

indexing 参考资料

概要: 基于指数变换的适应度计算。

描述:

该函数对目标函数值 **ObjV** 作指数变换, 使其变成受 β 影响的指数尺度的适应度值。

变换公式: $Fit = e^{-\beta * ObjV} + 1$

该函数遵循“最小适应度为 0”的约定 (特殊情况除外)。

语法:

```
FitnV = indexing(ObjV)
FitnV = indexing(ObjV, LegV)
FitnV = indexing(ObjV, LegV, Beta)
FitnV = indexing(ObjV, LegV, Beta, SUBPOP)
```

详细说明:

该函数先将个体的目标值 **ObjV** 进行 0-1 标准化, 然后进行指数变换, 最终返回一个代表种群适应度的列向量 **FitnV**。

LegV 是一个可选参数, 保存着个体对应的可行性的列向量, 0 表示该个体是非可行解, 1 表示是可行解。

Beta 是一个正实数, 其值影响指数变换。缺省情况下默认 **Beta** 为 1。

SUBPOP 是一个正整数, 代表子种群的数量。**SUBPOP** 必须能够被种群个体数整除。关于子种群的概念详见 **migrate** 参考资料。

注: **Geatpy** 的适应度遵循“种群目标函数值越大, 适应度越小”的原则。并且当子种群的所有个体的目标函数值相等时, 其对应的适应度值均为 1。

特别注意:

本函数是根据传入参数 **ObjV** 来计算适应度的, 且遵循“种群目标函数值越大, 适应度越小”的原则, 因此在调用本函数前, 需要对传入的 **ObjV** 乘上‘maxormin’(最大最小化标记)。但是, 由于返回的是 **FitnV**, 它与 **ObjV** 在含义上无关了, 因此不需要对其乘上‘maxormin’进行还原。

应用实例:

现有一个拥有 10 个个体的种群, 每个个体的目标函数值为 1, 2, 2, 4, 5, 10, 9, 8, 7, 6, 若该种群包含 2 个子种群, 求其指数尺度变换的适应度值。

```
ObjV = np.array([[1],[2],[2],[4],[5],[10],[9],[8],[7],[6]])
LegV = np.array([[1],[1],[1],[1],[1],[ 1],[1],[1],[1],[1]])
FitnV = indexing(ObjV, LegV, 2, 2) # 设定Beta值为2, SUBPOP为2
```

$$FitnV = \begin{pmatrix} 2 \\ 1.60653066 \\ 1.60653066 \\ 1.22313016 \\ 1.13533528 \\ 1.13533528 \\ 1.22313016 \\ 1.36787944 \\ 1.60653066 \\ 2 \end{pmatrix}$$

解析: 该种群拥有 10 个个体, 但因为有 2 个子种群, 所以前 5 个为第一个子种群, 后 5 个为第二个子种群。由结果可见, 目标函数值越大的个体适应度值越小。