第 5 章 基于块坐标更新的高维多目标贝叶斯优化方法

5.1 引　　言

第 4 章主要考虑了如何能更充分合理地利用现实世界中的并行计算资源，以加快优化算法的收敛性；同时考虑了在批处理函数评估方式下，如何能更好地平衡贝叶斯优化中利用和探索之间的关系，进而提出具有双目标获取函数和自适应采样策略的 Adaptive Batch-ParEGO 算法。虽然该算法在实现并行化函数评估的同时，能够很好地平衡解的收敛性与多样性，但是该方法主要聚焦于低维决策空间中的昂贵的多目标优化问题。在高维决策空间中，由于维度灾难（Curse of Dimensionality）[21] 和边界问题（Boundary Issue）[29]，Adaptive Batch-ParEGO 的优化性能明显下降，导致优化效果很差。上述两个问题也是其他多目标贝叶斯优化方法面临的巨大挑战，包括基于 EGO、基于 HV 和基于预测熵的大多数多目标贝叶斯优化方法。此外，当前聚焦于高维多目标贝叶斯优化方法的相关研究较少。该类研究或对决策空间或目标空间的假设性过强，或没有解决因决策空间维度过高带来的维度灾难问题。所以，本章主要从上述两大挑战入手，对高维多目标贝叶斯优化方法展开研究，旨在缓解高维决策空间中的维度灾难和边界问题，有效地求解高维昂贵的多目标优化问题。

具体而言，受块坐标下降（Block Coordinate Descent，BCD）[323] 的启发，本章提出了一种有效的、泛化的基于块坐标更新的高维多目标贝叶斯优化方法（Multi-objective Bayesian Optimization with Block Coordinate Updates，Block-MOBO）。Block-MOBO 首先将高维决策变量空间划分为不同的块，并依次对这些块进行优化；不同算法迭代中块内的决策变量不同，每次迭代只考虑一个块用于构建高斯代理模型。在当前迭代中，未在块内的决策变量值与上下文向量生成策略近似。上下文向量通过从当前帕累托最优集中复制相应决策维度的值嵌入帕累托先验知识。然后，Block-MOBO 通过优化以贝叶斯和多目标方式构建的ϵ-贪心获取函数推荐下一个候选解，以解决边界问题。因此，候选解要么由具有平衡利用和探索之间关系的获取函数获得，要么以概率ϵ由嵌入帕累托支配关系即 θ-支配等级（θ-Dominance Rank）的获取函数得到。本章的主要贡献可概括

如下。

（1）提出了适用于高维昂贵多目标问题的 Block-MOBO，它采用块坐标更新降低决策空间的维度，每次算法迭代只考虑部分决策变量，其余变量值采用融入帕累托前沿知识的上下文向量生成。采用块坐标更新方法可以避免对高维决策空间和目标空间的过强假设，从而有效地缓解维度灾难。

（2）引入了ϵ-贪心获取函数，其从贝叶斯和多目标优化角度推荐下一个候选解；并且通过θ-支配等级融入帕累托支配关系，实现利用和探索两者之间的平衡，进而缓解边界问题导致的解质量低的问题。

（3）在三个标准合成的多目标测试问题上，对比了其他多目标贝叶斯优化方法，验证了 Block-MOBO 的有效性。

本章剩余部分组织如下。5.2 节阐述提出的基于块坐标更新的高维多目标贝叶斯优化方法框架和算法具体细节。5.3 节展示了实验设置和对比结果与分析。最后，第 5.4 节对本章内容做了总结。

5.2　基于块坐标更新的高维多目标贝叶斯优化的研究方法

本节主要详细介绍基于块坐标更新的高维多目标贝叶斯优化方法（Block-MOBO）的算法框架和具体细节。

5.2.1　算法框架

Block-MOBO 整体框架如算法 5.1 所示。具体而言，Block-MOBO 首先初始化种群X_0、权重向量$\boldsymbol{\Lambda}$、理想点z^*和最差点z^{nad}等（步骤 1～4）。然后，该方法用原昂贵的多目标优化问题评估X_0得到其真实目标函数值$F(X_0)$，并更新相关参数（步骤 5～8）。接下来，Block-MOBO 从D维决策变量维度中随机选择d维子决策变量维度进行优化（步骤 10）。然后，通过增广切比雪夫函数和θ-支配等级（θ-Dominance Rank）计算到目前为止的已评估解的标量代价值，用于构建高斯模型和ϵ-贪心获取函数$a_t(x)$（步骤 11～14）。接下来，Block-MOBO 通过最大化d维获取函数$a_t(x)$推荐下一个候选解（步骤 15），而其余的$D-d$维决策变量用上下文向量进行赋值，并将子决策空间重组为D维决策空间（步骤 16～17）。最后，Block-MOBO 对新的候选解进行评估，并更新高斯模型和相关参数（步骤 18～21）。在 Block-MOBO 中，步骤 10～21 不断迭代，直到满足算法终止条件 Eval＞MaxEvals 为止。

算法 5.1　Block-MOBO 整体框架

输入：昂贵多目标问题 MOP；最大真实函数评估次数 MaxEvals；块大小 d；初始观察点个数 N_{ini}
输出：近似帕累托最优集 P^*、近似帕累托前沿 PF^*

//初始化//
用 LHS[112] 生成 N_{ini} 个初始解：$X_0 \leftarrow \{x_1^1, x_1^2, \cdots, x_1^{N_{\text{ini}}}\}$
初始化权重向量 $\boldsymbol{\Lambda}$、聚合值选择概率 Prob_s、理想点 z^* 和最差点 z^{nad}
$\text{Eval} \leftarrow 0, t \leftarrow 0$
$P^* \leftarrow X_0, \text{PF}^* \leftarrow \varnothing$
//真实函数评估//
用原 MOP 评估初始解集 X_0 得到初始 F-值：$F(X_0) \leftarrow \{y_0, y_1, \cdots, y_{N_{\text{ini}}}\} \mid y_i = f(x_i), i = \{1, 2, \cdots, N_{\text{ini}}\}$
更新理想点 z^* 和最差点 z^{nad}
$\text{PF}^* \leftarrow F(X_0)$
$\text{Eval} \leftarrow \text{Eval} + N_{\text{ini}}$
while Eval＜MaxEvals do
　　块划分得到待优化块 Cor_t^d 和非优化块 Cor_t^{D-d}：$\{\text{Cor}_t^d, \text{Cor}_t^{D-d}\} \leftarrow \text{BlockPartition}^{D-D}$)
　　//目标函数聚合//
　　用算法 5.2 对目标值 $F(X_0)$ 进行聚合获得标量代价值 $S_{\lambda_t}^d$
　　//高斯模型//
　　构建用于建立高斯模型的数据集：$D_t \leftarrow \{X_t, S_{\lambda_t}^d\}$
　　在 D_t 上构建高斯模型：$S_{\lambda_t}^d \sim \mathcal{GP}(\boldsymbol{\mu}_t(x^d), \boldsymbol{\sigma}_t^2(x^d))$
　　构建ϵ-贪心获取函数 $a_t(x)$
　　//候选解推荐//
　　优化获取函数得到下一个候选解：$x_{t+1}^d \leftarrow \text{argmax}_{x^d}$
　　上下文向量生成：$x_{t+1}^{D-d} \leftarrow \text{GenerateContextVector}(D-d)$
　　$x_{t+1} \leftarrow [x_{t+1}^d, x_{t+1}^{D-d}]$
　　//候选解评估及更新//
　　用原 MOP 对 x_{t+1} 进行评估获得其 F-值：$y_{t+1} \leftarrow F(x_{t+1})$
　　$\text{Eval} \leftarrow \text{Eval} + 1, t \leftarrow t + 1$
　　$P^* \leftarrow P^* \cup x_{t+1}, \text{PF}^* \leftarrow \text{PF}^* \cup F(x_{t+1})$
　　更新理想点 z^* 和最差点 z^{nad}
end while

5.2.2　初始化

类似于其他基于 EGO 的多目标贝叶斯优化方法，Block-MOBO 首先利用拉丁超立方采样(LHS)[309]初始化含有 N_{ini} 个个体的解集 $X_0 = \{x_1, x_2, \cdots, x_{N_{\text{ini}}}\}$，其中 $x_i = (x_i^1, x_i^2, \cdots, x_i^D)$，$i \in [1, 2, \cdots, N_{\text{ini}}]$；同时初始化帕累托最优集 $P^* = X_0$。此外，类似 MOEA/D[15] 和 NSGA-Ⅲ[16]，Block-MOBO 还初始化如式(4-1)所示的一组均匀分布的权重向量 $\boldsymbol{\Lambda} =$

$\{\lambda_1,\lambda_2,\cdots,\lambda_N\}$。对于一个含有 M 个子目标的多目标优化问题，$\boldsymbol{\lambda}_j$ 是一个 M 维向量，即

$$\boldsymbol{\lambda}_j=(\lambda_{j,1},\lambda_{j,2},\cdots,\lambda_{j,M})^{\mathrm{T}},j\in[1,2,\cdots,N] \tag{5-1}$$

其中，$\lambda_{j,k}\geqslant 0,k\in[1,2,\cdots,M]$，并且 $\sum_{k=1}^{M}\lambda_{j,k}=1$。除此，理想点 z^* 和最低点 z^{nad} 分别用 X_0 的目标值 $f_j,j\in[1,2,\cdots,M]$ 的最小值和最大值近似。在整个搜索过程中，两者的值不断被更新为截至当前目标值 $f_j,j\in[1,2,\cdots,M]$ 的最小值和最大值。

5.2.3　函数评估与目标函数聚合

为了评估已知观察点 X_t，Block-MOBO 首先根据原昂贵的多目标优化问题计算其对应的真实目标函数值，即

$$F_t(X_t)=(f_t^1(x),f_t^2(x),\cdots,f_t^M(x))^{\mathrm{T}} \tag{5-2}$$

类似于 ParEGO，在获得目标函数值后，Block-MOBO 通过如式(4-1)所示的初始化权重向量对 M 个不同的目标值进行聚合，得到对应的标量代价值 $S_{\lambda_t}^d$（算法 5.1 中步骤 11）。不同于 ParEGO，我们针对高维决策空间中的目标聚合做了两方面调整。一方面，只在 d 维而非 D 维决策空间内对目标函数进行聚合。另一方面，提出了ϵ-贪心聚合(ϵ-Greedy Scalarization)函数，其包含两个不同的聚合函数：增广切比雪夫函数和如定义 5.1 所示的 θ-支配等级。使用增广切比雪夫函数的目的是确保帕累托最优性；θ-支配等级则是为了在高维决策空间中寻找非支配解，并通过融入目标向量之间的 θ-支配关系[19]缓解边界问题。因此，将目标空间中的 θ-支配关系引入贝叶斯优化过程中，显式地平衡了利用和探索之间的关系，从而促进解收敛性和多样性的平衡关系。下文首先对 θ-支配等级进行定义，然后再介绍ϵ-贪心目标聚合的详细过程。

1. θ-支配等级

θ-支配等级将多目标优化中衡量多个目标向量之间的支配关系引入多目标贝叶斯优化中，旨在显式地平衡搜索过程中利用和探索之间的关系。其具体定义如下。

定义 5.1：θ-支配等级

设 $P_t^*=X_t$ 是 d 维决策空间中当前得到的近似帕累托最优集，则 P_t^* 中的任意一个解 x_t' 至多能被其他 $|P_t^*|-1$ 个解 θ-支配。具体而言，θ-支配等级为

$$G(x_t,P_t^*)=1-\frac{|\{x_t'|\ x_t'<_\theta x_t\ \wedge\ x_t\neq x_t',\forall x_t,x_t'\in P_t^*\}|}{|P_t^*|-1} \tag{5-3}$$

其中，θ 是定义 1.4 中的 θ-支配关系。

根据上述定义，对于 P_t^* 中的解 x_t，当 P_t^* 中没有其他解 θ-支配 x_t 时，其具有最大等级值 $G(x_t,P_t^*)=1$。相反，当其被 P_t^* 中的所有其他解 θ-支配时，$G(x_t,P_t^*)=0$。如此，θ-支配等级保留了目标空间[20]中的 θ-支配关系。换言之，Block-MOBO 将多个目标

值之间的支配关系融入多目标贝叶斯优化方法中。θ-支配等级是在归一化目标空间 $\hat{F}(x)=(\hat{F}_1(x),\hat{F}_2(x),\cdots,\hat{F}_M(x))$ 中计算的(算法 5.2 中步骤 1),即

$$\hat{F}_i(x)=\frac{f_i(x)-z_i^*}{z_i^{\text{nad}}-z_i^*},i\in[1,M] \tag{5-4}$$

其中,z^* 和 z^{nad} 分别是理想点和最差点。

如图 5.1 所示,Block-MOBO 在计算 θ-支配等级标量代价 S_{λ_t} 时,用初始化的 N 个 M 维均匀分布的权重向量 $\boldsymbol{\lambda}$,根据式(1-3)中的距离 $d_{j,2}(x)$,将当前帕累托最优集 P_t^* 划分为 N 个簇 $\boldsymbol{C}=\{C_1,C_2,\cdots,C_N\}$。每个簇中都有与之相联系的一个或多个解。图 5.1 中,距离 $d_{j,1}(x)$ 和 $d_{j,2}(x)$ 分别如式(1-2)和式(1-3)。

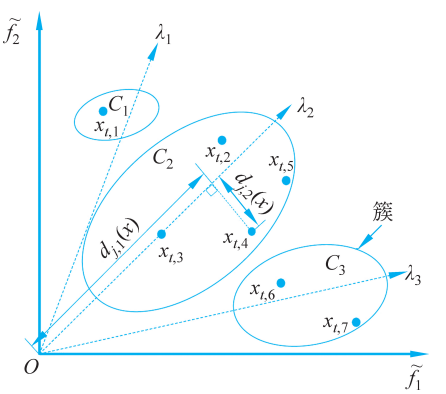


图 5.1 θ-支配等级中的聚类

对于 P_t^* 中的每个解 x_t,其 $G(x_t,P_t^*)$ 的计算仅限于当前簇内,而非整个帕累托最优集 P_t^* 中。具体而言,如果一个解 $x_t\in P_t^*$ 与第 i 个簇 $C_i,i\in[1,N]$ 相关联,则 x_t 只与 C_i 中的解而非 P_t^* 中的所有解进行比较,然后确定它是否被当前簇内的其他解 θ-支配或者 θ-支配其他解。例如,在图 5.1 中,$x_{t,2}$ 只与簇 C_2 中的 $x_{t,3}$、$x_{t,4}$ 和 $x_{t,5}$ 比较,而不会与其他解进行比较。

2. ϵ-贪心聚合

ϵ-贪心聚合借鉴了昂贵的多目标优化问题中基于代理模型方法关于贪心利用的研究[306-308],利用上述提出的 θ-支配等级对原 M 维目标值进行聚合,以达到促进收敛性以及更好地平衡收敛性和多样性的效果。ϵ-贪心聚合过程如算法 5.2 所示。

算法 5.2 ϵ-贪心聚合

输入:M 维目标函数值 $F_t(x)$;聚合概率 ϵ
输出:标量聚合值 S_{λ_t}

1: $\hat{\boldsymbol{F}}(x)\leftarrow$NormalizeObjectiveSpace($M$)
2: **if** rand()$\geqslant\epsilon$ **then**
3: $S_{\lambda_t}\leftarrow$Tchebycheff Function($\hat{f}_i(x)$)
4: **else**
5: $d_{j,1}(x_t)\leftarrow\dfrac{\|\hat{\boldsymbol{F}}(x)^{\mathrm{T}}\lambda_j\|}{\|\lambda_j\|}$

6：　$d_{j,2}(x_t) \leftarrow \left\| \hat{\boldsymbol{F}}(x)^{\mathrm{T}} - d_{j,1}(x_t)\dfrac{\lambda_j}{\|\lambda_j\|} \right\|$

7：　$\{C_1, C_2, \cdots, C_N\} \leftarrow \mathrm{Cluster}(\boldsymbol{\Lambda}, P_t^*)$

8：　$\{F_1', F_2', \cdots\} \leftarrow \theta\text{-NondominatedSort}(P_t^*, \{C_1, C_2, \cdots, C_N\})$
　　//θ-Dominance Rank//

9：　$S_{\lambda_t} \leftarrow \theta\text{-Dominance-Rank}(\{F_1', F_2', \cdots\})$

10：**end if**

具体而言，首先利用式(5-4)对目标函数值进行归一化，然后随机生成一个概率值 rand()。当 rand()$\geqslant\epsilon$时，Block-MOBO 中的标量代价值由式(4-2)中的增广切比雪夫函数计算得到；反之，首先根据 θ-支配关系的定义，计算距离 $d_{j,1}(x)$ 和 $d_{j,2}(x)$，并根据 $d_{j,2}(x)$ 和初始化的权重向量对当前近似帕累托最优集进行聚类，得到多个不同的簇。接下来，在每个簇内，对其中所有解进行 θ-非支配排序(θ-Non-dominated Sorting)得到不同等级的帕累托前沿$\{F_1', F_2', \cdots\}$。最后，根据分级的帕累托前沿计算式(5-2)中的 θ-支配等级，并将其作为标量代价值用于构建高斯代理模型。

5.2.4　块坐标更新

块坐标下降(Block Coordinate Descent，BCD)方法的有效性已在机器学习中大规模甚至超大规模的优化问题上得到验证[324-326]。其主要思想是在每次算法迭代中，BCD 方法将决策空间变量划分为不同的块，每次只最小化一个坐标块的函数值，同时保持其他块中的变量不变。对于严格凸(或拟凸、半变量或伪凸)的可微问题和不可微、可分问题，BCD 方法的全局收敛性和局部收敛性已经得到了很好的证明。受该类方法启发，将 BCD 的思想自然地借鉴到高维多目标贝叶斯优化方法中，用于求解昂贵的多目标优化问题，从而提出 Block-MOBO。块坐标更新与普通 BCD 有诸多不同。如算法 5.3 所示，块坐标更新主要包括 4 方面：块划分(Block Partition)、候选解推荐、上下文向量生成(Context Vector Generation)和块重组(如算法 5.1 中步骤 10、15～17 所示)。接下来，将对这 4 个步骤进行详细介绍。因为获取函数部分相对复杂，所以本部分相关内容将在 5.2.5 节中单独介绍。

算法 5.3　块坐标更新

1：**while** Eval<MaxEvals **do**

2：　$/\boldsymbol{d}_t \leftarrow \{id_t^1, id_t^2, \cdots, id_t^d\}, id_t^k \in [1, D], k \in [1, d]$
　　//块划分//

3：　$\{x_{t+1}^d, x_{t+1}^{D-d}\} \leftarrow \mathrm{BlockPartition}(x_t^D)$
　　//候选解推荐//

4： $x_{t+1}^{d} \leftarrow \arg\max_{x^d} a_t(x)$
//上下文向量生成//
5： $x_{t+1}^{D-d} \leftarrow$ ContextVectorGeneration$(D-d)$
//块重组//
6： $x_{t+1}^{D} \leftarrow x_{t+1}^{d} \cup x_{t+1}^{D-d}$
7： $t \leftarrow t+1$
8： **end while**

1. 块划分

在每次算法迭代中，Block-MOBO 首先将 D 维决策空间 x_D 随机划分为 x_d 和 x_{D-d} 两个不相交的块，即 $x_D = x_d \cup x_{D-d}$ 且 $x_d \cap x_{D-d} = \varnothing$，其中 $1 < d \ll D \circ x_d$ 和 x_{D-d} 两个块随算法迭代而发生变化，即 x_t^d 和 x_{t+1}^d 在不同的算法迭代过程中包含不同的决策变量维度。在每次迭代中，Block-MOBO 只考虑优化块 x_d。换言之，Block-MOBO 每次迭代只需要求解 d 维多目标优化问题，即

$$\min \boldsymbol{F}_d(x) = (f_1(x), f_2(x), \cdots, f_M(x))^{\mathrm{T}}, x \in R^d \tag{5-5}$$

且不同算法迭代中的 d 维多目标问题是不同的。此处不考虑 $d=1$ 的情况，主要是出于两方面考虑。首先，我们只关注 $D \geqslant 10$ 维的昂贵的多目标问题。在此类问题中，如果将所有 D 维决策变量逐个维度优化，则在有限的函数评估次数范围内，可能会出现只能优化 D 个决策维度中部分维度的情况，并不能保证所有维度都可以被优化。其次，由于多目标优化问题的可分性及决策空间与目标空间之间的复杂映射关系，每次只优化一个维度可能会影响 M 个子目标之间的平衡关系。

2. 上下文向量生成

如何在每次迭代中只考虑块 x^d 的情况下保证收敛性是一个关键点。在本书中，用嵌入帕累托最优知识的上下文向量赋值给包含 $D-d$ 个决策维度的、未被考虑的块，以促进收敛。具体而言，Block-MOBO 从当前帕累托最优集 P^* 中以概率 $1-p$ 复制 x^* 相应的 $D-d$ 维决策变量值作为上下文向量，即

$$x^* = \operatorname{argmin} F(x), x_{t+1}^{D-d} = [x^*]^{D-d} \tag{5-6}$$

而以概率 p 将上下文决策向量取值为决策变量取值范围内的随机值，即 $x_{t+1}^{D-d} = \mathrm{uniform}(x_{D-d})$。如果 P^* 中有多个解，选择具有最小标量代价值 $S_{\lambda_t}^{D-d}$ 的解用于赋值（算法 5.1 步骤 11）。

3. 块重组

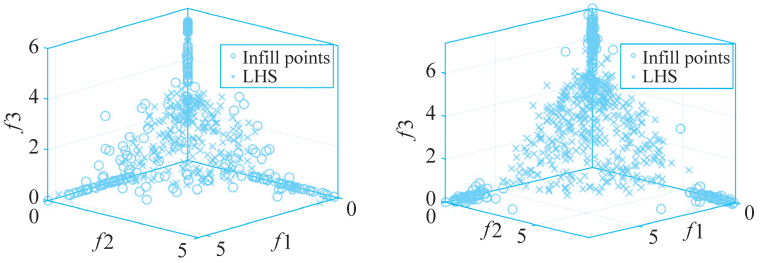
因为当前优化过程只聚焦于 d 维决策子空间，所以获取函数 $a_t(x)$ 也是在 d 维空间中优化的。通过优化 $a_t(x)$，可以得到下一个 d 维候选解。为了在原 D 维决策空间中对

候选解进行真实函数评估，将当前 d 维候选解和上下文向量生成的 $D-d$ 维解进行重组，得到原决策空间中的完整候选解 x_{t+1}^{D} 用于真实函数评估，即 $x_{t+1}^{D} \leftarrow x_{t+1}^{d} \cup x_{t+1}^{D-d}$。

通过块坐标更新，Block-MOBO 因为每次只需考虑 D 维决策空间中的 d 维子决策空间，既缓解了高维决策空间带来的维度灾难问题，又进一步提高了获取函数的优化效率。

5.2.5　ϵ-贪心获取函数

在贝叶斯优化方法中，平衡利用和探索之间关系的获取函数旨在充分利用已知和未知信息，进行全局最优搜索。然而，当前大多数多目标贝叶斯优化方法中的获取函数忽略了多目标优化中目标向量之间的帕累托支配关系，导致通过优化获取函数得到的候选解经常被已评估解支配。这加剧了多目标贝叶斯优化方法在高维决策空间中的边界问题[29]，又称过度搜索[98]，即将太多评估代价花费在搜索空间的边界附近。随着决策空间维度 D 的增加，这种情况将变得更加严重。图 5.2(a)和图 5.2(b)分别展示了 ParEGO 在 $D=20$ 和 $D=40$ 维 DTLZ2 问题上的边界问题。其中绿色点表示由 LHS 采样得到的初始解(图中 LHS)，橙色点代表 ParEGO 进行 MaxEvals=300 次函数评估、最大化 EI 得到的候选解(图中 Infill points)。从图 5.2 中可以看出，大多数非支配解由 LHS 初始化得到，而由获取函数获得的候选解大多位于搜索边界上，被初始解支配；当 $D=40$ 时，几乎所有的候选解都在边界上。文献[306-308]表明，虽然静态平衡利用和探索之间关系的解可能是最优解，但通过任何固定的利用-探索权重选择的解并不能保证收敛性。为了强调该问题，我们提出了ϵ-贪心(ϵ-greedy)获取函数(算法 5.1 中步骤 14)，其将多目标优化目标空间中的帕累托支配关系引入优化过程，以进一步促进候选解收敛性和多样性之间的平衡。



(a) 20维DTLZ2上的初始解和候选解　　(b) 40维DTLZ2上的初始解和候选解

图 5.2　ParEGO 中的边界问题示例(见彩插)

ϵ-贪心获取函数的定义基于上节提出的ϵ-贪心聚合函数，具体定义如式(5-7)所示。

$$a_t^d(x)=\begin{cases}E(\max(S_{\lambda_t}-S_{\max}),0) & p<\epsilon\\ E(\max(S_{\min}-S_{\lambda_t}),0) & p\geqslant\epsilon\end{cases} \tag{5-7}$$

其中，$S_{\max}$和$S_{\min}$分别是当前θ-支配等级和增广切比雪夫值的最优值。具体而言，ϵ-贪心获取函数定义如下。

如果$p<\epsilon$，那么ϵ-贪心获取函数为

$$a_t^d(x)=(\hat{S}_{\lambda_t}-S_{\max})\Phi\left(\frac{\hat{S}_{\lambda_t}-S_{\max}}{s}\right)+s\phi\left(\frac{\hat{S}_{\lambda_t}-S_{\max}}{s}\right) \tag{5-8}$$

如果$p\geqslant\epsilon$，那么ϵ-贪心获取函数为

$$a_t^d(x)=(S_{\min}-\hat{S}_{\lambda_t})\Phi\left(\frac{S_{\min}-\hat{S}_{\lambda_t}}{s}\right)+s\phi\left(\frac{S_{\min}-\hat{S}_{\lambda_t}}{s}\right) \tag{5-9}$$

其中，$\hat{S}_{\lambda_t}$是DACE预测器，s是用于衡量不确定性的均方根误差(Root Mean Squared Error，RMSE)[37]，p是初始概率值$p=\text{rand}()$，$\Phi(\cdot)$和$\phi(\cdot)$分别是正态分布的分布函数和概率密度函数。

ϵ-贪心获取函数随标量代价$\hat{S}_{\lambda_t}$单调递减，随不确定性s单调递增。通过优化该获取函数得到的候选解，要么能够很好地平衡收敛性和多样性之间的关系，要么以概率ϵ保持多目标优化目标空间中的帕累托支配关系，使得候选解尽可能地不被已评估解支配，从而缓解高维决策空间中的边界问题。

5.2.6 高斯模型及候选解推荐

算法5.1中高斯代理模型(步骤13)和获取函数的优化(步骤15)是在d维决策子空间进行的。在得到标量代价值后，可以得到用于建立高斯模型的已知观察点集，即

$$D_{1:t}=\left\{x_t,y_t\mid y_t=\hat{S}_{\lambda_t},t=\{1,2,\cdots,T\},T=\left\lfloor\frac{\text{MaxEvals}}{B}\right\rfloor\right\} \tag{5-10}$$

其中，$x_t=\{x_t^1,x_t^2,\cdots,x_t^{N^{\text{res}}}\}$，$\hat{S}_{\lambda_t}$是5.2.3～5.2.5节得到的标量代价值。建立在标量成本上的高斯模型为

$$S_{\lambda_t}^d\sim\mathcal{GP}(\mu_t(x),\sigma_t^2(x)) \tag{5-11}$$

其中，高斯代理模型的均值函数和方差函数分别为$\mu_t(x)$和$\sigma_t^2(x)$。因为高斯模型是建立在d维而非D维决策空间，所以降低了决策空间维度。给定当前高斯代理模型，新的d维候选解x_{t+1}^d的标量代价值的预测值分布为

$$P(x_{t+1}^d\mid(x_{1:t}^d,y_{1:t}^d))\sim\mathcal{GP}(\mu_{t+1}(x),\sigma_{t+1}^2(x)) \tag{5-12}$$

接下来，在d维子空间上建立ϵ-贪心获取函数$a_t^d(x)$，并对其进行优化获得下一个候选解，即

$$x_{t+1}^d=\text{argmax}\ a_t^d(x) \tag{5-13}$$

得到 d 维子空间上的候选解 x_{t+1}^{d} 后，采用块重组（算法 5.1 步骤 17）得到 D 维决策空间中的下一个候选解，即

$$x_{t+1} = x_{t+1}^{d} \cup x_{t+1}^{D-d} \tag{5-14}$$

最后，对新的候选解 x_{t+1} 用原昂贵的多目标优化问题进行评估得到其对应的 F-值，同时更新理想点 z^{*}、最差点 z^{nad}、当前帕累托最优解 P^{*} 和帕累托最优前沿 PF^{*} 等（算法 5.1 步骤 19～21），用于下一个算法迭代中高斯代理模型的构建。

5.3 实　　验

本节主要介绍用于验证 Block-MOBO 有效性的实验设置和结果与分析。具体而言，5.3.1 节介绍具体的实验设置；5.3.2 节介绍所有相关算法在标准多目标测试集上的实验结果和分析；5.3.3 节和 5.3.4 节分析块坐标更新和ϵ-贪心获取函数对 Block-MOBO 优化性能的影响；5.3.5 节和 5.3.6 节分别分析块大小 d 和上下文向量对 Block-MOBO 优化性能的影响。

5.3.1　实验设置

为了验证 Block-MOBO 的有效性，将其与其他 9 种基线方法在 3 个标准多目标测试问题上进行了对比分析实验。

1. 基线方法

9 种基线方法包括随机搜索（Random Search，RS）、多目标进化算法 NSGA-Ⅱ[12] 和 SMS-EMOA[327]，以及多目标贝叶斯优化方法 ParEGO[38]、MOEA/D-EGO[40]、ReMO[97]、Multi-LineBO、K-RVEA[50] 和 MOEA/D-ASS[44]。其中，Multi-LineBO 是单目标贝叶斯优化方法 LineBO[80] 的变体方法。

2. 测试问题

3 个标准多目标测试问题包括 DTLZ[159]、WFG[36] 和 mDTLZ[168]。所有问题都采用 3 个优化目标。

3. 评价指标

为了对比所有相关方法，本章采用了 2.6 节中介绍的反世代距离（IGD）[30]、世代距离（GD）[171]、豪斯多夫距离（Δ_p）[174]、超体积（HV）和 CPU 计算时间等评价指标对优化结果进行评价与分析，从而衡量相关算法的优化性能。

4. 实现细节

所有相关方法的初始解个数为 $N_{\text{ini}} = 11D - 1$。每个算法在每个测试实例上独立运