

crtfld 参考资料

概要: 译码矩阵（区域描述器）生成函数。

描述:

该函数根据输入的参数生成译码矩阵（区域描述器）FieldD 或 FieldDR。

语法:

当 Encoding 为’RI’ 或’P’ 时：
FieldDR = crtfld(Encoding, varTypes, ranges, borders)
FieldDR = crtfld(Encoding, varTypes, ranges, borders, contraction)
当 Encoding 为’BG’ 时：
FieldD = crtfld(Encoding, varTypes, ranges, borders, precisions)
FieldD = crtfld(Encoding, varTypes, ranges, borders, precisions, codes)
FieldD = crtfld(Encoding, varTypes, ranges, borders, precisions, codes, scales)

详细说明:

译码矩阵（区域描述器）的结构较为复杂，该函数提供了一个自动化的方法来生成区域描述器，译码矩阵（区域描述器）有 FieldD 和 FieldDR 两种，相关概念详见概念详见 bs2int 以及 boundfix 的参考资料，或者查看 “Geatpy 数据结构” 文档。

Encoding 是染色体的编码方式，’BG’: 二进制/格雷编码; ’RI’: 实整数编码; ’P’: 排列编码。

当编码整数时，此时该函数会对 borders 标记为 0 的变量的范围往里收缩，同时对范围进行向里取整，最后把 borders 元素全设为 1。例如：

$$ranges = \begin{pmatrix} 1 & 2 & 3.1 \\ 3 & 4 & 5 \end{pmatrix}$$
$$borders = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

则 ranges 会被修正为：

$$ranges = \begin{pmatrix} 1 & 3 & 4 \\ 2 & 4 & 5 \end{pmatrix}$$

相应地，borders 会被修正为：

$$borders = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

varTypes 是一个连续/离散标记 (numpy 的 array 类型的行向量)，表示对应的变量是连续抑或是离散的（0 表示连续，1 表示离散）。

ranges 是一个 2 行 n 列的矩阵 (numpy 的 array 类型)，代表 n 个变量的边界范围。

其中第 0 行代表各个变量的下界；第 1 行是代表各个变量的上界。

borders 是一个 2 行 n 列的矩阵，代表 n 个变量是否包含区间的边界，0 表示不包含该边界，1 表示包含。

其中第 0 行代表是否包含各个变量的下界；第 1 行是代表是否包含各个变量的上界。

contraction 是一个一维 list，表示决策变量的“边界收缩率”，其元素必须是非负的，缺省或为 None 时默认元素全为 0。当控制变量范围不包含边界时，crtfld 函数会根据 precisions 把范围收缩一定的程度。比如精度设为 1 时，而边界为 0（即不包含范围边界），则 crtfld 函数会把范围往里收缩 0.1。

precisions 是一个一维 list，表示实数的二进制/格雷码编码的编码精度。例如其中一个元素是 4，表示对应变量的编码可以精确到小数点后 4 位。

注意：crtfld 函数的第 3 个参数具有两重含义，当 Encoding 为’R’,’I’ 或’P’ 时，第 3 个参数的含义是 contraction；当 Encoding 为’BGR’ 或’BGI’ 时，该第 3 个参数的含义是 precisions。

codes 是可选参数，是一个一维 list，表示变量是用什么方式编码的，例如其中一个元素为 0 时表示对应的变量是采用标准二进制编码，1 表示格雷编码。

scales 是可选参数，是一个一维 list，指明变量用的是算术刻度还是对数刻度，其元素为 0 或 1。例如其中一个元素是 0，表示对应变量是采用算术刻度；1 表示采用对数刻度。缺省或为 None 时默认其元素全为 0。采用对数刻度时，对应的决策变量的范围不能包含 0，即决策变量范围的上界和下界必须是同号的。

特别注意：crtfld 函数会自动对变量的 ranges 范围以及 borders 边界进行处理，最终返回一个规范的区域描述器。

应用实例:

例 1：下面欲创建包含变量 x_1, x_2 的整数值种群，2 个变量的区间范围分别是 [-3,5) 和 [2,10], 分别使用对数刻度的标准二进制编码和算术刻度的格雷编码，创建一个区域描述器来描述它。

```
import numpy as np
import geatpy as ea
x1 = [-3, 5] # 自变量1的范围
x2 = [2, 10] # 自变量2的范围
b1 = [1, 0] # 自变量1的边界
b2 = [1, 1] # 自变量2的边界
varTypes = np.array([0, 0]) # 生成连续/离散标记
ranges = np.vstack([x1, x2]).T # 生成自变量的范围矩阵
borders = np.vstack([b1, b2]).T # 生成自变量的边界矩阵
precisions = [0, 3] #
    实数的二进制/格雷编码的精度，0表示精确到个位
codes = [0, 1] # 变量编码方式，分别采用二进制和格雷编码
scales = [0, 1] # 采用算术刻度
# 调用crtfld函数生成区域描述器
FieldD = ea.crtfld('BG', varTypes, ranges, borders, precisions,
    codes, scales)
```

$$\text{FieldD} = \begin{pmatrix} 4 & 13 \\ -3 & 2 \\ 5 & 10 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

FieldD 中，第一行的lens 参数是根据变量的范围计算得到的。本例中两个决策变量分别需要用 4 位和 13 位的二进制数来进行编码，因此lens 参数的值是 [4,13]。

例 2：欲创建一个包含变量 x_1, x_2, x_3 的实数值种群，3 个变量的区间范围分别是：(-2.5,2), [3,5], [-4.8,3.6)，分别精确到小数点后 2、3、4 位。创建一个描述它的区域描述器：

```
import numpy as np
import geatpy as ea
x1 = [-2.5, 2] # 自变量1的范围
x2 = [3, 5] # 自变量2的范围
x3 = [-4.8, 3.6] # 自变量3的范围
b1 = [0, 0] # 自变量1的边界
b2 = [1, 1] # 自变量2的边界
b3 = [1, 0] # 自变量3的边界
varTypes = np.array([0, 0, 1]) # 生成连续/离散标记
ranges = np.vstack([x1, x2, x3]).T # 生成自变量的范围矩阵
borders = np.vstack([b1, b2, b3]).T # 生成自变量的边界矩阵
contraction = [2, 3, 4] # 各变量的边界收缩率
# 调用crtfld函数生成区域描述器
FieldDR = ea.crtfld('RI', varTypes, ranges, borders, contraction)
```

$$\text{FieldDR} = \begin{pmatrix} -2.49 & 3.0 & -4. \\ 1.99 & 5. & 3. \\ 0. & 0. & 1. \end{pmatrix}$$