

第 5 章

基于分解的多目标 分布估计算法

5.1 概 述

在现实生活中,很多的问题都具有多个优化目标。一般地,这些目标是相互冲突的。也就是说,试图提高一个目标会导致其他目标质量的下降。对于多目标优化问题,我们希望获得尽可能多的非支配的解,以供决策者根据实际情况选择最终的方案。演化算法是基于种群的元启发式算法,这一特性使得演化算法适用于解决多目标优化问题。近年来,演化多目标算法得到了广泛重视,研究人员设计了很多适用于多目标问题的演化算法,这些算法已经解决很多实际问题^[45-47]。本章主要介绍一种改进的多目标分布估计算法,称为 s -MEDA/D (scale Adaptive Decomposition based Multi-objective Estimation of Distribution Algorithm, 基于问题规模自适应分解的多目标分布估计算法)。MEDA/D 在经典多目标演化算法——基于分解的多目标演化算法(MOEA/D)的基础上,利用概率向量对每个分解后的子问题建模。本章将利用遗传算法范式的 MOEA/D 改进为利用分布估计范式的 MEDA/D。在此基础上,本章基于概率向量提出了规模自适应的生成算子。最后,本章通过在多目标 0-1 背包问题上的实验说明算法的有效性。

本章的结构安排如下: 5.2 节介绍 MOEA/D, 首先介绍权重分解方法和切比雪夫分解方法, 随后完整地描述 MOEA/D 框架和 MEDA/D; 5.3 节详细介绍规模自适应生成算子, 并证明该算子保持多样性的能力; 5.4 节介绍多目标的 0-1 背包问题和 s -MEDA/D 在 9 个实例上的实验结果; 5.5 节总结本章内容。

5.2 基于分解的多目标演化算法框架

5.2.1 分解方法

1. 权重分解方法

通过对每个目标赋予不同权重, 可以将多目标问题转化为多个单目标问题。目标函



数的权重和定义如下:

$$\begin{aligned} \max g^w(\mathbf{x} \mid \boldsymbol{\lambda}) &= \sum_{i=1}^m \lambda_i f_i(\mathbf{x}) \\ \text{subject to } \mathbf{x} &\in \Omega \end{aligned} \quad (5-1)$$

其中, $\boldsymbol{\lambda} = [\lambda_1 \lambda_2 \cdots \lambda_m]^T$ 是权重向量。通常, 权重向量 $\boldsymbol{\lambda}$ 应当满足归一化条件和正定条件。也就是说, 对于所有 $i \in \{1, 2, \dots, m\}$, $\sum_{1 \leq i \leq m} \lambda_i = 1$ 并且 $\lambda_i \geq 0$ 。在满足这个条件的前提下, 理论已经证明: 如果原问题是凸的, 所有帕累托最优解都可以通过加权分解方法得到^[48]; 如果原问题是非凸的, 上述结论并不成立。一个权重向量对应于一个子问题。在 MOEA/D 中, 在算法的初始化阶段就生成算法所需要的所有权重向量, 这些向量在整个计算过程中保持不变。

2. 切比雪夫分解方法

为了介绍切比雪夫分解方法, 首先给出理想目标向量(ideal objective vector)和空想目标向量(utopian objective vector)的定义^[49]。对于理想目标向量, 每个分量的值为对应目标函数所有可能取值的最大值, 即对每一个 $i \in \{1, 2, \dots, m\}$ 都有 $z_i^* = \sup_{\mathbf{x} \in \Omega} f_i(\mathbf{x})$ 。空想目标向量定义如下, 对于所有 $i \in \{1, 2, \dots, m\}$, $z_i^{**} = z_i^* + \epsilon_i$ 并且 $\epsilon_i > 0$ 。这两个向量都可以作为参考向量使用^[49]。一个多目标问题的切比雪夫分解方法遵循如下形式:

$$\begin{aligned} \min g^t(\mathbf{x} \mid \boldsymbol{\lambda}, \mathbf{z}^*) &= \max\{\lambda_i \mid f_i(\mathbf{x}) - z_i^*\} \\ \text{subject to } \mathbf{x} &\in \Omega \end{aligned} \quad (5-2)$$

其中, $\boldsymbol{\lambda} = [\lambda_1 \lambda_2 \cdots \lambda_m]^T$ 是权重向量。理论已证明, 当参考点是空想目标向量时, 对于任意一个帕累托最优解 $\mathbf{x}^* \in \Omega$, 都存在一个向量 $\mathbf{0} < \mathbf{w} \in \mathbb{R}^m$ 使得 $\mathbf{x}^*$ 是这个加权切比雪夫问题的解^[48]。一般地, 理想目标向量是未知的。因此, 在 MOEA/D 中, 参考点在每轮迭代中都需要更新。参考点的每一分量都被设定为当前所有目标向量对应分量的最大值。

5.2.2 MOEA/D 框架

MOEA/D^[50] 框架如算法 5-1 所示。该算法的具体过程描述如下。第 1~5 行是初始化部分。MOEA/D 也采用精英策略, 在算法的整个计算过程中, 维护一个外部种群用于存储算法发现的所有非支配解。这个外部种群(External Population, EP)最终作为算法的输出结果。因此算法的第 1 行初始化外部种群。因为在计算的开始时还没有任何解, 外部种群被初始化为空集。作为基于分解的多目标演化算法, MOEA/D 需要采用某种分解方法将原问题分解为多个单目标的问题。无论使用权重分解方法还是使用切比雪夫分解方法都需要一组权重向量。在第 3 行, 算法初始化一组权重向量。令 K 表示权重向





量的数量,并且这组权重向量记为 $\Delta = \{\lambda_1, \lambda_2, \dots, \lambda_K\}$ 。在第3行,对每个权重向量,找出和它最邻近的 T 个向量,参数 T 称作邻域大小。例如,对于 λ_i ,将和它最邻近的 T 个向量的下标保存在 $B(i)$ 中。若 λ_j 是 λ_i 的最邻近向量,那么 $j \in B(i)$ 。然后,用随机的或基于问题的方法生成一个初始种群,并对该种群中的每个个体计算其适应度。如果算法使用权重方法分解问题,并不需要参考点,因此不需要第5行。如果算法使用切比雪夫分解方法,需要根据目前得到的目标向量重新计算参考点。第6~13行是算法的主循环。在每一轮迭代中,MOEA/D 按顺序处理所有子问题。在一个子问题中,算法利用其邻域中的解生成一个新解 y 。MOEA/D 使用遗传算子生成新解。在本章提出算法中,新解由规模自适应生成算子生成,这个算子将在5.3节中阐述。生成的解需要利用与问题相关的方法提高质量。若生成的新解并不在可行域内,也需要使用与问题相关的方法修复这个解。和第5步一样,如果涉及参考点,需要在第10步更新参考点。因为每个新生成的解来自一个子问题的邻域,利用了这个邻域中其他子问题的相关信息,所以,在生成一个新解后,应该更新邻域中的子问题的解。用 $g(x)$ 表示式(5-1)中的 $g^*(x|\lambda)$ 或者式(5-2)中的 $g^t(x|\lambda, z^*)$ 。若 $g(y) \geq g(x_j)$, 那么令 $x_j = y$ 。第12行更新外部种群。这一步包含两个子步骤:首先删除所有被 $F(y)$ 支配的解;然后将 $F(y)$ 加入到外部种群中。

算法 5-1 MOEA/D 框架

- 1: 初始化外部种群为空集
- 2: 初始化一组权重向量
- 3: 对每一个权重向量,找出和它最邻近的权重向量
- 4: 生成初始种群并且计算种群中个体的适应度
- 5: 根据需要,初始化参考点 z
- 6: **while** 未达到终止条件 **do**
- 7: **for** $i=1, 2, \dots, T$ **do**
- 8: 对第 i 个子问题,生成一个新解 x
- 9: 利用问题相关的方法提高 x 的质量
- 10: 根据需要,更新参考点
- 11: 更新邻域中子问题的解
- 12: 更新外部种群
- 13: **end for**
- 14: **end while**

5.3 规模自适应生成算子

本章设计的规模自适应生成算子适用于0-1编码问题。这个算子使用了概率向量作





为概率模型。对于第 i 个子问题,对应的概率向量记为

$$\mathbf{p}^i = [p_1^i \ p_2^i \ \cdots \ p_n^i] \quad (5-3)$$

其中, n 为问题的维度。第 i 个子问题的相邻向量的下标记录在集合 $B(i) = \{i_1, i_2, \dots, i_r\}$ 中。分量 p_j^i 定义如下:

$$p_j^i = \frac{\sum_{l=1}^T x_{j^{i_l}}^i + \xi}{T + 2\xi} \quad (5-4)$$

其中, T 表示邻域大小。令 $x_{j^{i_l}}^i$ 表示对应第 i_l 个子问题当前解的第 j 个分量。式(5-4)中 ξ 表示附加到概率向量上的一个较小的量。这个附加量用于提高多样性,其具体定义如下:

$$\xi = \frac{Ts}{n - 2s} \quad (5-5)$$

其中,参数 s 是规模自适应生成算子的参数。通过采样 $\mathbf{p}^i$,可以得到一个新解 $\mathbf{y}^i = [y_1 \ y_2 \ \cdots \ y_n]^\top$ 。

$$y_i = \begin{cases} 1, & \text{rand}(0,1) < p_j^i \\ 0, & \text{其他} \end{cases} \quad (5-6)$$

其中, $\text{rand}(0,1)$ 表示一个能生成在 $0 \sim 1$ 均匀分布的随机函数。

规模自适应生成算子如算法 5-2 所示。

算法 5-2 规模自适应生成算子

```

1: procedure Reproduce( $\mathbf{X}, B(i), \mathbf{y}^{\text{new}}$ )
2:    $\mathbf{P} \leftarrow 0$ 
3:   for each  $l \in B(i)$  do
4:     for  $j = 1, 2, \dots, n$  do
5:       if  $x_j^{i_l} = 1$  then
6:          $p_j \leftarrow p_j + 1$ 
7:       end if
8:     end for
9:   end for
10:   $\xi \leftarrow (Ts)/(n - 2\xi)$ 
11:  for  $j = 1, 2, \dots, n$  do
12:     $p_j \leftarrow (p_j + \xi)/(T + 2\xi)$ 
13:    if  $\text{rand}(0,1) < p_j$  then
14:       $y_j^{\text{new}} \leftarrow 1$ 
15:    else
16:       $y_j^{\text{new}} \leftarrow 0$ 
17:    end if
18:  end for
19: end procedure

```





式(5-5)中的参数 s 决定了从 $\mathbf{p}^i$ 新生成的解中最小变异个数的数学期望。对于一个子问题,即使其邻域中所有的解都相同,通过规模自适应生成算子生成的新解也会有一些分量和其父代不同,变异个数的数学期望为 s 。证明如下。

证明: 假设对于所有 $l \in B(i)$, $\mathbf{x}^l = \mathbf{x}^* = [x_1^* \ x_2^* \ \cdots \ x_n^*]^T$, 即邻域中所有的解都相同。将式(5-5)代入式(5-4)可得

$$p_j = \frac{(n-2s) \sum_{l=1}^T x_j^{i_l} + Ts}{(n-2s)T + 2Ts} = \frac{(n-2s)Tx_j^* + Ts}{(n-2s)T + 2Ts} = \frac{(n-2s)x_j^* + s}{n} \quad (5-7)$$

如果 $x_j^* = 0$, 那么 $p_j = s/n$, 结合式(5-6)可以得到

$$P(y_j = 1 \mid x_j^* = 0) = \frac{s}{n} \quad (5-8)$$

如果 $x_j^* = 1$, 那么 $p_j = (n-s)/n$, 结合式(5-6)可以得到

$$P(y_j = 0 \mid x_j^* = 1) = 1 - \frac{n-s}{n} = \frac{s}{n} \quad (5-9)$$

因此,对于解中的一个分量,采样得到的结果和原始值不同的概率是 s/n 。概率向量假设各变量是相互独立的。令 N 表示变化比特的个数, N 是一个服从参数为 n 和 s/n 的二项分布的随机变量,即 $N \sim B(n, s/n)$, 那么 N 的数学期望为 s 。

因此,一个新解与其父代不同的概率为

$$P_{\text{change}} = 1 - \left(1 - \frac{s}{n}\right)^n \quad (5-10)$$

当 $n \gg s$ 时(由于问题的规模通常远大于参数 s , 因此该条件在一般情况下均成立), P_{change} 基本不受参数 n 的影响。证明如下。

证明: 令 $\epsilon = s/n$, ϵ 接近 0。将式(5-10)改写为

$$P_{\text{change}} = 1 - \left(1 - \frac{s}{n}\right)^n = 1 - (1 - \epsilon)^{s/\epsilon} \quad (5-11)$$

在 0 点附近,对 ϵ 计算泰勒级数可得

$$P_{\text{change}}(\epsilon) = 1 - (1 - \epsilon)^{s/\epsilon} = 1 - e^{-s} + o(\epsilon^2) \approx 1 - e^{-s} \quad (5-12)$$

其中, $o(\epsilon^2)$ 是二阶小量。因此, P_{change} 基本不随 n 的变化而变化。

如果直接使用 ξ 作为规模自适应生成算子的参数,也可以达到提高计算中种群多样性的目的。但是,这个参数的作用会随着问题的规模而变化,而且它没有直观、明确的物理意义。作为一种防止概率向量收敛的方法,其实质和变异算子相似。在文献[20]中对 PBIL 的实验说明,将概率向量和变异算子结合可以提高算法的性能。但是变异算子是一个基于遗传算法范式设计的算子,按照分布估计范式,本章将遗传算子集成到概率模型中。当 s 设定为 0 时,式(5-5)中的 $\xi = 0$, 此时规模自适应生成算子和 MOED/A 中使用





的一样^[51]。

5.4 测试与实验

5.4.1 测试问题

单目标 0-1 背包问题作为经典的组合优化问题已经获得广泛的关注和研究^[33]。单目标 0-1 背包问题可以扩展为任意多目标问题。多目标 0-1 背包问题描述如下：给定 n 个物品和 m 个背包。每个物品都有重量和价值两个属性,同时每个背包具有一定的容量。如果选择了一个物品,需要将其放入一个背包中。算法的目标是在保证每个背包中的物品的总重量小于其容量的条件下找到一个可以最大化所有物品总价值的方案。

多目标 0-1 背包问题的形式化定义如下：

$$\begin{aligned} \max \mathbf{F}(\mathbf{x}) &= [f_1(\mathbf{x}) \ f_2(\mathbf{x}) \ \cdots \ f_m(\mathbf{x})]^T \\ f_i(\mathbf{x}) &= \sum_{j=1}^n p_{ij} x_j, i=1,2,\cdots,m \\ \text{subject to } \sum_{j=1}^n w_{ij} x_j &\leq c_i, i=1,2,\cdots,m \end{aligned} \quad (5-13)$$

其中, $p_{i,j}$ 和 $w_{i,j}$ 分别为第 i 个背包中的第 j 个物品的价值和重量, c_i 是第 i 个背包的容量。对所有的 $j \in (1,2,\cdots,n)$, $\mathbf{x}=[x_1 \ x_2 \ \cdots \ x_n]^T$, $x_j \in \{0,1\}$ 即为问题的一个解。和单目标 0-1 背包的编码方案一样, $x_j=1$ 表示选取了第 j 个物品。如果在算法运算的过程中产生的解不是可行解,那么需要修复算子修复这个解。在本章中,算法使用与 MOGLS^[52] 和 MOEA/D^[50] 中相同的修复算子,如算法 5-3 所示。

算法 5-3 多目标 0-1 背包问题修复算子

```

1: procedure repairOperator( $\mathbf{x}$ )
2:   while  $\mathbf{x}$  不满足条件 do
3:      $J = \{j \mid 1 \leq j \leq n, x_j = 1\}$ 
4:      $I = \left\{ i \mid 1 \leq i \leq m, \sum_{j=1}^n w_{ij} x_j > c_i \right\}$ 
5:      $k = \arg \min_{j \in J} g(\mathbf{x}) - g(\mathbf{x}^{j-}) / \sum_{i \in I} w_{ij}$ 
6:      $x_k = 0$ 
7:   end while
8: end procedure

```

其中, $g(\mathbf{x})$ 是目标函数, $\mathbf{x}^{j-}$ 是 $\mathbf{x}$ 将 j 个分量改写为 0 后的解。这个修复算子使用贪





心算法,不断移除价值低但重量大的物品,直到满足限制条件^[50]。

为了测试算法并和 MOEA/D 比较,本章使用了多目标 0-1 背包问题的 9 个实例作为测试问题。包含 3 种不同的目标和 3 种不同问题规模的组合。每个实例记为 KN- n - m ,其中, n 表示物品的数量, m 表示目标的个数,即背包的数量。

5.4.2 参数研究

本节主要研究参数 s 的影响。在 5.3 节中已经证明,参数 s 具有明确的物理意义。 s 不应该取过大的值,否则算法将倾向于随机搜索。即使一个较小的 s 也可以显著提高解的质量。例如,当 $s=0.4$ 时,问题的规模是 $n=500$,生成的解与父代不同的概率至少为

$$P_{\text{change}} = 1 - \left(1 - \frac{0.4}{500}\right)^{500} > 32\% \quad (5-14)$$

需要指明的是,这是在极端情况下取得的结果,也就是所有邻域中的解都相同的时候。邻域中的解一般并不相同,那么由算子生成不同的解的概率更大。在探究参数 s 的影响时,本节使用 Hypervolume 作为评价指标。这是因为 Hypervolume 不需要其他算法作为参考。在本节中,测试 $s \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ 。图 5-1 为算法性能随参数 s 的变化情况。图 5-1(a)是使用权重分解方法时的结果,而图 5-1(b)是使用切比雪夫分解方法时的结果。

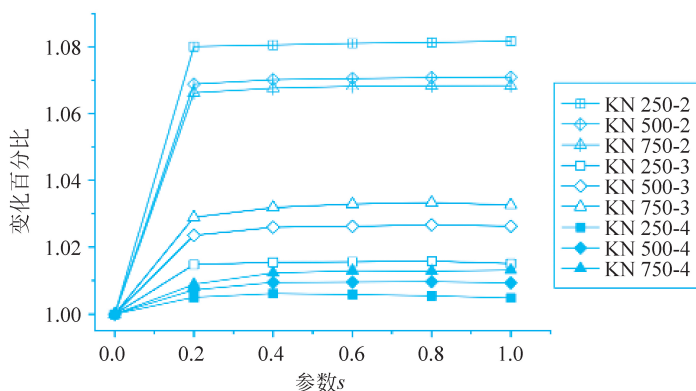
在计算 Hypervolume 时,同样选取了原点作为计算的参考点。对于不同目标个数的问题,Hypervolume 的变化范围很大($1 \times 10^7 \sim 1 \times 10^{18}$),因此要对结果进行归一化,从而使得结果可以直观地比较。将所得到的结果除以相同问题下 $s=0$ 时的结果。那么 y 轴为相对值,因此对于所有实例,结果都从 1.00 开始。从图 5-1 中可以看出,改进的算子相对于 MEDA/D 是有效的, s -MEDA/D 得到的结果要优于 MEDA/D 得到的结果。对于二目标问题,随着 s 的增加,Hypervolume 也增加;对于三目标问题,算法的性能在 $s=0.8$ 时达到最大,当 $s=1.0$ 时性能出现下降。 s -MEDA/D 对参数并不敏感,所以得到的结果波动较小。

5.4.3 实验设置与性能指标

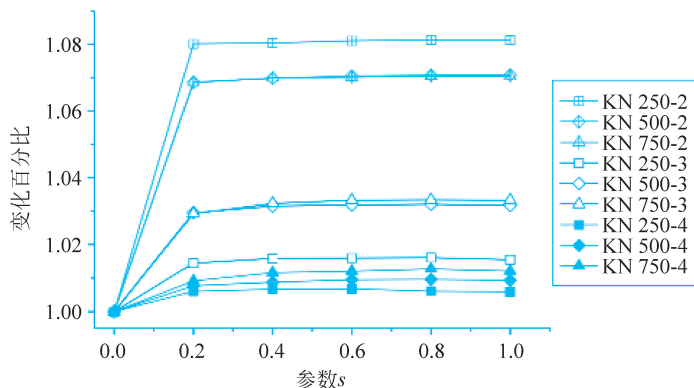
在 MOEA/D 框架中,首先需要分解多目标问题。无论使用上述的权重分解方法还是切比雪夫分解方法,都需要生成一组权重向量。权重向量的数量由参数 H 决定。对于每一个权重向量,其分量的值从以下集合中选取:

$$\left\{ \frac{0}{H}, \frac{1}{H}, \dots, \frac{H}{H} \right\} \quad (5-15)$$





(a) 权重分解



(b) 切比雪夫分解

图 5-1 算法性能随参数 s 的变化

这种方式产生的权重向量的个数为 $N = C_{H+m-1}^{m-1}$ 。为了实现公平的比较,本章算法中的权重向量的生成方式与 MOEA/D 和 MEDA/D 是相同的。具体的设置如下:对于两目标问题的 3 个不同规模的实例,权重向量的数量分别为 150、200 和 250。对于三目标问题,参数 H 设定为 25;对于四目标问题, H 设定为 12。在 MOEA/D 框架中,还有一个重要的参数 T ,即向量的邻域大小。对于所有实例,邻域大小取 $T=10$,这个设置与 MOEA/D^[50] 一样。MOEA/D 中变异算子的变异概率为 0.01。 s -MEDA/D 中参数 s 设定为 0.4。

实验中,算法在达到最大评价次数时终止。本章实验所使用的终止条件与 MOGLS^[52] 和 MOEA/D^[50] 中使用的终止条件一致。对于每一组实例都进行 30 次独立实验。本章实验参数如表 5-1 所示。给定两个包含近似帕累托最优解的集合 A 和 B ,通过计算覆盖指标 $C(A, B)$ 可以反映两组解的优劣。 $C(A, B)$ 表示被 A 中解支配的解在





B 中所占的比例,其定义如下:

$$C(A, B) = \frac{|\{u \in B \mid \exists v \in A : v \text{ dominates } u\}|}{|B|} \quad (5-16)$$

Zitzler 等人^[53] 使用解集所覆盖的区域的大小作为解集的评判标准,称为 Hypervolume^[53]。Hypervolume 最大的优势在于可以独立评价一个解集的质量。但是 Hypervolume 需要一个参考点。在本章的实验中,对于所有的实例,都使用原点作为参考点。对于不同算法的实验结果,使用显著性为 95% 的威尔科克森符号秩检验计算算法得到结果是否存在显著的差异。

表 5-1 本章实验参数

实例	N	最大评价数 ($\times 10^3$)	T	P_{mutation} (MOEA/D)	s (s -MOEA/D)
KN-250-2	150	75	10	0.1	0.4
KN-500-2	200	100			
KN-750-2	250	125			
KN-250-3	351	150			
KN-500-3	351	125			
KN-750-3	351	150			
KN-250-4	455	125			
KN-500-4	455	150			
KN-750-4	455	175			

5.4.4 实验结果

s -MEDA/D 和 MOEA/D 实验结果比较如表 5-2 所示。具有显著差别的结果用粗体表示。

从表 5-2 可以得出如下结论。对于覆盖指标, s -MEDA/D 在 9 个实例上都优于 MOEA/D。对于问题规模为 250 的 3 个实例, $C(s\text{-MEDA/D}, \text{MOEA/D})$ 在 85% 左右, 而 $C(\text{MOEA/D}, s\text{-MEDA/D})$ 小于 10%。对于问题规模为 750 的 3 个实例, $C(s\text{-MEDA/D}, \text{MOEA/D})$ 大于 99.5%, 而 $C(\text{MOEA/D}, s\text{-MEDA/D})$ 为 0。这说明在这 3 个实例上, s -MEDA/D 得到的绝大多数解都支配了 MOEA/D 得到的解, 而 MOEA/D 得到的结果中没有任何解支配了 s -MEDA/D 得到的解。对于 Hypervolume 指标, 除了 KN-250-2 以外, 其他所有的结果都表明 s -MEDA/D 显著优于 MOEA/D。



表 5-2 s -MEDA/D 与 MOEA/D 实验结果比较

问题	m	C-指标		Hypervolume 指标	
		$C(s\text{-MEDA/D}, \text{MOEA/D})$	$C(\text{MOEA/D}, s\text{-MEDA/D})$	s-MEDAD	MOEA/D
基于权重分解					
KN-250	2	0.877 179 ± 0.067 145	0.060 381 ± 0.052 836	(9.832 523 ± 0.010 875)E+07	(9.833 248 ± 0.006 205)E+07
	3	0.894 277 ± 0.032 293	0.036 038 ± 0.015 913	(9.276 245 ± 0.009 366)E+11	(9.240 465 ± 0.006 721)E+11
	4	0.841 459 ± 0.037 736	0.028 743 ± 0.010 869	(8.006 844 ± 0.009 728)E+15	(7.936 617 ± 0.006 874)E+15
KN-500	2	0.983 875 ± 0.013 797	0.000 279 ± 0.001 044	(4.061 472 ± 0.002 780)E+08	(4.053 955 ± 0.002 866)E+08
	3	0.994 918 ± 0.005 147	0.000 397 ± 0.000 909	(7.665 099 ± 0.008 639)E+12	(7.581 944 ± 0.006 951)E+12
	4	0.989 432 ± 0.004 501	0.000 226 ± 0.000 322	(1.335 180 ± 0.002 071)E+17	(1.306 883 ± 0.002 032)E+17
KN-750	2	0.996 841 ± 0.008 467	0.000 000 ± 0.000 000	(8.900 160 ± 0.006 867)E+08	(8.853 172 ± 0.007 248)E+08
	3	0.999 755 ± 0.001 035	0.000 000 ± 0.000 000	(2.693 014 ± 0.002 851)E+13	(2.649 068 ± 0.003 116)E+13
	4	0.999 262 ± 0.000 478	0.000 000 ± 0.000 000	(7.029 037 ± 0.011 975)E+17	(6.778 892 ± 0.016 631)E+17
基于切比雪夫分解					
KN-250	2	0.850 588 ± 0.084 157	0.090 291 ± 0.056 651	(9.829 682 ± 0.014 494)E+07	(9.833 269 ± 0.007 742)E+07
	3	0.890 765 ± 0.040 440	0.039 502 ± 0.020 306	(9.277 022 ± 0.009 184)E+11	(9.239 961 ± 0.006 990)E+11
	4	0.835 567 ± 0.033 432	0.031 279 ± 0.013 262	(8.005 976 ± 0.010 006)E+15	(7.936 000 ± 0.010 029)E+15
KN-500	2	0.978 685 ± 0.024 043	0.000 976 ± 0.002 797	(4.060 804 ± 0.003 585)E+08	(4.060 804 ± 0.003 585)E+08
	3	0.995 862 ± 0.004 070	0.000 273 ± 0.000 551	(7.666 164 ± 0.007 918)E+12	(7.585 415 ± 0.008 352)E+12
	4	0.988 791 ± 0.005 239	0.000 429 ± 0.001 172	(1.334 330 ± 0.002 023)E+17	(1.306 300 ± 0.001 688)E+17
KN-750	2	0.998 460 ± 0.003 947	0.000 000 ± 0.000 000	(8.902 155 ± 0.005 378)E+08	(8.853 976 ± 0.006 809)E+08
	3	0.999 687 ± 0.000 850	0.000 000 ± 0.000 000	(2.693 431 ± 0.002 328)E+13	(2.649 361 ± 0.003 369)E+13
	4	0.999 144 ± 0.000 685	0.000 000 ± 0.000 000	(7.028 353 ± 0.010 077)E+17	(6.780 390 ± 0.018 409)E+17