



**摘要** 本章介绍了数据驱动优化背后的定义和动机；介绍了两种基本的数据驱动优化范式：离线和在线数据驱动优化；介绍了用于代理辅助进化优化的各种基于种群和基于个体的启发式代理模型管理策略；还介绍了已有的数学模型管理策略，例如，信任域方法和获取函数，其也称为填充标准；提出了一种代理辅助进化搜索鲁棒最优解的方法；最后，给出了用于评估代理模型引导进化优化质量的性能指标。

## 5.1 引言

大多数进化算法假设解析的目标函数可用于质量评估，并且质量评估的计算是很廉价的，因此可以进行数以万计的评估来寻找问题的最优解。这些假设可能不适用于许多现实世界的优化问题。此类问题的第一类属于一大类工程优化问题，其质量评估依赖于耗时的数值分析，例如，计算流体动力学模拟和有限元分析。其中包括空气动力学设计优化问题，例如，涡轮发动机设计、高升力机翼设计、飞机机身设计和赛车设计。其他示例包括复杂工业流程的优化、药物设计和材料设计。这些数值模拟和数值分析在计算上非常密集，每次评估需要几分钟到几小时，甚至几天。第二类问题涉及人类的判断和评估，其难以用数学表达式来建模，例如，服装设计、艺术和音乐设计以及其他工具或产品的美学设计。最后，还有一些现实世界的优化问题，只能根据日常生活或生产过程中收集的数据进行优化。

因此，基于从历史记录、数值模拟或物理实验中收集的数据进行优化的问题称为数据驱动的优化问题<sup>[4,170]</sup>。数据驱动进化优化的跨学科研究领域涉及数据科学、机器学习和进化算法等技术，如图 5.1 所示。在进化的数据驱动优化框架中，首先会收集数据，因为数据可能会受到噪声或错误的影响，所以需要预处理以提高数据质量。然后从数据中构建代理模型，即机器学习模型，来逼近真实的目标函数和/或约束函数。给定近似的目标或约束函数后可以应用进化算法来进行优化搜索。

根据代理模型的类型<sup>[4]</sup>，数据可分为直接数据和间接数据：

- **直接数据**是具有决策变量、相应目标和/或约束值的样本，它们来自模拟或实验。近

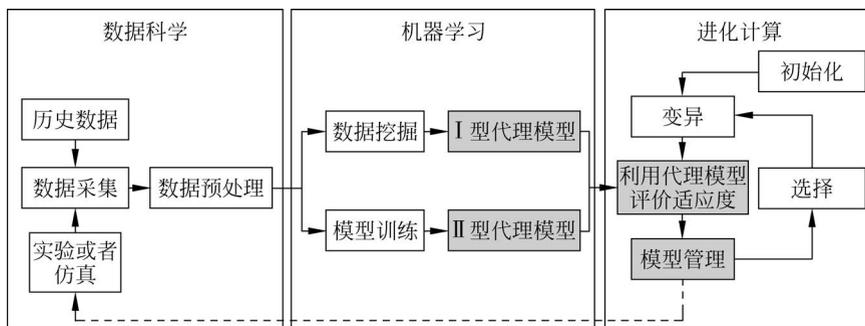


图 5.1 数据驱动的进化优化框架图

似目标或约束函数的代理模型(称为 I 型代理模型)可以直接从此类数据构建。

- **间接数据**不是带有决策变量、对应目标和/或约束值的样本,但计算目标和约束函数需要这些数据。间接数据可用于进一步训练代理模型(称为 II 型代理模型)。

I 型和 II 型代理模型都可以是通用模型或特定模型。现有的机器学习模型可以用作通用模型,因为不需要对问题有太多的先验知识。如果有可用的有用信息,那么问题域中的特定模型(例如具有保真度可控的仿真<sup>[170]</sup>)可能比通用模型更准确。

与传统的进化算法不同,数据驱动的进化优化过程采用代理模型进行函数评价和模型管理策略,以有效控制模型和数据<sup>[171]</sup>。为了控制模型,代理模型的模型类型、超参数或集成结构可以代代更改。为了控制数据,进化算法可以主动采样新数据。但是,在某些情况下,新数据无法由进化算法控制,或者根本没有新数据。因此,根据优化过程中是否可以收集到新数据,数据驱动的进化算法可以分为离线算法和在线算法<sup>[170]</sup>。

需要注意的是,数据驱动的进化优化,特别是在线数据驱动的进化优化,也被称为代理辅助进化优化<sup>[171-172]</sup>,或具有近似适应度函数的进化优化<sup>[171]</sup>。一些离线数据驱动的优化,例如,在文献<sup>[170]</sup>中,不涉及显式代理模型。

## 5.2 离线与在线数据驱动的优化

### 5.2.1 离线数据驱动的优化

在离线数据驱动的进化优化过程中,无法主动采样或持续生成新数据,这使得优化与仅基于离线数据的代理模型高度相关。换句话说,质量和数量都极大地影响了离线数据驱动进化算法的代理模型和优化性能,因为代理模型的准确性无法得到验证<sup>[173]</sup>。

(1) **不理想的质量**: 一般来说,现实世界中的数据可能存在不理想的质量,例如,带噪声、不平衡,甚至不完整的数据。从那些不理想的数据中建立高精度的代理模型是非常困难的。因此,使用这种糟糕的代理模型很容易误导优化方向。

(2) **极端数量**: 无论是大数据还是小数据,都对离线数据驱动的进化算法提出了挑战。在大数据的情况下,代理模型构建的计算成本很高。在小数据的情况下,由于数据缺乏,代理模型的准确性无法保证。

为了应对这些挑战,如何有效地利用有限的数​​据来构建强大而准确的代理模型是离线数据驱动的进化算法的关键问题。以下技术可能会有所帮助,其中一些技术已被用于离线数据驱动的进化优化。

(1) **数据预处理**: 当离线数据不理想时,数据预处理是提高数据质量和代理模型的一种非常直接的方法。例如,文献[174]中高炉问题的数据含噪非常厉害。在构建 Kriging 模型之前,使用局部回归平滑方法<sup>[175]</sup>来进行去噪。

(2) **数据挖掘**: 为了应对大数据驱动进化算法中的高计算成本,应该使用现有的数据挖掘技术提取有效的数据来减少数据冗余<sup>[176]</sup>。例如,聚类算法已被应用于在急救系统设计问题中生成多精度代理模型<sup>[170]</sup>。

(3) **合成数据生成**: 在一些极端情况下,特别是对于高维问题,数据量太小,无法建立代理模型。可以生成合成数据来丰富原始数据,例如,熔镁炉优化问题<sup>[177]</sup>,在算法中,采用低阶多项式模型来生成合成数据。

(4) **半监督学习**: 处理数据不足的另一方法是半监督学习<sup>[178]</sup>,其在模型训练过程中使用未标记数据中的隐含信息。因此,可以减少所需的离线数据量<sup>[179-180]</sup>。

(5) **模型选择**: 虽然有各种类型的代理模型,但它们的优点是不同的。由于离线数据驱动的进化算法没有机会验证或更新其代理模型,因此代理模型的选择(即模型选择)对算法很重要。例如,在文献[181]中,采用验证误差作为选择一种代理模型的判据。

(6) **集成学习**: 为了提高代理模型的鲁棒性,根据集成学习理论<sup>[135]</sup>,可以将大量同构/异构模型组合成一个整体。选择性集成学习算法<sup>[173]</sup>已被用于离线数据驱动的单目标优化。

(7) **多形式优化**: 除了数据和建模技术,多问题建模方式,如多精度适应度评估<sup>[182]</sup>,可以进一步辅助搜索<sup>[183]</sup>。

(8) **迁移学习**: 此外,解决相似问题的过程共享相似的搜索路径,其中可以重用公共知识以节省计算成本<sup>[183]</sup>。迁移优化和多任务优化<sup>[184-186]</sup>可能有助于离线数据驱动的进化算法。

## 5.2.2 在线数据驱动​​的优化

与离线数据驱动的优化不同,在线数据驱动的优化在进化搜索过程中有新的数据,可以利用在线数据来提高代理模型的质量。因此,如何利用在线数据是在线算法性能的关键。请注意,离线数据会影响性能,5.2.1节介绍的处理离线数据的技术也可以应用于在线算法。

实际上,在线数据驱动优化有两种情况:在线数据是否可以被算法控制<sup>[170]</sup>。在数据流的情况下,在线数据驱动的进化算法无法控制数据的生成,因此主要难点是如何从动态数据中捕获信息。

当在线数据驱动的进化算法可以主动对在线数据进行采样时,如果应用适当的模型管理策略,那么算法性能可以得到很大的提升。在这种情况下,在线数据采样的频率和选择都需要在优化过程中进行调整。在线数据采样的目的是以经济的方式提高代理模型的全局和局部精度,从而提高优化性能<sup>[187-189]</sup>。现有在线代理管理方法的更多详细信息将在 5.3 节介绍。

## 5.3 在线代理模型管理方法

在在线数据驱动的代理辅助进化优化中,假设在优化过程中可以主动收集一些额外的数据来更新代理并指导搜索。第一个问题当然是为什么在优化过程中需要收集额外的数据。这主要是因为,在优化之前训练的代理模型时可能会引入不是原始目标函数真正最优的最优值,称为假最优值<sup>[188]</sup>。图 5.2 是一个代理模型引入的错误最小值的示例,如果进化算法在代理模型  $\hat{f}(x)$  而不是真正的目标函数  $f(x)$  上搜索,代理模型  $\hat{f}(x)$  引入了一个假的最小值  $x^\Delta$ 。因此算法会终止于这个距离真正最小值  $x^*$  很远的差解  $x^\Delta$ 。

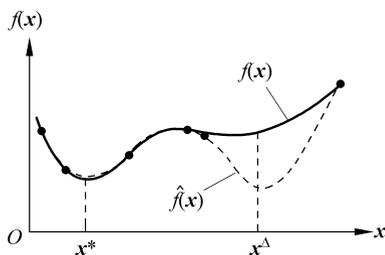


图 5.2 代理模型引入的错误最小值的示例

因此,代理模型管理是一个主动查询新数据点和更新代理的过程,这对于代理模型辅助进化优化至关重要。在代理模型管理中,通过执行数值仿真或物理实验递归地确定一个或多个要查询的解决方案是最重要的。目前已经提出了不同的选择解作为采样点的方法,通常通过选择代理模型预测的有希望的解,或者那些预测的适应度最不确定的解,或者群体的代表性解。代理模型管理方法可分为基于种群、基于世代和基于个体的方法。

### 5.3.1 基于种群的模型管理

模型管理的一个早期想法是基于种群,其中多个种群同时共同进化,每个种群都使用不同精度的模型。在最简单的情况下,涉及两个种群:一个使用代理模型作为适应度函数;另一个使用真实的适应度函数,即耗时的数值模拟或昂贵的物理实验。以空气动力学优化为例,可以使用 3 种方法评估设计的涡轮叶片的压力损失,即三维 Navier-Stokes 求解器、二维 Navier-Stokes 求解器或代理模型(例如神经网络)。在这几种适应度评估方法中,在准

确性和计算复杂度之间存在权衡,即评价方法越准确,进行评价的计算量就越大。在这种情况下,可以使用3个种群进行进化优化,每个种群使用不同的评价方法。如图5.3所示,其中实心点表示高精度评价方法,例如,三维 Navier-Stokes 求解器;阴影点表示中等精度评价方法,例如,二维 Navier-Stokes 求解器;圆圈表示低精度代理模型。种群之间交换个体的方式有两种方式。如图5.3(a)所示,第一种方法使用较低精度评价方法的种群中的个体发送到使用较高精度方法的种群。相比之下,第二种方法基于中等精度方法进化的个体与使用较低或较高精度评价方法的个体交换。这里的假设是,通过使用不同精度的模型共同进化3个种群,与仅采用最高精度的评价方法的情况相比,可以加快优化速度。在基于种群的方法中,一旦使用最高精度方法进行了适应度评估,代理模型就将被更新。使用多精度数据进行训练的模型方法(例如,协同 Kriging 法)也可能会有所帮助。

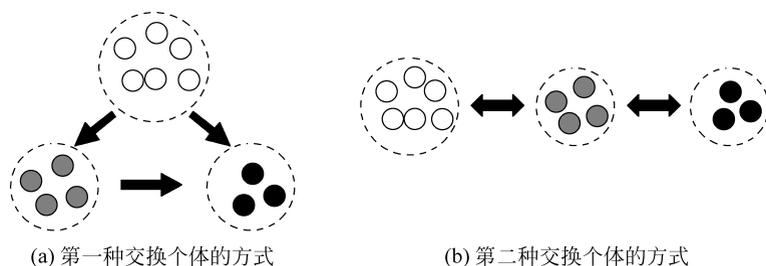


图 5.3 基于种群的模型管理

### 5.3.2 基于世代的模型管理

顾名思义,基于世代的模型管理策略旨在按代来管理代理模型。最简单的想法是,优化开始之前收集的一些数据来训练代理模型。然后对代理模型执行优化,直到它收敛。然后,将使用真实的适应度评估方法评价收敛解。接着,代理模型被更新之后是对代理模型的另一轮搜索。图5.4提供了基于世代的代理模型管理的流程图。

需要注意的是,上述基于简单生成的模型管理框架存在模型陷入局部最优的风险。因此,建议除了通过对代理模型进行搜索找到的最佳值之外增加一些随机解作为样本。

上述方法的一个直接变体是在每个给定的代数中切换评价方法。例如,使用昂贵的适应度评估方法进行5代优化,然后使用这些代生成的数据更新代理模型,并在代理模型上运行优化20代。这个过程一直重复,直到计算资源用完。文献[188]中提出了一种基于自适应代的代理管理方法,旨在调整使用真实适应度评估的频率。这里,定义了指定固定代数的循环。在一个周期内,使用昂贵的适应度评估(例如,计算流体动力学模拟)的初始频率是预先定义的。例如,一个循环包含10代,其中5代使用计算流体动力学模拟,接下来的5代使用代理模型。在每个周期结束时,根据这5代种被评价的解来估计模型的平均近似误差,并将之作为代理模型的精度。根据平均误差的变化,调整使用昂贵的适应度评估的频率。也

就是说,如果平均误差减小,那么可以降低使用昂贵适应度的频率;相应地,如果平均误差增加,那么使用昂贵适应度的频率就会增加。注意,在调整过程中,最小频率为1,即原始适应度函数在每个循环的至少一代中使用。图5.5给出了自适应的基于世代的代理模型管理的说明。

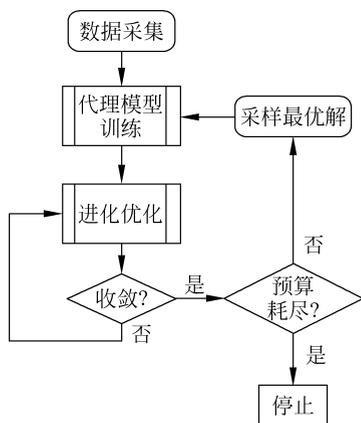


图 5.4 基于世代的代理模型管理

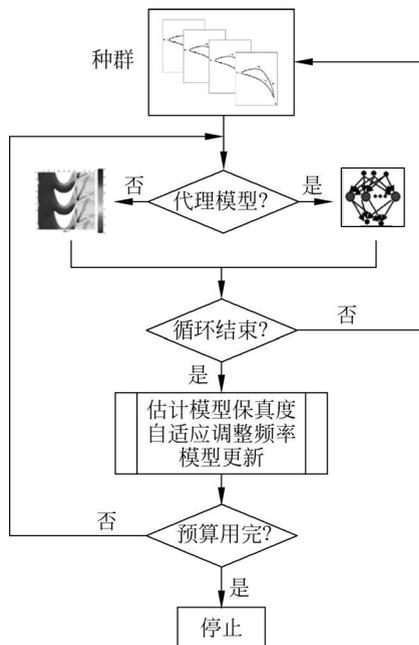


图 5.5 自适应的基于世代的代理模型管理的说明

### 5.3.3 基于个体的模型管理

基于个体的模型管理可能是最灵活和最广泛使用的代理模型方法。目前已经提出了基于个体的模型管理的不同类型,大致可分为预选法、随机策略、最佳策略、基于聚类的方法和基于不确定性的方法。

(1) 预选是一种筛选方法,它使用代理来减少随机性并加速进化。例如,对于种群大小为  $N$  的遗传算法,可以生成  $2N$  个后代,而不是生成  $N$  个后代。然后,可以使用经过训练的代理模型来选择更好的  $N$  个个体,然后将使用昂贵的适应度评估方法重新评估。类似地,对于  $(\mu, \lambda)$  进化策略,可以生成  $\lambda'$  后代,而不是从  $\mu$  父代个体生成后代,然后使用昂贵的适应度函数选择更好的  $\lambda$  后代个体进行重新评估。

(2) 随机策略不会像预选那样扩大种群规模。取而代之的是,从后代种群中随机选择指定数量的个体进行评估,使用昂贵的适应度函数。对于种群规模  $N$ ,可以选择  $N' < N$  个体。

(3) 最佳策略。与随机策略相反,使用原始适应度函数从  $N'$  个后代个体中选择最好的

$N$  个个体进行评价。通过对随机策略和最佳策略的经验比较<sup>[188]</sup>,可以发现,最佳策略是一种更有效的模型管理方法,因为当使用真实适应度函数评估种群中大约 50% 的较好个体时,种群将收敛到正确的最优值。相比之下,如果使用随机策略,至少 80% 的个体必须使用真实适应度函数评价总数才能收敛到正确的最优值。

(4) 聚类方法的目的是减少使用昂贵的真实适应度函数评价的个体数量,方法是将种群分组为多个类,然后仅使用真实适应度函数评估最靠近每个聚类中心的个体。每个类中的其余个体是根据到聚类中心的距离或使用正常代理模型来估计的。或者,可以将最佳策略应用于每个类,以便使用昂贵的适应度函数仅评价每个类中个体的一个子集<sup>[190]</sup>。

(5) 基于不确定性的策略。虽然使用真实适应度函数选择由代理预测的最佳个体进行评估是很有意义的,但从优化和模型更新的角度来看,选择远离现有训练样本的个体也是合理的。对于模型更新,那些远离存在训练样本的点可能有效提高模型质量,而对于搜索,在这些未探索区域采样解将促进探索。估计个体预测适应度不确定性的一种想法是计算解与所有训练数据之间的倒置距离<sup>[189]</sup>:

$$\delta_i = \frac{1}{\sum_{j=1}^m \frac{1}{\| \mathbf{x}_i - \mathbf{x}^{(j)} \|}} \quad (5.1)$$

其中, $\delta_i$  是第  $i$  个个体的不确定性程度, $\mathbf{x}^{(j)}$  是第  $j$  个训练数据, $m$  是训练样本总数, $\| \cdot \|$  通常情况是欧氏距离。

与选择最不确定的解的想法不同,在选择个体进行评估时,可以尝试最大限度地降低总体的总体不确定性。也可以将个体的质量与其不确定性相结合:

$$r_j = \sum_{i \in Q_j} p_i \delta_i \quad (5.2)$$

其中, $p_i$  是第  $i$  个个体被选中的概率,通常与其预测的适应度值成正比。

### 5.3.4 模因算法中的信任域方法

将进化搜索与局部搜索相结合的模因算法也可以通过代理模型加速。在代理模型辅助模因算法中,通常局部搜索由代理模型辅助进行。为此,基于信任域的方法是在传统优化中发展起来的一种基于黑盒模型的搜索方法,在代理模型辅助模因算法中经常被采用。Powell 在文献[52]中提出的信任域方法是一种迭代优化方法,在非线性优化领域发挥着重要作用。假设优化问题的一个解  $\mathbf{x}$ ,在信任域方法中,将在该解附近构建模型,并通过求解以下问题获得最优步长  $\mathbf{s}$ :

$$\begin{aligned} \min q(\mathbf{s}) &= f(\mathbf{x}) + \mathbf{g}^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T H \mathbf{s} \\ \text{s. t.} \quad & | \mathbf{s} | \leq \delta \end{aligned} \quad (5.3)$$

其中,  $\mathbf{g} = \nabla f(\mathbf{x})$ ,  $\mathbf{H}$  是  $n \times n$  的  $\nabla^2 f(\mathbf{x})$  对称近似矩阵,  $\delta > 0$  是信任域半径大小。一旦最优步长  $\mathbf{s}_b$  在本次迭代中被选择,  $\mathbf{x} + \mathbf{s}_b$  将会被评价。真实函数下降  $f(\mathbf{x}) - f(\mathbf{x} + \mathbf{s})$  与其预测值下降  $q(0) - q(\mathbf{s})$  的比率  $\rho$ , 即

$$\rho = \frac{f(\mathbf{x}) - f(\mathbf{x} + \mathbf{s})}{q(0) - q(\mathbf{s})}$$

在决定步骤是否应该可接受以及如何为下一次迭代调整信任区域半径方面起着重要作用。如果  $\rho$  接近 1, 则说明模型比较好, 区域可以放大; 相反, 如果  $\rho$  太小, 则区域需要缩小。具体来说, 给定一个阈值  $\eta$ , 如果  $\rho < \eta$ , 则  $\mathbf{s}$  被拒绝,  $\mathbf{x}(t+1) = \mathbf{x}(t)$ , 且  $\delta(t+1) = \gamma\delta(t)$ ; 如果  $\rho \geq \eta$ , 则  $\mathbf{s}$  被接受,  $\mathbf{x}(t+1) = \mathbf{x} + \mathbf{s}$ , 且  $\delta(t+1) \geq \delta(t)$ 。其中,  $\gamma$  是一个预设参数。

算法 5.1 给出了信任域方法的伪代码。在算法 5.1 中,  $\eta$ ,  $\gamma_i$  和  $\gamma_d$  为预设参数,  $0 < \eta$ ,  $\gamma_d < 1$  且  $\gamma_i \geq 1$ 。在算法 5.1 中, 模型将首先在给定的信任域  $\delta$  中构建。然后对模型进行求解, 这是信任域方法中非常重要的一步。目前已经提出了许多优化方法来有效地解决这个问题。然而, 当优化问题的维度越来越大时, 一些方法变得越来越低效。

---

算法 5.1 信任域方法伪代码

---

```

输入:  $\mathbf{x}$ , 优化问题的一个解
输出:  $\mathbf{x}$ 
1:  $k = 0$ 
2: while 终止条件未达到 do
3:   用式(5.3)构建模型  $q(\mathbf{s})$ 
4:   搜索最优步长  $\mathbf{s}_b$ 
5:   if  $\rho \geq \eta$  then
6:      $\mathbf{x} = \mathbf{x} + \mathbf{s}_b, \delta = \gamma_i$ 
7:   else
8:      $\delta = \gamma_d$ 
9:   end if
10:   $k = k + 1$ 
11: end while

```

---

## 5.4 贝叶斯模型管理

贝叶斯优化(Bayesian Optimization, BO)是一类基于机器学习的优化方法, 用于解决黑盒问题或计算昂贵问题。一般来说, 贝叶斯优化建立一个目标函数的概率表示, 并试图找到一个有价值的位置进行真实的目标评价, 以便在计算资源非常有限的情况下找到最优解。贝叶斯优化的概率表示称为代理模型, 用于近似目标函数。获取函数利用后验知识来引导采样以进行昂贵的目标评价。算法 5.2 给出了贝叶斯优化的伪代码。首先将对多个位置进行采样, 并在评价后用于训练代理模型。然后优化激活函数搜索有价值的解, 该解将使用真实的目标函数评价, 并进一步用于更新代理模型。

贝叶斯优化最流行的代理模型是高斯过程模型, 4.2.5 节已对其进行了介绍, 在此不再重复。获取函数考虑探索和开采之间的权衡, 因此其用来确定决策空间中哪个数据点使用

真实的昂贵目标函数进行评价。一般地,位置  $\mathbf{x}_t = \underset{\mathbf{x}}{\operatorname{argmax}} \mathcal{A}(\mathbf{x} | \mathcal{D}_{1:t-1})$  会被采样,其中  $\mathcal{A}$  表示获取函数,  $\mathcal{D}_{1:t-1} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_{t-1}, y_{t-1})\}$  表示存储所有被采样并被真实评价过的数据集。流行的获取函数也称作填充准则,包括改进概率 (Probability of Improvement, PI)、期望改进 (Expected Improvement, EI)、置信上界 (Upper Confidence Bound, UCB)。

算法 5.2 贝叶斯优化伪代码

---

```

1: 在决策空间采样  $N_I$  个数据并使用黑盒或昂贵问题进行评价;
2: 从  $N_I$  个数据中选择最好的位置, 记为  $\mathbf{x}_b$ ;
3: 训练代理模型  $M$ ;
4: while  $N_I \leq N_{FE}$ 
5:     优化获取函数  $\mathcal{A}$  得到其最优位置;
6:     使用黑盒或昂贵问题评价该位置;
7:     更新到目前为止找到的最好位置  $\mathbf{x}_b$ ;
8:     更新代理模型;
9: end while
10: 输出最优解  $\mathbf{x}_b$ ;

```

---

## 5.4.1 获取函数

### 1. 改进概率

定义改进概率为

$$\text{PI}(\mathbf{x}) = \psi\left(\frac{f(\mathbf{x}_b) - \mu(\mathbf{x}) - \xi}{\sigma(\mathbf{x})}\right) \quad (5.4)$$

其中,  $\mu(\mathbf{x})$  和  $\sigma(\mathbf{x})$  是在  $\mathbf{x}$  处的均值和方差;  $f$  是待优化的目标函数;  $\mathbf{x}_b$  是估计的最小点;  $\xi$  是控制探索程度的参数;  $\psi(\cdot)$  表示标准高斯分布的累积分布函数。

### 2. 期望改进

定义期望改进为

$$\text{EI}(\mathbf{x}) = (f(\mathbf{x}_b) - \mu(\mathbf{x}) - \xi) \psi\left(\frac{f(\mathbf{x}_b) - \mu(\mathbf{x}) - \xi}{\sigma(\mathbf{x})}\right) + \sigma(\mathbf{x}) \phi\left(\frac{f(\mathbf{x}_b) - \mu(\mathbf{x}) - \xi}{\sigma(\mathbf{x})}\right) \quad (5.5)$$

其中,  $\phi(\cdot)$  表示标准高斯分布的密度函数。

### 3. 置信下界

定义置信下界为

$$\text{LCB}(\mathbf{x}) = \mu(\mathbf{x}) - \beta\sigma(\mathbf{x}) \quad (5.6)$$

其中,  $\beta$  是控制探索的参数。

## 5.4.2 进化贝叶斯优化

贝叶斯优化不仅在黑盒优化领域非常流行,在机器学习方面也非常流行,这主要归功于



EBO-MOP  
代码



EBO-SOP  
代码

其数学上的可靠思想可用于模型管理,即获取函数。然而,贝叶斯优化的成功应用主要限制在低维系统和单目标优化上。首先,高斯过程的时间复杂度为训练样本数量的立方,这对于求解高维问题是不现实的。其次由于获取函数是分段线性和高度多模态的,这使得使用数学规划来优化获取函数是非常重要的。图 5.6 提供了一个说明性的例子,其中图 5.6(a)绘制了要估计的原始函数(用实线表示),用于训练的 9 个训练样本(用菱形表示),以及获得的高斯过程模型(虚线表示期望预测,阴影区域表示不确定度)。此外,图 5.6(b)绘制了获得的获取函数,用于选择下一个查询(采样)并添加到训练数据中的数据,它是高度多模态的。获取函数值最大的解用三角形表示,该解采样后被添加到训练集中。

最后,贝叶斯优化是针对单目标优化而提出的,这使它不能直接应用于多目标优化中,特别是当目标数增多时。

由于上述原因,元启发式算法被广泛应用于求解单目标或多目标贝叶斯优化中的获取函数。图 5.7 展示了使用进化算法求解贝叶斯优化中获取函数的一般框架,称为进化贝叶斯优化。

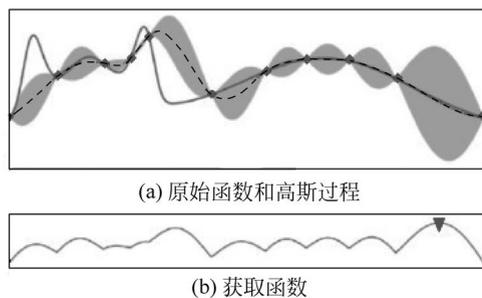


图 5.6 贝叶斯优化示例

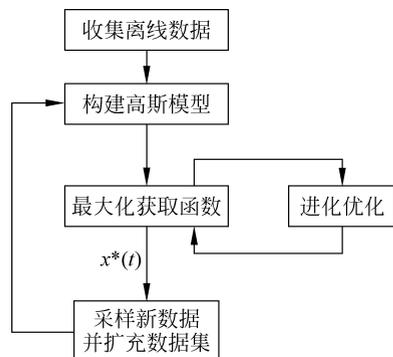


图 5.7 进化贝叶斯优化框架

注意,对于多目标优化提出了不同的求解想法,要么将多目标优化问题转换为单目标问题,从而使用传统的获取函数,要么根据不同目标的获取函数值进行非支配排序,然后选择采样点。更多细节将在第 7 章讨论。

### 5.4.3 贝叶斯进化优化

在贝叶斯优化中使用进化算法优化获取函数与在进化优化中使用获取函数作为基于个体的模型管理策略之间的区别还没有被关注到。对于后者,优化是运行在代理模型上的,而不进行对获取函数的优化。相反,子代种群中每个个体的获取函数值都要进行计算,然后选择具有最大获取函数值的那些个体(一个或多个)进行采样(即使用实际目标函数进行适应度评价)。图 5.8 为使用获取函数作为模型管理的数据驱动进化优化框架。

进化贝叶斯优化(Evolutionary Bayesian Optimization, EBO)和贝叶斯进化优化(Bayesian



BEO-MOP  
代码



BEO-SOP  
代码

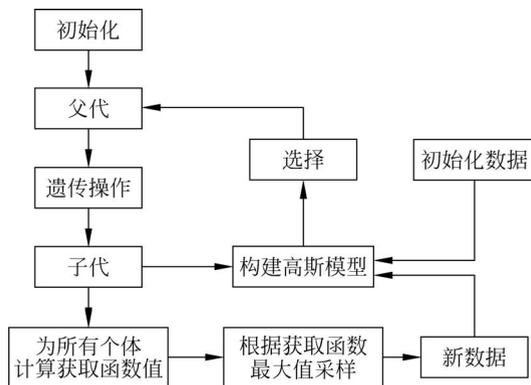


图 5.8 贝叶斯进化优化框架

Evolutionary Optimization, EBO) 之间存在着一些差异。在 EBO 中, 最优值的搜索基本上是单点搜索, 而获取函数的优化则使用基于种群的搜索。一旦找到获取函数的最优解, 即当种群收敛时, 优化解将被采样并在高斯过程模型更新之前添加到训练数据中。然后, 获取函数也被更新, 并且获取函数的新一轮优化又重新开始。重复此过程, 直到计算资源耗尽。

相反, 在 BEO 中, 进化算法运行在高斯过程上, 期望值用于适应度评估的位置选择。若干代之后, 一般在种群收敛之前, 对子代种群中的每个解计算获取函数值, 并对具有最大获取函数值的解进行采样并添加到训练数据中。随后, 更新高斯过程模型并继续进化搜索。图 5.9 展示了 EBO 和 BEO 中解的选择差异。对于 EBO, 进化算法用于寻找获取函数值最大的解, 记作  $x^{\text{EBO}}$ 。而对于 BEO, 在种群收敛到当前高斯过程模型的最优值之前计算种群的获取函数值, 因此具有最大获取函数值的子代个体为  $x^{\text{EBO}}$ 。因此, 新采样的解会有所不同, 导致下一轮搜索中的高斯过程模型不同。文献[191]对 BEO 和 EBO 在多目标优化中的有效性进行了经验对比, 尽管经验分析的范围可能有限, 但结果表明 BEO 更有效。

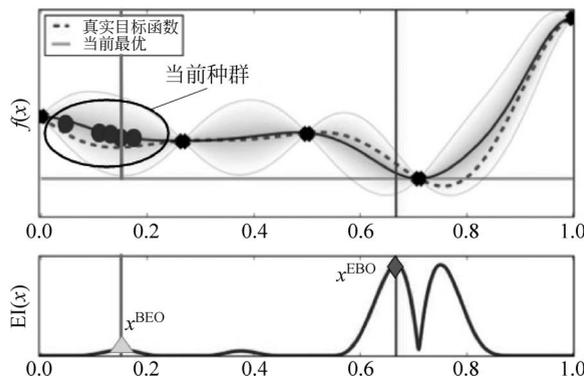


图 5.9 BEO 和 EBO 中采样解比较

## 5.5 贝叶斯约束优化

在许多实际工程问题中,优化问题的约束往往是黑盒的或计算成本高昂的。因此,提出了使用代理模型来替换约束函数,从而在搜索优化问题可行最优解时节省计算成本。通常,建立约束优化问题的代理模型可采用3种方法。

- (1) 为一个约束函数构建一个代理模型。
- (2) 为所有约束函数构建一个代理模型。
- (3) 建立分类模型,用于区分可行解和不可行解。

理想情况下,构建的代理模型尽可能地接近真实约束函数。然而,类似于为目标函数训练模型,一般很难很好地拟合原约束函数。如果一个可行解被估计为一个不可行解,那么可能无法找到真正的最优可行解。相反,如果一个不可行解被估计为一个可行解,那么其搜索方向可能会被误导。因此,选择使用真实约束函数评价的解非常重要,这些解将用于更新代理模型以辅助尽可能准确地估计一个解的可行性。通常,约束值的估计准确性可以通过对具有最大估值不确定性的解进行真实评价来提高。然而,约束问题优化的主要目标是寻找最优可行解。因此,在进行昂贵的约束问题优化时,应同时考虑算法的探索和开采平衡以及约束函数估值准确性的提高。通常,昂贵约束问题的填充准则可以分为两类:一类是在填充准则中考虑目标函数和约束条件,另一类是单独考虑目标函数和约束条件从而寻找使用正确目标函数和约束函数评价的解。接下来,我们将对这两类准则进行详细说明。

### 5.5.1 约束优化的获取函数

使用惩罚函数将一个约束优化问题转换为一个无约束问题是求解约束问题的常用方法。因此,在昂贵约束问题的优化中,可以为无约束问题建立代理模型,并且可直接采用为无约束优化提出的填充准则来求解昂贵约束优化问题。

然而,还有一些方法主要致力于解决优化过程中的约束问题。由于高斯过程模型可以同时提供估值均值和估值不确定性,因此提出了一些使用高斯过程来分别估计目标函数和约束函数值的方法,并且提出了不同的同时考虑目标函数和约束条件的填充准则,用于选择有价值的解进行真实的目标函数和约束函数评价。在高斯过程模型辅助的约束优化中,解的每个约束都通过高斯过程模型进行估计,为此其具有可行性的概率。因此,它被称为概率方法。在概率方法中,约束优化问题可以作为一个无约束优化问题,通过最大化改进期望乘积和可行性概率来表示和解决:

$$\max_{\mathbf{x}} EI(\mathbf{x}) \times P(\hat{g}(\mathbf{x}) \leq 0) \quad (5.7)$$

其中,

$$I(\mathbf{x}) = \max\{f^* - \hat{f}(\mathbf{x}), 0\} \quad (5.8)$$

$$\text{EI}(\mathbf{x}) = (f^* - \hat{f}(\mathbf{x}))\psi\left(\frac{f^* - \hat{f}(\mathbf{x})}{s_f(\mathbf{x})}\right) + s_f(\mathbf{x})\phi\left(\frac{f^* - \hat{f}(\mathbf{x})}{s_f(\mathbf{x})}\right) \quad (5.9)$$

以及

$$P(\hat{g}(\mathbf{x}) \leq 0) = \psi\left(\frac{g^* - \hat{g}(\mathbf{x})}{s_g(\mathbf{x})}\right) \quad (5.10)$$

在式(5.8)~式(5.10)中,  $\text{EI}(\mathbf{x})$ 是解  $\mathbf{x}$  在其目标上的改进期望,  $P(\hat{g}(\mathbf{x}) \leq 0)$ 表示可行性概率。  $f^*$  和  $g^*$  分别表示到目前为止最优适应度和最优约束冲突值。  $\hat{f}(\mathbf{x})$ 和  $\hat{g}(\mathbf{x})$ 分别表示适应度和约束函数值的估值,  $s_f(\mathbf{x})$ 和  $s_g(\mathbf{x})$ 分别表示目标函数和约束函数预测方差的平方根。  $\psi(\cdot)$ 是标准正态累积函数,  $\phi(\cdot)$ 是标准概率密度函数。请注意, 如果存在多个约束, 那么可行性概率是个体概率的乘积。因此, 概率方法的数学模型是:

$$\max_{\mathbf{x}} \text{EI}(\mathbf{x}) \times \prod_{i=1}^l P(\hat{g}_i(\mathbf{x}) \leq 0) \quad (5.11)$$

其中,  $l$  表示约束的数量。

从式(5.11)可以看出, 如果任何约束的可行性概率非常低, 那么约束 EI 值将接近 0。因此, 在这种情况下, 式(5.11)不能作为填充准则来选择解以进行真实的昂贵评价。 Jiao 等<sup>[192]</sup>提出了一种改进的概率方法, 针对两种情况分别使用了不同的填充准则来选择使用昂贵目标函数和约束函数进行评价的解。其中一个填充准则用于没有可行解的情况, 其数学模型为

$$\max \text{EI}(\mathbf{x}) = \int_0^{g^*} \prod_{i=1}^l \psi\left(\frac{z - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) dz - g^* \times \prod_{i=1}^l \psi\left(\frac{-\hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) \quad (5.12)$$

当存在可行解时, 则使用另一个填充准则, 其数学表达式为

$$\text{EI}(\mathbf{x}) = \left[ (f^* - \hat{f}(\mathbf{x}))\psi\left(\frac{f^* - \hat{f}(\mathbf{x})}{s_f(\mathbf{x})}\right) + s_f(\mathbf{x})\phi\left(\frac{f^* - \hat{f}(\mathbf{x})}{s_f(\mathbf{x})}\right) \right] \times \prod_{i=1}^l \psi\left(\frac{-\hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) \quad (5.13)$$

### 5.5.2 两阶段获取函数

针对昂贵约束优化的另一种类型的填充准则由 COBRA(Constrained Optimization By Radial basis function interpolation)和扩展的 ConstrLMSRBF<sup>[193]</sup>提供, 两者都使用了两阶段方法, 其中第一阶段用于找到可行点, 第二阶段旨在改善可行点。在 COBRA 中, 通过在第一阶段优化以下目标函数来寻找使用昂贵函数评价的可行点。

$$\min \sum_{i=1}^l \max\{\hat{g}_i(\mathbf{x}), 0\}$$

$$\begin{aligned}
\text{s. t. } \quad & \mathbf{a} \leq \mathbf{x} \leq \mathbf{b} \\
& \hat{g}_i(\mathbf{x}) + \varepsilon_i \leq 0, \quad i = 1, 2, \dots, l \\
& \|\mathbf{x} - \mathbf{x}_j\| \geq \rho, \quad j = 1, 2, \dots, n
\end{aligned} \tag{5.14}$$

其中,  $\varepsilon_i$  表示解  $i$  的边缘, 其用于强迫解远离 RBF 约束边界, 从而增加进入可行区域的机会。  $\rho$  是解远离其先前位置所需距离的阈值,  $\rho = \gamma \ell([\mathbf{a}, \mathbf{b}])$ , 其中,  $0 < \gamma < 1$  称为距离需求因子,  $\ell([\mathbf{a}, \mathbf{b}])$  是区域  $[\mathbf{a}, \mathbf{b}] \subseteq \mathbf{R}^D$  最小边的长度。

在第二阶段, 基于目标函数和约束函数的 RBF 模型辅助, 通过优化以下数学模型来搜索更好的可行点。

$$\begin{aligned}
\min \quad & \hat{f}(\mathbf{x}) \\
\text{s. t. } \quad & \mathbf{a} \leq \mathbf{x} \leq \mathbf{b} \\
& \hat{g}_i(\mathbf{x}) + \varepsilon \leq 0, \quad i = 1, 2, \dots, l \\
& \|\mathbf{x} - \mathbf{x}_j\| \geq \rho, \quad j = 1, 2, \dots, n
\end{aligned} \tag{5.15}$$

对于式(5.15)的最优解将使用昂贵的目标函数和约束函数进行评价, 然后边界将被调整。请注意, 在 COBRA 中, 如果给出了或者通过空间填充设计获得了一个初始可行解, 那么 COBRA 方法将只运行第二阶段。

扩展的 ConstrLMSRBF 是 ConstrLMSRBF<sup>[193]</sup> 的一种扩展, 是一种基于启发式代理模型的方法, 并假设在初始点中存在一个可行解。与 COBRA 类似, 扩展 ConstrLMSRBF 也是两阶段结构, 其中第一阶段用于寻找可行点, 第二阶段期望找到更好的可行解。在扩展 ConstrLMSRBF 的第一阶段, 如果当前最优解不可行, 那么将通过当前最优位置的部分或所有组成进行正态扰动来生成一组随机候选点。然后, 在所有候选点中预测的约束冲突数最少的解将使用昂贵的目标函数和约束函数进行评价, 并用于更新到目前为止找到的最优解。请注意, 如果有多个解具有相同的最少约束冲突数, 则将对具有最大约束冲突值最小的解使用昂贵函数进行评价。继续迭代直到找到一个可行解, 然后运行和 COBRA 第二阶段相同的第二阶段。

## 5.6 代理模型辅助的鲁棒性优化

### 5.6.1 鲁棒性优化的双目标公式

使用显式平均方法搜索鲁棒最优解需要额外的适应度评价。因此若单次适应度评估比较耗时, 则将带来很大的问题。一种解决方法是引入基于代理模型的适应度估计, 以辅助单目标和多目标鲁棒优化。基于代理模型的鲁棒优化其双目标公式表示为

$$\hat{f}_{\text{exp}}(\mathbf{x}^0) = \frac{1}{d} \sum_{i=1}^d \hat{f}(\mathbf{x}_i) \tag{5.16}$$

$$\hat{f}_{\text{var}}(\mathbf{x}^0) = \frac{1}{d} \sum_{i=1}^d [\hat{f}(\mathbf{x}_i) - \hat{f}_{\text{exp}}(\mathbf{x}^0)] \quad (5.17)$$

其中,  $\mathbf{x}^0$  是鲁棒性待估计的解,  $\hat{f}(\cdot)$  是近似适应度,  $\hat{f}_{\text{exp}}(\mathbf{x}^0)$  是待估计的期望适应度,  $\mathbf{x}_i = \mathbf{x}^0 + \delta_i$ ,  $\delta_i$  为扰动,  $d$  为样本数。样本可以利用拉丁超立方体采样来产生。

## 5.6.2 代理模型的构建

不同的代理模型构建技术如下<sup>[194]</sup>：

(1) **单代理模型**。在这种情况下, 每个个体建立一个模型, 即模型的拟合点与决策空间中个体的位置相同, 并且代理模型将用于所有样本点以估计期望的适应度和适应度方差。由于依赖于使用的模型, 单个代理模型可能在期望的扰动范围  $\delta$  内不能很好地工作。

(2) **最近模型**。这也是围绕每个个体建立单个代理模型, 然而, 最近模型会用于估计一个样本的适应度。请注意, 最近模型并不一定是和个体相关联的那个模型。

(3) **集成代理模型**。在这种方法中, 样本点 ( $\mathbf{x}_s$ ) 的函数值将通过模型的加权组合来进行估计, 这些模型对应于  $k$  个最近拟合点。期望的适应度可以通过以下公式进行估计：

$$\hat{f}_{\text{ENS}}(\mathbf{x}_s) = \frac{1}{\sum_{1 \leq i \leq k} \omega_i} \sum_{1 \leq i \leq k} \omega_i \hat{f}_i(\mathbf{x}_s)$$

$$\omega_i = \frac{1}{\|\mathbf{x}_{\text{fp}} - \mathbf{x}_s\|_2} \quad (5.18)$$

其中,  $\omega_i$  表示权重,  $k$  为集成大小,  $\mathbf{x}_{\text{fp}}$  是对应模型的拟合点。

(4) **多模型**。在该方法中, 每个样本周围构建单独模型, 其用于估计样本的适应度。

## 5.7 模型的性能指标

衡量代理模型的质量并不简单, 因为代理模型的目的不是准确地近似原始适应函数, 而是有效引导搜索。

如图 5.10 所示, 代理模型  $\hat{f}(\mathbf{x})$  没有很好地拟合原始目标函数  $f(\mathbf{x})$ 。然而, 如果在代理模型上运行优化, 那么  $\hat{f}(\mathbf{x})$  的最小值会非常接近  $f(\mathbf{x})$  的最小值。从这个角度来看,  $\hat{f}(\mathbf{x})$  是一个完美的代理模型。因此, 代理模型的性能指标应该与机器学习中的性能指标不同, 在机器学习中对没见过的数据的预测准确性或分类准确性是非常重要的。

### 5.7.1 精度

最广泛使用的度量代理模型质量的方法是个体的原始适应度函数  $f(\mathbf{x})$  以及代理模型的输出  $\hat{f}(\mathbf{x})$  之间的均方误差：

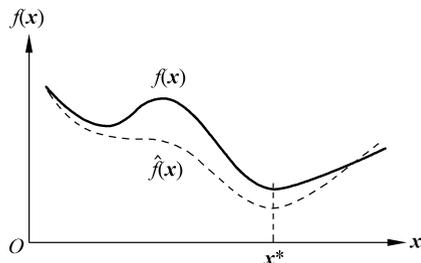


图 5.10 具有相对较差估计质量的代理模型可以很好地引导向原始问题真正的最优解( $x^*$ )进行搜索

$$E = \frac{1}{d} \sum_{j=1}^d (\hat{f}(\mathbf{x}_j) - f(\mathbf{x}_j))^2 \quad (5.19)$$

均方误差是用于质量度量估计的  $d$  个个体的平均,例如,在基于个体的模型管理中,  $n = \lambda$  个子代个体,或在一个控制周期,  $n = \eta\lambda$  个控制个体。请注意,在减少过拟合时应考虑质量度量。

一般来说,具有好的估值准确性的模型可以确保个体的适应度能得到正确估计,从而使好的个体保留到下一代。然而,从进化优化的角度来看,如果个体的排序是正确的,能够确保正确选择,那么个体的实际适应度其实并不重要。接下来,将介绍一些关于正确选择的性能指标而不是模型的准确性。请注意,某些度量的定义依赖于环境选择策略。

### 5.7.2 基于选择的性能指标

在下面的讨论中,以  $(\mu, \lambda)$ -选择,  $\lambda \geq 2\mu$  的连续优化为例。原则上,这可以扩展到任何其他选择策略。

第一种度量是基于估值模型正确选择的个体数:

$$\rho^{(\text{sel})} = \frac{\xi - \langle \xi \rangle}{\mu - \langle \xi \rangle} \quad (5.20)$$

其中,  $\xi (0 \leq \xi \leq \mu)$  是正确选择的个体数,也就是说,使用模型选出的这些个体中包含使用原始适应度函数进行评价后选中的个体。在随机选择的情况下,  $\xi$  的期望

$$\begin{aligned} \langle \xi \rangle &= \sum_{m=0}^{\mu} m \frac{\binom{\mu}{m} \binom{\lambda - \mu}{\mu - m}}{\binom{\lambda}{\mu}} \\ &= \frac{\mu^2}{\lambda} \end{aligned} \quad (5.21)$$

用于标准化。可以看出,如果所有  $\mu$  个父代个体全部被正确选择,那么该度量会达到其最大值  $\rho^{(\text{sel})} = 1$ ,若该度量值为负值,则表示基于代理模型选择的个数会比随机选择还要差。

$\rho^{(\text{sel})}$  度量只计算正确选择个体的绝对数。然而,若  $\rho^{(\text{sel})} < 1$ ,则该度量不能表示第  $(\mu + 1)$

个或者最差的子代个体是否被选用,从而显著影响进化优化过程。

接下来,扩展  $\rho^{(\text{sel})}$  度量使其包括所选个体的等级,这是根据原始适应度函数计算的。若基于代理模型选用个体的等级高于根据基于原始适应度函数选用的个体等级平均值,则我们认为代理模型是好的。

扩展  $\hat{\rho}^{(\text{sel})}$  度量的定义如下。如果基于原始适应度函数的第  $m$  个最好个体选用了,那么代理模型的得分为  $\lambda - m$ 。因此,代理模型的质量可以通过选用个体的累加得分来表示,记为  $\pi$ 。请注意,如果所有的  $\mu$  个个体都被正确选择,那么  $\pi$  达到最大值:

$$\begin{aligned}\pi^{(\text{max.})} &= \sum_{m=1}^{\mu} (\lambda - m) \\ &= \mu \left( \lambda - \frac{\mu + 1}{2} \right)\end{aligned}\quad (5.22)$$

与式(5.20)类似,在纯随机选择中, $\hat{\rho}^{(\text{sel})}$  度量通过线性转换  $\pi$ ,使用最大值  $\pi^{(\text{max.})}$  和期望  $\langle \pi \rangle = \frac{\mu\lambda}{2}$  来进行定义:

$$\hat{\rho}^{(\text{sel})} = \frac{\pi - \langle \pi \rangle}{\pi^{(\text{max.})} - \langle \pi \rangle} \quad (5.23)$$

### 5.7.3 等级相关性

代理模型的质量也可以通过等级相关性来衡量。等级相关性的定义为

$$\rho^{(\text{rank})} = 1 - \frac{6 \sum_{l=0}^{\lambda} d_l^2}{\lambda(\lambda^2 - 1)} \quad (5.24)$$

它是两个变量等级之间单调关系的一种度量。在这里, $d_i$  是第  $i$  个子代基于原始适应度函数  $f(\mathbf{x})$  与基于代理模型  $\hat{f}(\mathbf{x})$  的等级之间的差别。 $\rho^{(\text{rank})}$  的范围介于  $[-1, 1]$  区间。 $\rho^{(\text{rank})}$  值越大,两个变量等级之间具有正斜率的单调关系越强。与  $\rho^{(\text{sel})}$  相比,等级相关性不仅考虑所选个体的等级,还考虑所有个体的等级。这使其能够提供一个好的代理模型估计能力用于区分好的和坏的个体,这是代理模型辅助进化优化中实现正确选择的基础。

### 5.7.4 适应度相关性

适应度相关性是指代理模型预测的适应度与实际适应度之间的相关性,尽管其不是正确预测适应度的必需条件,但其提供了另一种代理模型是否能够确保正确选择的度量方法:

$$\rho^{(\text{corr})} = \frac{\frac{1}{d} \sum_{j=1}^d (\hat{f}_j(\mathbf{x}) - \tilde{f}(\mathbf{x})) (f_j(\mathbf{x}) - \bar{f}(\mathbf{x}))}{\sigma^f \sigma^{\hat{f}}} \quad (5.25)$$

其中,  $\tilde{f}(\mathbf{x})$  和  $\bar{f}(\mathbf{x})$  分别表示估计值和真实值的平均值,  $\sigma^f$  和  $\hat{\sigma}^f$  分别表示代理模型输出值和原始适应值函数的标准偏差。

相关性的特征与上述基于等级的度量和均方误差有关。它不是代理模型和原始适应度函数之间差异性的度量,但其计算了它们之间的单调关系。此外,这个度量的范围是已知的,因此相比于均方误差,  $\rho^{(\text{corr})}$  更容易进行计算。此外,  $\rho^{(\text{corr})}$  是可微的,其允许基于梯度的方法来自适应代理模型。

## 5.8 总结

本章提供了数据驱动进化优化的基本定义,并介绍了单目标优化中代理模型管理的初步思路。在接下来的章节中,我们将扩展这些基本的代理模型管理方法以应对各种挑战。