

基于集成代理协同进化算法求解大规模昂贵优化问题

摘要

代理辅助进化算法 (SAEAs) 在解决昂贵优化问题 (EOPs) 方面表现出了良好的性能, 这些问题的真实评估在计算或物理上都很昂贵。然而, 现有的 saea 大多只关注低维问题, 很少考虑解决大规模的 EOPs (LSEOPs)。为了填补这一研究空白, 原论文文提出了一个基于集成代理的协同进化优化器来处理 LSEOPs。首先, 通过对大规模决策变量进行特征选择, 用低维数据子集训练局部代理模型, 其中一部分用于构建选择性集成代理, 以更好地逼近目标 LSEOP。然后, 设计了一个以集成代理为指导的协同进化优化器, 通过运行两个种群来协同解决目标 LSEOP 和简化的辅助问题。通过共享两个种群的子代信息, 促进协同进化过程, 利用简化辅助问题的搜索经验, 帮助求解目标 LSEOP。最后, 采用有效的填充选择准则对集成代理进行更新, 提高其近似性能。本文在原论文基础上尝试了不同的特征选择的标准, 并进行实验与原文结果对比, 实验结果说明我们的改进方法在大规模的低维问题上表现较好。

关键词: 集成代理; 大规模昂贵优化问题 (LSEOP); 代理辅助进化算法 (SAEA)

1 引言

在许多实际应用中, 经常需要优化具有各种复杂性质的单目标优化问题, 如非凸性、多模态和不可微性。传统的数学方法无法有效解决这些复杂的优化问题, 而一些启发式算法, 如进化算法 (EA) 和粒子群优化器 (PSO) 更为有效和高效。通常, 这些启发式算法隐含地假设候选解的评估以获得其客观值和约束是容易和廉价的。然而, 这一假设在一些现实场景中并不总是成立 [19], 例如, 数值模拟或物理实验, 通常在计算或物理上昂贵的评估。这类问题通常被称为昂贵优化问题 (EOPs)。由于 EOPs 的真实评价数量有限, 传统的启发式算法无法快速找到满意的结果。因此, 原论文提出了一种新的 EAs 来解决 EOPs, 通常称为代理辅助 EAs (SAEAs)。SAEAs 使用少量评估的解决方案作为训练样本, 训练一个代理模型, 以取代计算昂贵的评估, 这可以显着降低求解 EOPs 的计算成本。大多数现有的 SAEAs 只专注于处理不超过 300 个维度的 EOPs, 但是在现实应用中也存在一些大型 EOPs (LSEOPs), 最多有 1000 个维度。为了填补这一研究空白, 原论文提出了一种基于集成代理的协同进化优化器 (ESCO) 来解决 LSEOPs 问题。

2 相关工作

2.1 代理辅助进化算法

Kriging 模型是 SAEA 中使用最广泛的代理模型，它被证明在处理 EOPs 方面具有良好的性能。文献 [12] 提出了一种高效的全局优化器 (EGO)，首先应用 Kriging 模型对 4 个决策变量分别为 2、2、3 和 6 的测试 EOPs 进行优化。在 [13] 中，提出了一种针对 20-50 个决策变量的中等规模 EOPs 的高斯过程辅助 EA (GPME)，其中使用降维技术将训练数据映射到较低维空间，并提出了一种新的代理模型感知搜索机制，专注于搜索有希望的子区域。在 [18] 中，提出了一种多目标填充准则驱动的高斯过程辅助粒子群算法 (MGPSLPSO)，用于解决具有多达 100 个决策变量的 EOPs 和具有 31 个决策变量的合成现实问题。在该方法中，设计了一个多目标填充准则来优化代理模型中的两个独立目标，即近似适应度的最小化和不确定性的最大化。然后，采用非支配排序法 [3] 选择真函数评价的解。文献 [2] 设计了求解 LSEOPs 的 SACC 算法，利用 Kriging 模型作为代理，对原问题分解的低维子问题进行近似，然后利用自适应差分进化 (adaptive differential evolution, DE) 算法 [23] 对每个子问题进行优化。

RBF 是 SAEAs 中另一个常用的代理模型。文献 [17] 提出了一种基于 RBF 的代理辅助协同群优化器 (SACOSO)，用于求解决策变量多达 100 个的 EOPs。在该方法中，设计了两种类型的代理辅助 PSO 来协同搜索目标 EOPs 的全局最优。在 [21] 中，提出了一个有效的 ESAO 来解决 EOPs 与多达 100 个维度和翼型设计问题与 12 个维度。在 ESAO 中，首先采用全局 RBF 模型选择由突变和交叉生成的最优解，然后采用自然优化器搜索具有一定数量训练的局部代理模型的最优解目前最好的样品之一。在 [20] 中，设计了一个具有选择性代理集成的离线 SAEA (DDEA-SE) 来解决多达 100 维的离线 EOPs，其中使用 bagging 技术离线训练大量代理 [1]，[6] 并自适应选择其中的一个子集用于估计进化过程中的适应度。文献 [7] 利用随机特征设计了一种新的 SAEA 选择方法，称为 SAEA-RFS。该算法采用随机特征选择方法获得若干子问题，然后利用 RBF 模型辅助的进化优化器对这些子问题进行顺序优化。在 [11] 中，一种基于随机排序的数据驱动 SRK-DDEA 是针对离线 LSEOPs 而设计的，它提出了一个随机排序来管理四个不同核的 RBF 模型。在 [10] 中，提出了一种新的使用三训练 (TT-DDEA) 的离线数据驱动 EA。该算法采用半监督学习方法选择具有高置信度适应度预测的解，丰富了代理模型的训练数据。

2.2 集成代理

为了利用不同的代理模型，一些 SAEAs 使用多个代理或集成代理。文献 [22] 提出了一种基于代理的 PSO (SBPSO) 来解决两个 EOPs 和一个二维的实际案例研究。该算法结合传统优化方法和粒子群算法的优点，采用混合代理模型逼近昂贵的黑箱。在 [14] 中，一种名为 SAEA-UGC 的新型 SAEA 被报道用于处理 20 个具有 30 个决策变量的广泛使用的测试 EOPs。原论文提出了一种新的基于不确定性分组的填充准则，用于全局搜索，而局部搜索采用仅由一定数量的当前最优解训练的 RBF 模型

3 本文方法

3.1 ESCO 方法概述

原文提出了一个基于集成代理的协同进化优化器来处理 LSEOPs 即 ESCO 方法。首先, 通过对大规模决策变量进行特征选择, 用低维数据子集训练局部代理模型, 其中一部分用于构建选择性集成代理, 以更好地逼近目标 LSEOP。然后, 设计了一个以集成代理为指导的协同进化优化器, 通过运行两个种群来协同解决目标 LSEOP 和简化的辅助问题。通过共享两个种群的子代信息, 促进协同进化过程, 利用简化辅助问题的搜索经验, 帮助求解目标 LSEOP。最后, 采用有效的填充选择准则对集成代理进行更新, 提高其近似性能, 下面将详细介绍各个步骤。

3.2 选择性集合代理

选择性集成代理的主要思想是独立生成多个局部代理, 然后从中选择一部分来构建一个选择性集成, 以提高预测 LSEOPs 解的适应度值的代理模型的质量, 如图1所示。首先, 该方法通过在原始数据集上使用特征选择和随机抽样来创建不同的训练数据子集, 用于训练不同的局部代理。由于高度非线性模型的线性组合可以显著降低生成误差的方差, 因此采用属于非线性模型的 RBF 网络 (RBFN) 作为替代。

为了对 LSEOPs 进行特征选择, 使用肯德尔秩相关系数 (Kendall rank correlation coefficient, KRCC) 来衡量每个决策变量与目标之间的相关性。KRCC 计算方法介绍如下, 假设选取第 i 个决策变量 x_i 为例, $\{(x_i^1, y^1), \dots, (x_i^n, y^n)\}$ 表示训练数据集 TD 中 x_i 的 n 个样本和相应的目标值。然后, 对任意一对两个样本 (x_i^j, y^j) 和 (x_i^k, y^k) 进行相互比较。当 $x_i^j < x_i^k \& y^j < y^k$ or $x_i^j > x_i^k \& y^j > y^k$ 时, 这对样品是一致的, 而当 $x_i^j < x_i^k \& y^j > y^k$ or $x_i^j > x_i^k \& y^j < y^k$ 时, 这对样品是不一致的。因此, 第 i 个决策变量 x_i 与目标 y 之间的 KRCC (τ_i) 可计算为:

$$\tau_i = \frac{n_c - n_d}{n(n-1)/2} \quad (1)$$

其中 n_c 为一致对数, n_d 为不一致对数。 τ_i 的值位于 $[-1, 1]$ 。在这里, τ_i 的值接近于 -1 或 1 表明 x_i 与目标 y 的相关性较强。特别是当 $x_i=0$ 时, x_i 与目标 y 无关, 因此在训练 RBFN 模型时可以忽略 x_i 。计算出各决策变量与目标之间的 KRCC 的相关值后, 利用 $KRCC(\tau_i)$ 可以计算出第 i 个决策变量 x_i 对目标 y 的贡献率, 如下所示:

$$r_i = \frac{|\tau_i|}{\sum_{j=1}^d |\tau_j|} \quad (2)$$

d 是 x_i 的维数。因此, t 个训练集 ($S_1^{train}, S_2^{train}, \dots, S_t^{train}$) 和 t 个测试集 ($S_1^{test}, S_2^{test}, \dots, S_t^{test}$) 通过根据 (2) 的贡献率进行轮盘赌选择 and 随机抽样产生。 t 个训练集用于训练 t 个局部 RBF 模型 (m_1, m_2, \dots, m_t), 而 t 个测试集用于选择其中的一些数据集来构建最终的选择性集成。

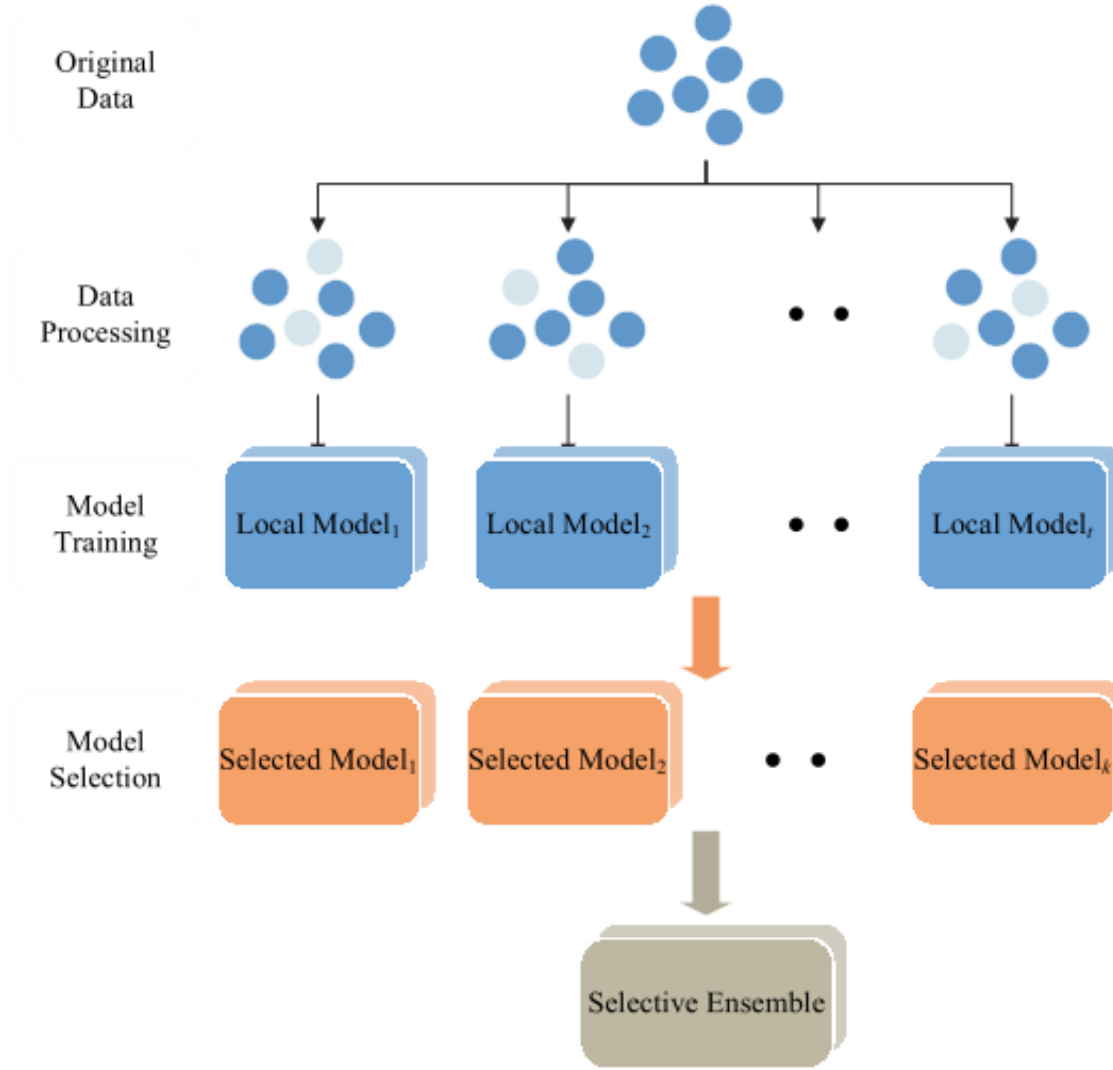


图 1. 构造选择性集成图

如 [20] 和 [8] 的实验研究，集成学习方法或代理模型的集成在提高 SAEAs 的性能方面显示出有效性。因此，本文还采用了集成方法，将不同的替代指标进行组合，以提高预测精度。本文采用 Kullback-Leibler 散度 (KLD) 来选择局部模型，它能够评估两个概率分布之间的相似性，从而降低预测的不确定性。在数理统计中，KLD 可以反映一个概率分布与另一个概率分布的差异程度。考虑两个正态分布 A 和 B，如下：

$$A \sim \mathcal{N}(\mu_1, \sigma_1^2), B \sim \mathcal{N}(\mu_2, \sigma_2^2) \quad (3)$$

从 B 到 A 的 KLD 常记为 $\text{KLD}(A|B)$ ，可由下式计算：

$$\text{KLD}(A|B) = \log \frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2} \quad (4)$$

其中， μ_1 和 μ_2 分别为平均值， σ_1 和 σ_2 分别为 A 和 B 两个正态分布离 $\mathcal{N}(\bullet, \bullet)$ 的标准差。

对于每个局部模型，可以得到 S_i^{test} 的预测适应度值 Y_p 。接下来，根据式 (4) 可以计算出 Y_p 到真实适应度值 Y_r 的 KLD 值 [即 $\text{KLD}(Y_r|Y_p)$]。值得注意的是，KLD 是不对称的，

$KLD(Y_r|Y_p)$ 不等于 $KLD(Y_p|Y_r)$ 。在获得所有 t 个局部 RBFN 模型的 KLD 值后，选择局部模型的一个子集来基于它们的 KLD 值构建集成。

该方法采用 top-k 最佳加权选择策略 (TOP-K WSS)，只选择 KLD 值最小的前 k 个局部模型形成选择性集成。具体而言，Top-K WSS 构建的选择性集合如下：

$$se = \{m_i | i \in \mathbf{I}\} \quad (5)$$

其中 se 为 Top-K WSS 产生的选择性集合， \mathbf{I} 为 k 个最佳模型的指标集。同时，继承代理的预测值计算方式如下：

$$y_{se} = \sum_{i \in \mathbf{I}} w_i y_i \quad (6)$$

$$w_i = \frac{KLD_i}{\sum_{j=1}^t KLD_j} \quad (7)$$

其中 y_i 为第 i 个局部模型 m_i 的输出值， w_i 为 m_i 的权值， KLD_i 为 m_i 的 KLD 值。

为了阐明这个过程的细节，2是构造选择性集成代理的伪代码。算法 2 的输入包括存储在数据库 TD 中的一组样本数据，而输出是选择性集成代理 se 和局部模型集 LS。首先初始化集成的局部模型总数 t 和选择模型数 k ，并在第 1 行中将局部模型集 LS 设置为空集。在第 2 行中，根据 (1) 和 (2) 计算每个决策变量对目标的贡献率 r_i 。然后在第 4-12 行中，使用不同的训练样本依次构建 t 个局部模型。具体来说，变量索引 ind_i 是通过轮盘选择从原始变量向量中获得的，以便在第 4 行中构建第 i 个本地代理。为了生成第 i 个本地代理的训练数据子集，在第 5-11 行中使用了与概率相关的抽样方法。TD 中的每个样本 s_i 被选为第 i 个局部模型的训练样本的概率为 0.75，加入到训练集 S_i^{train} 中。采用概率相关抽样方法选择 t 个训练数据集后，对数据 $(S_1^{test}, S_2^{test}, \dots, S_t^{test})$ 也是通过保存未用作训练样本的剩余解生成的。在第 12 行和第 13 行中，使用 S_i^{train} 构造了一个局部模型 m_i ，并将带有 ind_i 的 m_i 添加到 LS 中。 m_i 的 KLD 值由第 14 行和第 15 行中的 (3) 和 (4) 计算。最后，在第 17-19 行中，Top-K WSS 被用于构建最终的选择性集合 se 。值得注意的是，每次在演化过程中更新 TD 时都会触发模型选择过程。

在这种方法中，轮盘选择所选择的变量数量是随机的，因此每个训练子集的变量维数是不同的。由于我们的方法使用了概率依赖抽样，因此每个训练集的大小不是固定的，可以获得具有更大多样性的局部代理。此外，在一些 SAEAs [15]、[9] 中，通常将训练集的大小设置为大于测试集的大小。因此，选择样本作为训练样本的概率设为 [15] 建议的 0.75，也可以采用其他值，如 [9] 建议的 0.8。

Algorithm 2 Selective-Ensemble-Surrogate (*TD*)

Input: *TD* (training dataset)

```
1: Initialize  $t, k$ , and set local models set  $LS = \emptyset$ 
2: Calculate the contribution ratio  $r_j$  ( $j = 1, 2, \dots, d$ ) of all
   decision variables to the objective by (1)-(2)
3: for  $i = 1$  to  $t$  do
4:   Obtain  $ind_i$  from the variable vector by roulette wheel
     selection
5:   for each sample  $s$  in  $TD$  do
6:     if  $U(0, 1) < 3/4$  then
7:       Add  $s(:, ind_i)$  to  $S_i^{train}$ 
8:     Else
9:       Add  $s(:, ind_i)$  to  $S_i^{test}$ 
10:    end if
11:  end for
12:  Construct the local model  $m_i$  based on  $S_i^{train}$ 
13:   $LS = LS \cup \{m_i, ind_i\}$ 
14:  Predict the objective values  $Y_p$  for all samples in  $S_i^{test}$ 
15:  Calculate the KLD value  $KLD(Y_r | Y_p)$  from  $Y_p$  to their
     true objective values  $Y_r$  by (4)
16: end for
17: Sort models in  $LS$  in ascending order based on their KLD
   values
18: Preserve only the first  $k$  models in  $LS$ 
19: Construct the selective ensemble  $se$  using  $LS$  by (5)
20: return  $LS, se$ 
```

图 2. 算法 2

3.3 代理辅助协同进化优化器

在构建了选择性集成之后, 进一步使用 SACO 来搜索更有希望的解, 其中使用上节构建的选择性集成代理 se 来近似解的适应度值, 而不是真函数评估。为了说明中美合作所的运行情况, 给出了其伪代码3。从 TD 中选择 n 个最优解作为初始总体 P , 同时通过拉丁超立方体抽样初始化辅助总体 P_a [16]。值得注意的是, P 的维数与原始 LSEOP 相同, 而 P_a 的维数与训练结果一致, P 的维数与 se 中第 c 个局部模型的维数一致 (c 为 $[1, k]$ 中的随机整数, k 为所选模型的个数)。在每一代中, 通过交配选择策略分别从 P 和 P_a 中选择两个大小为 $n/2$ 的亲本。然后, 通过依次运行模拟二元交叉 [4] 和多项式突变 [5], 应用两个亲本群体产生大小相同的对应子代 (O 和 O_a)。之后, 对于 P , 依次用 O_a 的决策变量替换 O 中选择的独立决策变量, 并保留 O 中剩余的决策变量, 产生另一个子代群体 Q , 同时用 O 中对应的独立决策变量形成 P_a 的第二个子代集合 Q_a 。然后, 将 P 和 P_a 与其对应的两个子代群体组合为联合群体 (U 和 U_a)。下一步, U 中的解用 se 求值, 而 U_a 中的解总是用 LS 中的第 c 个局部模型 m_c 求值。一旦我们的替代物预测了所有解的适应度值, 将分别从 U 和 U_a 中选择 n 个最佳解, 并将其保存到下一代。最后, 返回 P 作为中美合作所的最终输出。

从算法 3 可以清楚地看出, 两个种群 P 和 P_a 同时进化, 分别解决原始问题 f 和简化后的辅助问题 f_a 。由于 f_a 只包含 f 中的部分决策变量, 因此 f_a 的优化比 f 的优化更容易。因此, 如图 2 所示, 这种共同进化过程可以通过两个共同进化群体所产生的后代共享来快速找到更有希望的解决方案。换句话说, P 的收敛速度会被 P_a 加快, 其中 P_a 中的解通常比 P 中的解收敛得更好。

Algorithm 3 SACO (TD, LS, se)

Input: TD (training dataset), LS (local surrogates), se (selective ensemble)

- 1: Initialize n, g_{\max} and set generation counter $g = 0$
- 2: $P \leftarrow$ Select the best n samples in TD
- 3: $[m_c, ind_c] \leftarrow$ Select a local model from LS randomly
- 4: $P_a \leftarrow$ Initialize n solutions with the same dimension as ind_c
- 5: **while** $g \leq g_{\max}$ **do**
- 6: Generate offspring population O from P and O_a from P_a
- 7: Generate another offspring population Q, Q_a by swapping ind_c decision variables of O and O_a
- 8: $U \leftarrow P \cup O \cup Q; U_a \leftarrow P_a \cup O_a \cup Q_a$
- 9: Evaluate U by se and U_a by the local model m_c
- 10: Update P and P_a respectively using U and U_a
- 11: $g = g + 1$
- 12: **end while**
- 13: **return** P

图 3. 算法 3

3.4 填充方案选择

由于一个真正的函数评估必须消耗 LSEOPs 的大量计算时间，因此应该仔细选择要真正评估的解决方案。这些选择的解决方案被命名为填充解决方案，它们被保存到 TD 中并用于重新训练代理模型。如果选择太多不重要和冗余的解决方案作为填充解决方案，那么将浪费非常有限的真实评估次数，从而导致对降低计算成本没有意义。

本文提出了一种简单有效的填充选择准则，并给出了该准则的伪代码⁵。选择准则是基于预测的适应度值和候选解与样本之间的距离。在获得的最终子代群体 P 中进行填充溶液的识别过程。首先，去除一些不重要的解，将剩余的解选择到候选池 c 中。具体而言，对于 P 中的每个解 s ，如果我们的集成代理预测的近似目标值小于 TD 中最佳样本 b 的真实目标值，并且决策空间中 s 到所有样本的最小欧氏距离大于预定义阈值 l ，对 P 的每个解执行此过程后， C 中的所有解按照其大小从小到大排序近似适应度值，然后选择最佳近似适应度值的解对代理进行重新训练。在候选池 C 中不存在解的特殊情况下，将从 p 中随机选择一个解。注意，这种特殊情况概率非常小，不会对我们的性能产生显著影响。

Algorithm 4 Infill-Solutions-Selection (TD, P)

Input: TD (training dataset), P (population obtained by Algorithm 3)

```

1:  $b \leftarrow$  Obtain the best sample in  $TD$ 
2: Initialize  $l$  and set  $C = \emptyset$ 
3: for each solution  $s$  in  $P$  do
4:   if  $s < b$  then
5:      $d \leftarrow$  Calculate the minimum distance in the decision
       space between  $s$  and all the samples of  $TD$ 
6:     if  $d > l$  then
7:        $C \leftarrow C \cup s$ 
8:     end if
9:   end if
10: end for
11: if  $C \neq \emptyset$  then
12:   Sort all solutions in  $C$  based on their predicted fitness
     values
13:    $s \leftarrow$  Select a solution with best predicted value in  $C$ 
14: Else
15:    $s \leftarrow$  Select a solution from  $P$  randomly
16: end if
17: return  $s$ 

```

图 4. 算法 4

4 复现细节

4.1 与已有开源代码对比

本文是在原论文已有开源代码的基础上进行改进的, 主要是参考了源代码的整体架构, 从而对算法细节进行了一些改进与创新, 在作者提供的这个代码的基础上进行了一些改进, 所做的工作主要是对集成代理模型的标准由 KL 散度替换成了交叉熵, 并分别对 F1 问题的 100 维和 1000 维进行了实验, 结果表明改进之后的方法在大规模问题的中低维问题上效果表现较好。

4.2 实验环境搭建

本次复现工作实验环境如下: Windows11、MatlabR2020b。

4.3 创新点

本次的学习的任务都是监督学习任务, 每个解都有对应的真实评价值, 故考虑将原文的 KL 散度替换成交叉熵来评价局部代理模型的优劣, 作为选择集成代理模型的指标, 结果如后文所示。

5 实验结果分析

5.1 复现结果分析

本次实验使用了 5 个基准 EOP (F1-F5), 即 1 个单模态 EOP 和 4 个多模态 EOP, 复现计算最优值 (全局最小值) 如图5所示, 每个基准问题进行 5 次实验取平均。在我们的实验研究中, 对于 $d = 100$ 或 300 的问题, 初始样本数量设置为 200, 对于 $d = 500$ 或 1000 的问题, 初始样本数量设置为 400 (d 是决策空间的维度), 真实函数求值的总数都设置为 1000。在我们的算法中, 局部模型的个数 t 设置为 10, Top-K WSS 策略中选择的局部模型个数 k 设为 3, 种群大小 n 和迭代最大值 $gmax$ 分别设置为 50 和 20。预定义的阈值 l 被设置为 $\sqrt{10^6 \times d}$,

problem	D	ESCO	复现结果
F1(Ellipsoid)	100	2.37E+02	2.39E+02
	300	2.3E+03	2.89E+03
	500	2.3E+04	2.56E+04
	1000	1.77E+05	2.99E+05
F2(Rosenbrock)	100	2.24E+02	2.55E+02
	300	1.05E+03	8.96E+02
	500	3.33E+03	4.40E+03
	1000	1.02E+04	2.29E+04
F3(Rastrigin)	100	4.06E+02	4.08E+02
	300	2.1E+03	2.06E+03
	500	4.03E+03	3.70E+03
	1000	8.32E+03	8.63E+03
F4(Ackley)	100	6.11E+00	6.18E+00
	300	8.52E+00	7.16E+00
	500	6.77E+00	5.73E+00
	1000	6.39E+00	6.29E+00
F5(Griewank)	100	1.16E+01	1.15E+01
	300	4.31E+01	4.67E+01
	500	3.36E+02	2.91E+02
	1000	1.42E+03	2.15E+03

图 5. 复现结果与原文对比

为了直观地展示 ESCO 的收敛性能, 绘制了 ESCO 在 5 个基准问题上 100-D(100 维)、300-D、500-D、1000-D 上的收敛曲线如图6、7、8、9所示, 其中 x 轴为真实适应度评价, y 轴为 5 次独立运行得到的函数均值。

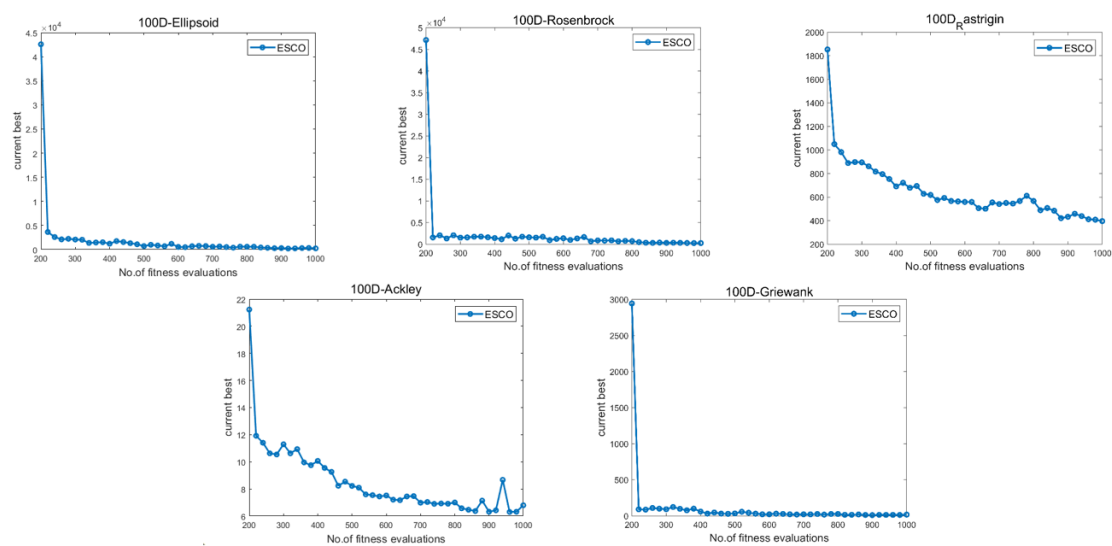


图 6. ESCO 在 5 个基准问题上 100 维的收敛图复现结果

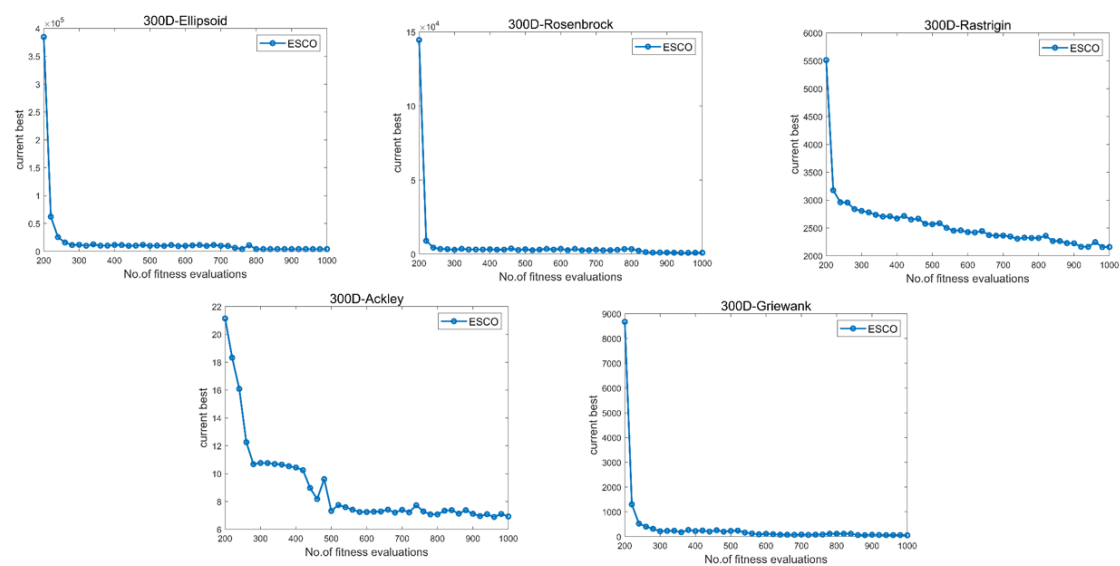


图 7. ESCO 在 5 个基准问题上 300 维的收敛图复现结果

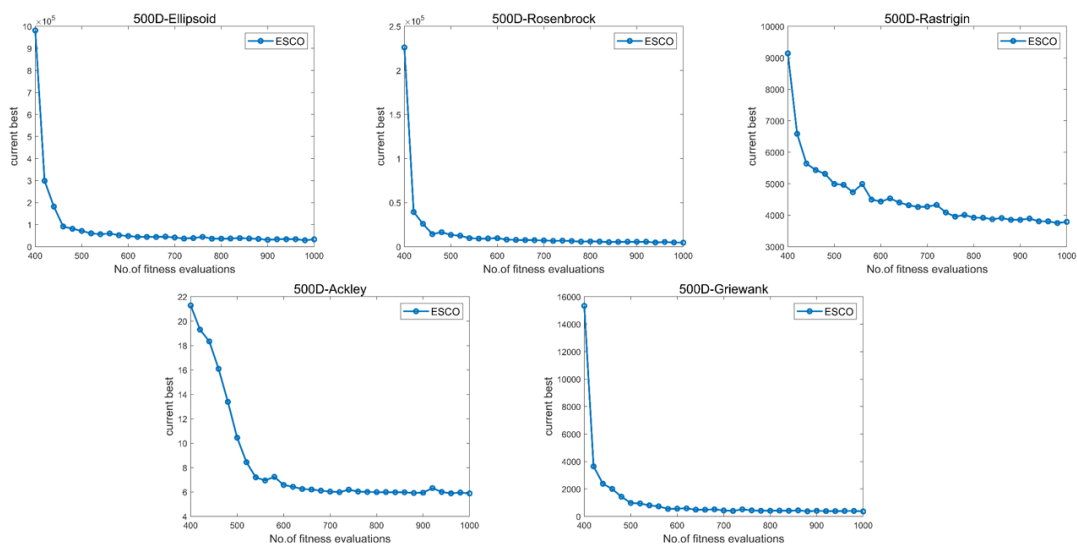


图 8. ESCO 在 5 个基准问题上 500 维的收敛图复现结果

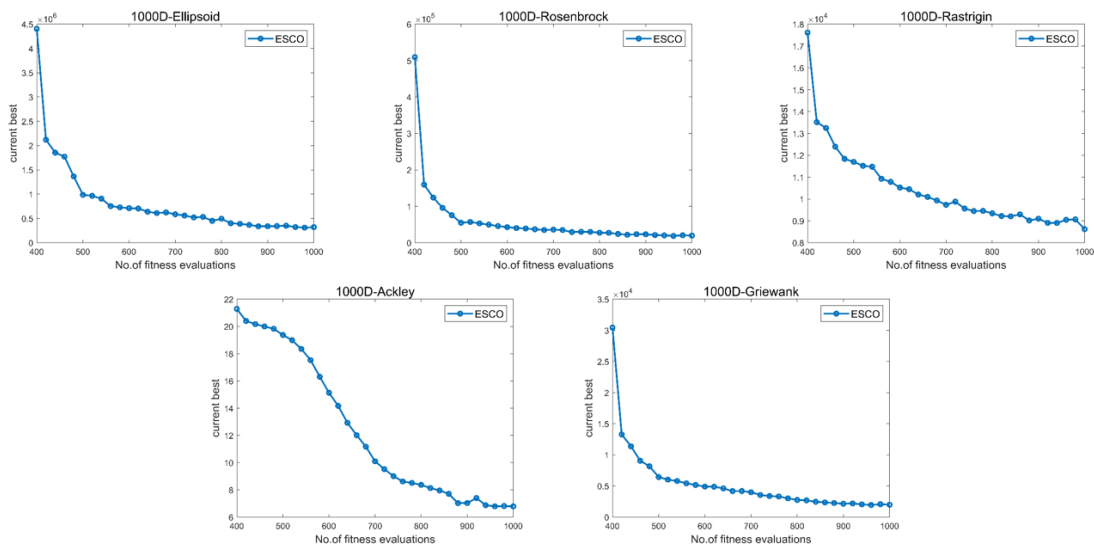


图 9. ESCO 在 5 个基准问题上 1000 维的收敛图复现结果

通过本次实验，可以看到，随着维数变大，算法的收敛速度明显会变慢。

5.2 优化结果分析

KL 散度与交叉熵都是衡量两个概率分布差异性的指标。交叉熵通常用于监督学习任务中，通常这种任务都有一组输入和相应的标签；而 KL 散度通常用于无监督学习中，这些任务没有对应的标签信息。而本次的学习的任务都是监督学习任务，每个解都有对应的真实评价值，故考虑将 KL 散度替换成交叉熵来评价局部代理模型的优劣，作为选择集成代理模型的指标，使用原论文方法和改进后的方法在 F1 问题的 100 维和 1000 维上分别进行 5 次实验，对比最后的最优值如表10、11所示。

100D结果						
	Times					MEAN
	1	2	3	4	5	
ESCO-cross_entropy	178.2839	166.3528	188.8795	159.7659	181.6748	174.9914
ESCO-KLD	177.2745	224.3615	317.2859	237.9427	237.5261	238.8781

图 10. ESCO-cross_entropy、ESCO-KLD 全局最优结果对比（100 维）

1000D结果						
	Times					MEAN
	1	2	3	4	5	
ESCO-cross_entropy	278783.3	306219.1	349300.3	298145.1	327642.8	312018.1
ESCO-KLD	298237.4	269495.3	348725.4	314392.2	266064	299382.8

图 11. ESCO-cross_entropy、ESCO-KLD 全局最优结果对比（1000 维）

通过结果我们可以看到，在 100 维的情况下，利用交叉熵作为选择继承代理的标准优化的结果更好，但是在 1000 维度下交叉熵改进的效果不明显。

6 总结与展望

本次复现的论文提出了一个基于集成代理的协同进化优化器，称为 ESCO，用于解决具有有限数量的函数评估的 LSEOPs。为了构建足够高质量的有效代理模型，ESCO 采用选择性集成代理，实现对 LSEOPs 目标值更准确有效的逼近。为了构建这样一个选择性集成代理，首先在通过特征选择和轮盘赌选择选择的数据子集上训练几个不同的局部代理模型。然后，ESCO 选取部分构建好的局部模型组成最终的选择性集成模型，进一步提高预测精度和效率。在这一部分中，原文提出了一种简单有效的策略 (Top-K WSS)，它选择 KLD 值最小的 Top-K 局部模型，并根据 KLD 值赋予它们不同的权重，组成一个集成代理。在获得选择性集成代理后，设计协同进化优化器，借助简化的辅助问题对原 LSEOP 进行优化，加快了 EA 的收敛速度。最后，提出了一种简单有效的填充选择准则，以识别具有良好收敛性和多样性的有前途的填充样本，有助于进一步提高代理模型的预测精度。

本文在原文基础上改进了集成代理模型的选择标准，利用交叉熵代替 KL 散度，效果在大规模的中低维问题上表现较原来相比有提升。未来还能从子问题的生成、代理模型的训练数据选取等方面进行改进。

参考文献

- [1] L. Breiman. Bagging predictors. *Mach. Learn*, 24(2):123–140, 1996.
- [2] Ivano De Falco, Antonio Della Cioppa, and Giuseppe A. Trunfio. Investigating surrogate-assisted cooperative coevolution for large-scale global optimization. *Information Sciences*, 482:1–26, 2019.

- [3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [4] Kalyanmoy Deb, Ram Bhushan Agrawal, et al. Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995.
- [5] Kalyanmoy Deb, Mayank Goyal, et al. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and informatics*, 26:30–45, 1996.
- [6] Jerome H. Friedman and Peter Hall. On bagging and nonlinear estimation. *Journal of Statistical Planning and Inference*, 137(3):669–683, 2007.
- [7] Guoxia Fu, Chaoli Sun, Ying Tan, Guochen Zhang, and Yaochu Jin. A surrogate-assisted evolutionary algorithm with random feature selection for large-scale expensive problems. In Thomas Bäck, Mike Preuss, André Deutz, Hao Wang, Carola Doerr, Michael Emmerich, and Heike Trautmann, editors, *Parallel Problem Solving from Nature – PPSN XVI*, pages 125–139, Cham, 2020. Springer International Publishing.
- [8] Dan Guo, Yaochu Jin, Jinliang Ding, and Tianyou Chai. Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems. *IEEE Transactions on Cybernetics*, 49(3):1012–1025, 2019.
- [9] Ahsanul Habib, Hemant Kumar Singh, Tinkle Chugh, Tapabrata Ray, and Kaisa Miettinen. A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 23(6):1000–1014, 2019.
- [10] Pengfei Huang, Handing Wang, and Yaochu Jin. Offline data-driven evolutionary optimization based on tri-training. *Swarm and Evolutionary Computation*, 60:100800, 2021.
- [11] Pengfei Huang, Handing Wang, and Wenping Ma. Stochastic ranking for offline data-driven evolutionary optimization using radial basis function networks with multiple kernels. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2050–2057, 2019.
- [12] Schonlau M. Welch W.J. Jones, D.R. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, page 455–492, 1998.
- [13] Bo Liu, Qingfu Zhang, and Georges G. E. Gielen. A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Transactions on Evolutionary Computation*, 18(2):180–192, 2014.
- [14] Qunfeng Liu, Xunfeng Wu, Qiuzhen Lin, Junkai Ji, and Ka-Chun Wong. A novel surrogate-assisted evolutionary algorithm with an uncertainty grouping based infill criterion. *Swarm and Evolutionary Computation*, 60:100787, 2021.

- [15] Linqiang Pan, Cheng He, Ye Tian, Handing Wang, Xingyi Zhang, and Yaochu Jin. A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 23(1):74–88, 2019.
- [16] Michael Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2):143–151, 1987.
- [17] Chaoli Sun, Yaochu Jin, Ran Cheng, Jinliang Ding, and Jianchao Zeng. Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Transactions on Evolutionary Computation*, 21(4):644–660, 2017.
- [18] Jie Tian, Ying Tan, Jianchao Zeng, Chaoli Sun, and Yaochu Jin. Multiobjective infill criterion driven gaussian process-assisted particle swarm optimization of high-dimensional expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(3):459–472, 2019.
- [19] Handing Wang, Yaochu Jin, and Jan O. Jansen. Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system. *IEEE Transactions on Evolutionary Computation*, 20(6):939–952, 2016.
- [20] Handing Wang, Yaochu Jin, Chaoli Sun, and John Doherty. Offline data-driven evolutionary optimization using selective surrogate ensembles. *IEEE Transactions on Evolutionary Computation*, 23(2):203–216, 2019.
- [21] Xinjing Wang, G. Gary Wang, Baowei Song, Peng Wang, and Yang Wang. A novel evolutionary sampling assisted optimization method for high-dimensional expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(5):815–827, 2019.
- [22] Jianqiao Chen Yuanfu Tang and Junhong Wei. A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions. *Engineering Optimization*, 45(5):557–576, 2013.
- [23] Jingqiao Zhang and Arthur C. Sanderson. Jade: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13(5):945–958, 2009.