

bs2int 参考资料

**概要:** 二进制/格雷编码矩阵到整数值矩阵的转换。

**描述:**  
该函数把二进制/格雷码编码的种群染色体矩阵解码成十进制整数的矩阵。

**语法:** Phen = bs2int(Chrom, FieldD)

**详细说明:**  
该函数根据区域描述器（又称译码矩阵）将用二进制/格雷码编码的种群染色体矩阵 **Chrom** 解码成十进制的整数表示的种群表现型矩阵 **Phen**。

二进制/格雷码种群染色体矩阵 **Chrom** 是诸如下图所示的矩阵，矩阵的每一行代表种群中的一个个体的染色体。

$$Chrom = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

译码矩阵 **FieldD** 具有下面的结构：

$$FieldD = \begin{pmatrix} lens \\ lb \\ ub \\ codes \\ scales \\ lbin \\ ubin \\ varTypes \end{pmatrix}$$

尽管上面的结构看起来像一个列向量，**FieldD** 实际上是一个矩阵，每一列对应一个变量。换言之，需要将染色体解码成多少个变量，那么 **FieldD** 就有多少列。

可以手写代码创建比较复杂的译码矩阵 **FieldD**，也可以调用 **crtfld** 函数来自动生成。详见“**crtfld 参考资料**”。

*lens* 包含染色体的每个子染色体的长度。**sum(lens)** 等于染色体长度。  
*lb* 和*ub* 分别代表每个变量的上界和下界。  
*codes* 指明染色体子串用的是标准二进制编码还是格雷编码。*codes[i] = 0* 表示第*i*个变量使用的是标准二进制编码；*codes[i] = 1* 表示使用格雷编码。

*scales*、*lbin*、*ubin* 和*varTypes* 的含义详见“**Geatpy 数据结构**”文档。因为本函数规定解码得到的变量均为整数，解码时使用算术尺度，并且包含变量的两个边界，因此在本函数传入的 **FieldD** 里面，这四个量并无实际用途，仅仅是为了兼容其他函数而传入统一格式的 **FieldD**。对于要让 **FieldD** 不包含变量边界的场合，需要树洞调整 **FieldD** 的上下界，或者调用 **crtfld** 函数以生成符合规格的 **FieldD**。

**应用实例:**  
考虑一个二进制编码的种群染色体矩阵：

$$Chrom = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

欲让其解码后得到两个决策变量，染色体的前 2 位表示第一个决策变量，后 3 位表示第二个元素，两决策变量的范围分别是 [-4,2] 和 [-2, 7]。用 **bs2int** 函数将 **Chrom** 解码转换成整数表现型。

```
import numpy as np
import geatpy as ea
Chrom = np.array([[1,0,0,0,1],
                  [0,1,0,0,0],
                  [0,0,1,1,0]])
FieldD = np.array([[2,3],[-4,-2],[2,7],[1,1],[1,1],[1,1],[1,1],[1,1]])
Phen = ea.bs2int(Chrom, FieldD) # 进行解码
```

解码后结果如下：

$$Phen = \begin{pmatrix} 2 & -1 \\ -2 & -2 \\ -4 & 3 \end{pmatrix}$$

**解释:** 对 **Chrom** 进行解码时，**bs2int** 函数是先把二进制矩阵转换成十进制自然数矩阵，然后把结果均匀映射到变量的区间上，得到解码结果。当使用格雷码进行解码时，**bs2int** 函数先将格雷码矩阵转换成标准二进制编码矩阵，然后再按上述方式转换。

**注:** 因为采用的是均匀的区间映射的方式，因此，当编码空间比解空间大时，会出现多个不同的染色体解码后得到的值是一样的情况；另外，若编码空间比解空间小，则会出现解空间中有些值无法通过解码得到（此时出现了较为明显的“汉明悬崖”）。