bs2rv 参考资料

概要: 二进制串到实值的转换。

描述:

该函数把二进制种群解码成十进制实数种群（无论它是标准的二进制编码还是格雷码）。

语法: Phen = bs2rv(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 解码成十进制的实数表示的种群矩阵 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。

$scales$ 指明每个子串用的是算术刻度还是对数刻度。$scales[i] = 0$ 为算术刻度，$scales[i] = 1$ 为对数刻度。对数刻度可以用于变量的范围较大而且不确定的情况，对于大范围的参数边界，对数刻度让搜索可用较少的位数，从而减少了遗传算法的计算量。

$lbin$ 和$ubin$ 指明了变量是否包含其范围的边界。0 表示不包含边界；1 表示包含边界。

解码公式：设二进制染色体的某个片段为：$b_k b_{k-1} b_{k-2} \cdots b_2 b_1$，它解码后表示为一个范围在 [lb,ub] 上的实数，设解码得到的值为 X 则：

$$\mathrm{X} = lb + \left(\sum_{i}^{k} b_i \cdot 2^{i-1} \right) \cdot \frac{ub - lb}{2^k - 1}$$

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 [2,10] 和 [2,10]。用 bs2rv 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6列的二进制种群矩阵
```

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

```
# 创建译码矩阵
FieldD = np.array([[3,3], [2,2], [10,10], [0,0], [0,0], [1,1], [1,1]])
Phen = bs2rv(Chrom, FieldD) # 进行解码
```

解码后结果如下：

$$\mathrm{Phen} = \begin{pmatrix} 6.57142857 & 2.0 \\ 2.0 & 10.0 \\ 4.28571429 & 3.14285714 \end{pmatrix}$$

若采用另一个译码矩阵，则会解码得到不同的表现型：

```
# 创建译码矩阵
FieldD = np.array([[3,3], [2,2], [10,10], [0,0], [1,1], [0,1], [1,0]])
Phen = bs2rv(Chrom, FieldD) # 进行解码
```

解码后结果如下：

$$\mathrm{Phen} = \begin{pmatrix} 5.46872706 & 2.0 \\ 2.44568909 & 8.17765434 \\ 3.6571582 & 2.44568909 \end{pmatrix}$$

特别说明：当使用对数刻度时，对应的变量范围不能包含 0。

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。详见“crtfld 参考资料”。

参考文献:

[1] R. B. Holstien, Artificial Genetic Adaptation in Computer Control Systems, Ph.D. Thesis, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, 1971.

[2] R. A. Caruana and J. D. Schaffer, “Representation and Hidden Bias: Gray vs. Binary Coding”, Proc. 6th Int. Conf. Machine Learning, pp153-161, 1988.

[3] W. E. Schmitendorgf, O. Shaw, R. Benson and S. Forrest, “Using Genetic Algorithms for Controller Design: Simultaneous Stabilization and Eigenvalue Placement in a Region”, Technical Report No. CS92-9, Dept. Computer Science, College of Engineering, University of New Mexico, 1992.