

bs2int 参考资料

概要: 二进制串到整数值的转换。

描述:

该函数把二进制种群解码成十进制整数种群（无论它是标准的二进制编码还是格雷码），并且能够支持大数运算。

语法: Phen = bs2int(Chrom, FieldD)

详细说明:

Phen = bs2int(Chrom, FieldD) 根据区域描述器（又称译码矩阵）将用二进制/格雷码编码的种群矩阵 Chrom 解码成十进制的整数表示的种群矩阵 Phen。

二进制/格雷码种群 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。

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

译码矩阵 FieldD 具有下面的结构：

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

其中，*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* 和 *ubin* 的含义详见 `bs2rv` 函数的参考资料。在本函数中，这三个量并无实际用途，仅为了兼容 `bs2rv` 函数而设。因为本函数规定解码时使用算术尺度，并且包含变量的两个边界。至于要使用不包含变量的边界的使用场合，需要调用 `crtfld` 函数以生成符合规格的译码矩阵 FieldD。

应用实例:

调用 `crtbp` 函数生成一个二进制种群 Chrom，代表 2 个变量，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。

```
Chrom = crtbp(3, 5) # 调用crtbp创建一个3行5列的二进制种群矩阵

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

# 创建译码矩阵
FieldD = np.array([[2,3], [-4,-2], [-2,4], [1,1], [1,1], [1,1], [1,0]])
Phen = bs2int(Chrom, FieldD) # 进行解码
```

解码后结果如下：

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

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，然后把结果映射到变量的区间上，得到解码结果。若解码后结果超出变量的范围，则取变量的边界值。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，然后再继续转换。

本例中，根据译码矩阵 FieldD 可知，Chrom 中每一行的前两列代表第一个变量。可以看到 [0 1] 和 [0 0] 解码后都得到-4，这正体现了区间映射的作用。

译码矩阵的结构比较复杂，但作为一个开放式框架，你可以手写比较复杂的译码矩阵 FieldD，也可以调用 `crtfld` 函数来自动生成。推荐通过调用函数的方式生成。