{-1} (1.02 \times 10^{-1}) -$	$2.23 \times 10^{-1} (5.87 \times 10^{-2}) \approx$	$2.54 \times 10^{-1} (5.35 \times 10^{-2}) -$	$9.97 \times 10^{-2} (4.07 \times 10^{-2})$
DTLZ6	3	$7.94 \times 10^0 (2.75 \times 10^{-1}) -$	$4.42 \times 10^0 (5.40 \times 10^{-1}) -$	$4.54 \times 10^0 (5.84 \times 10^{-1}) -$	$3.07 \times 10^0 (7.11 \times 10^{-1})$
	4	$8.02 \times 10^0 (2.61 \times 10^{-1}) -$	$4.35 \times 10^0 (4.66 \times 10^{-1}) -$	$6.99 \times 10^0 (7.87 \times 10^{-1}) -$	$3.46 \times 10^0 (4.82 \times 10^{-1})$
	6	$8.19 \times 10^0 (3.42 \times 10^{-1}) -$	$4.58 \times 10^0 (7.79 \times 10^{-1}) -$	$7.11 \times 10^0 (1.76 \times 10^{-1}) -$	$4.19 \times 10^0 (6.93 \times 10^{-1})$
	8	$8.18 \times 10^0 (2.75 \times 10^{-1}) -$	$5.78 \times 10^0 (4.49 \times 10^{-1}) -$	$7.21 \times 10^0 (5.28 \times 10^{-1}) \approx$	$3.60 \times 10^0 (5.11 \times 10^{-1})$
	10	$8.22 \times 10^0 (4.07 \times 10^{-1}) -$	$6.32e \times 10^0 (6.35 \times 10^{-1}) \approx$	$7.44 \times 10^0 (4.18 \times 10^{-1}) \approx$	$3.95 \times 10^0 (1.16 \times 10^0)$
DTLZ7	3	$6.85 \times 10^0 (7.29 \times 10^{-1}) -$	$1.15 \times 10^0 (1.69 \times 10^0) -$	$4.02 \times 10^0 (4.82 \times 10^0) -$	$5.36 \times 10^{-1} (2.26 \times 10^{-1})$
	4	$8.81 \times 10^0 (1.31 \times 10^0) -$	$2.14 \times 10^0 (3.24 \times 10^0) \approx$	$7.54 \times 10^0 (9.33 \times 10^{-1}) \approx$	$6.92 \times 10^{-1} (1.37 \times 10^{-1})$
	6	$1.29 \times 10^1 (1.44 \times 10^0) -$	$3.49 \times 10^0 (2.76 \times 10^0) -$	$1.36 \times 10^1 (1.65 \times 10^0) -$	$1.13 \times 10^0 (2.42 \times 10^{-1})$
	8	$1.72 \times 10^1 (2.18 \times 10^0) -$	$4.18 \times 10^0 (2.18 \times 10^0) -$	$2.26 \times 10^1 (2.27 \times 10^0) -$	$6.82 \times 10^{-1} (1.38 \times 10^{-1})$
	10	$2.18 \times 10^1 (3.56 \times 10^0) -$	$7.83 \times 10^0 (3.32 \times 10^0) -$	$2.86 \times 10^1 (2.30 \times 10^0) -$	$1.19 \times 10^0 (5.68 \times 10^{-1})$
+/-/ \approx	0/34/1	3/25/7	3/26/6		

找到的最优非支配解集. 三角形、正方形、菱形分别表示算法 K-RVEA、CSEA 和 SAExp-EMO 所获得最优非支配解集. 由图 2(a) 可知, SAExp-EMO 所获得非支配解的目标函数值比 K-RVEA 和 CSEA 都小. 在相同的评价次数下, 相比于 K-RVEA, CSEA 算法 SAExp-EMO 获得的种群更靠近真实的 Pareto 前沿, 说明 SAExp-EMO 算法有更快和更好的收敛

性, 同时从解的分布看, SAExp-EMO 所找到的目标空间具有更好解的分布性. 图 2(b) 为 K-RVEA, CSEA, 以及 SAExp-EMO 在 3 个目标 DTLZ1 上独立运行 20 次获得的 IGD 均值的收敛图. 由图 2(b) 可以看出, 在相同的评价次数下, SAExp-EMO 获得了比 K-RVEA 和 CSEA 更好的 IGD 值, 同时 SAExp-EMO 算法具有更快的收敛速度.



(a) K-RVEA, CSEA以及SAEExp-EMO在3个目标DTLZ1上获得的非支配面
(a) Non-dominated front obtained by K-RVEA, CSEA, and SAEExp-EMO on the three-objective DTLZ1, respectively



(b) K-RVEA, CSEA以及SAEExp-EMO在3个目标DTLZ1上获得的IGD均值的收敛图
(b) Convergence trends of mean IGD values obtained by K-RVEA, CSEA and SAEExp-EMO on three-objective DTLZ1, respectively

图 2 不同算法在 DTLZ1 上的性能结果对比

Fig.2 Performance comparison of different methods on three-objective DTLZ1 problem

4 结束语

针对代理模型辅助的计算费时多目标问题的优化, 本文提出了一种新的填充准则, 基于角度惩罚距离以及目标估值的平均不确定度, 改进期望提高计算方式, 用于选择使用真实目标函数计算的个体. 算法在 3、4、6、8 和 10 个目标的 DTLZ 基准测试问题上进行了测试, 和其他有代表性的代理模型辅助的多目标进化算法的实验结果相比, 本文所提方法具有更好的求解性能.

目前, 高斯过程模型面临最大的问题是当决策空间维度增加时, 训练时间会呈现指数级增长, 导致在决策空间高维上很难使用. 为此, 如何求解决策空间高维的多目标计算费时优化问题, 需要进一步展开研究.

References

- Zhang Han, Yang Ji-Bin, Zhang Ji-Ye, Song Peng-Yun, Xu Xiao-Hui. Pareto-based multi-objective optimization of energy management for fuel cell tramway. *Acta Automatica Sinica*, 2019, **45**(12): 2378–2392
(张晗, 杨继斌, 张继业, 宋鹏云, 徐晓惠. 燃料电池有轨电车能量管理Pareto多目标优化. *自动化学报*, 2019, **45**(12): 2378–2392)
- Coello Coello C A, Brambila G, Figueroa Gamboa S, Gamboa J F, Tapia M G C, Gómez R H. Evolutionary multiobjective optimization: open research areas and some challenges lying ahead. *Complex & Intelligent Systems*, 2020, **6**(1): 221–236
- Wang Liu-Jing, Zhang Gui-Jun, Zhou Xiao-Gen. Strategy self-adaptive differential evolution algorithm based on state estimation feedback. *Acta Automatica Sinica*, 2020, **46**(4): 752–766
(王柳静, 张贵军, 周晓根. 基于状态估计反馈的策略自适应差分进化算法. *自动化学报*, 2020, **46**(4): 752–766)
- Feng Wen-Qing, Gong Dun-Wei. Multi-objective evolutionary optimization with objective space partition based on online perception of pareto front. *Acta Automatica Sinica*, 2020, **46**(8): 1628–1643
(封文清, 巩敦卫. 基于在线感知pareto前沿划分目标空间的多目标进化优化. *自动化学报*, 2020, **46**(8): 1628–1643)
- Deb K, Pratap A, Genetic K, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGAII. *IEEE Transactions on Evolutionary Computation*, 2002, **6**(2): 182–197
- Li San-Yi, Wang Yan-Feng, Qiao Jun-Fei, Huang Jin-Hua. A regional local search strategy for NSGAII algorithm. *Acta Automatica Sinica*, 2020, **46**(12): 2617–2627
(栗三一, 王延峰, 乔俊飞, 黄金花. 一种基于区域局部搜索的NSGAII算法. *自动化学报*, 2020, **46**(12): 2617–2627)
- Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the performance of the strength pareto evolutionary algorithm. *Technical Report Gloriosastrasse*, 2001, **3242**(4): 742–751
- Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, **11**(6): 712–731
- Cheng R, Jin Y, Olhofer M, Sendhoff B. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2016, **20**(5): 773–791
- Zitzler E, Künzli S. Indicator-based selection in multi-objective search. In *Proceedings of: International Conference on Parallel Problem Solving from Nature, Berlin, Heidelberg*, 2004. 832–842
- Bader J, Zitzler E. HYPE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 2011, **19**(1): 45–76
- Li K, Deb K, Zhang Q, Kwong S. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 2015, **19**(5): 694–716
- Li M, Yang S, Liu X. Bi-goal evolution for many-objective optimization problems. *Artificial Intelligence*, 2015, **228**: 45–65
- Liu Qiang, Jiao Guo-Shuai. Layout optimization of aero-engine

- pipe clamps based on kriging model and NSGAI. *CAAI Transactions on Intelligent Systems*, 2019, **14**(2): 281–287
(柳强, 焦国帅. 基于Kriging模型和NSGA-II的航空发动机管路卡箍布局优化. 智能系统学报, 2019, **14**(2): 281–287)
- 15 Akhtar T, Shoemaker C A. Multi-objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection. *Journal of Global Optimization*, 2016, **64**(1): 17–32
- 16 Zhang Q, Liu W, Tsang E, Virginas B. Expensive multi-objective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 2010, **14**(3): 456–474
- 17 Chugh T, Jin Y, 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*, 2016, **22**(1): 129–142
- 18 Wang X, Jin Y, Schmitt S, Olhofer M. An adaptive Bayesian approach to surrogate-assisted evolutionary multi-objective optimization. *Information Sciences*, 2020, **519**: 317–331
- 19 Yang C, Ding J, Jin Y, Chai T. Off-line data-driven multi-objective optimization: Knowledge transfer between surrogates and generation of final solutions. *IEEE Transactions on Evolutionary Computation*, 2020, **24**(3): 409–423
- 20 Yang C, Ding J, Jin Y, Wang C, Chai T. Multi-tasking multi-objective evolutionary operational indices optimization of beneficiation processes. *IEEE Transactions on Automation Science and Engineering*, 2019, **16**(3): 1046–1057
- 21 Zhao Y, Sun C, Zeng J, Tan Y, Zhang G. A surrogate-ensemble assisted expensive many-objective optimization. *Knowledge-Based Systems*, 2020, **211**: 106520
- 22 Knowles J. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 2006, **14**(1): 50–66
- 23 Pan L, He C, Tian Y, Wang H, Zhang X, Jin Y. A classification based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2019, **23**(1): 74–88
- 24 Zhang J, Zhou A, Zhang G. A classification and pareto domination based multiobjective evolutionary algorithm; In: Proceedings of the 2015 IEEE Congress on Evolutionary Computation. Sendai, Japan: 2015. 2883–2890
- 25 Freeny A. Empirical model-building and response surfaces. *Technometrics*, 1987, **30**(2): 229–231
- 26 Yang H, Liu J. An adaptive RBF neural network control method for a class of nonlinear systems. *IEEE/CAA Journal of Automatica Sinica*, 2018, **5**(4): 457–462
- 27 Sun C, Jin Y, Cheng R, Ding J, Zeng J. Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Transactions on Evolutionary Computation*, 2017, **21**(4): 644–660
- 28 He Zhi-Kun, Liu Guang-Bin, Zhao Xi-Jing, Wang Ming-Hao. Overview of Gaussian process regression. *Control and Decision*, 2013, **28**(8): 1121–1129
(何志昆, 刘光斌, 赵曦晶, 王明昊. 高斯过程回归方法综述. 控制与决策, 2013, **28**(8): 1121–1129)
- 29 Nelson D, Wang J. Introduction to artificial neural systems. *Neurocomputing*, 1992, **4**(6): 328–330
- 30 Cortes C, Vapnik V N. Support-vector networks. *Machine Learning*, 1995, **20**(3): 273–293
- 31 Liao P, Sun C, Zhang G, Jin Y. Multi-surrogate multi-tasking optimization of expensive problems. *Knowledge-Based Systems*, 2020, **205**(6): 106262
- 32 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. 基于多点加点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)
- 33 Guo D, Jin Y, Ding J, Chai T. A multiple surrogate assisted decomposition based evolutionary algorithm for expensive multi/many-objective optimization. *IEEE Transactions on Cybernetics*, 2018, **49**(3): 1012–1025
- 34 Deb K, Thiele L, Laumanns M, Zitzler E. Scalable multi-objective optimization test problems. In: Proceedings of the 2002 IEEE Congress on Evolutionary Computation. Honolulu, USA: 2002. 1: 825–830
- 35 Habib A, Singh H K, Chugh T, Ray T, Miettinen K. A multiple surrogate assisted decomposition based evolutionary algorithm for expensive multi/many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2019, **23**(6): 1000–1004

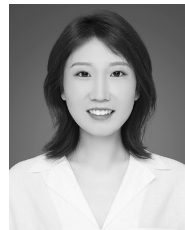


孙超利

太原科技大学教授。主要研究方向为计算智能和机器学习。

E-mail: chaoli.sun@tyust.edu.cn

(SUN Chao-Li Professor at Taiyuan University of Science and Technology. Her research interest covers computational intelligence and machine learning.)



李贞

太原科技大学硕士研究生。主要研究方向为计算智能和机器学习。

E-mail: s20180522@stu.tyust.edu.cn

(LI Zhen Master student at Taiyuan University of Science and Technology. Her research interest covers computational intelligence and machine learning.)



金耀初

英国萨里大学教授。主要研究方向为计算智能,机器学习,计算生物学和计算神经科学等。本文通信作者。

E-mail: yaochu.jin@surrey.ac.uk

(JIN Yao-Chu Professor at University of Surrey, UK. His research

interest covers interdisciplinary areas that bridge the gap between computational intelligence and machine learning, computational neuroscience, and computational biology. Corresponding author of this paper.)