

ranking 参考资料

概要: 根据目标函数值排序的适应度分配

描述: 该函数实现了” 基于等级划分的适应度分配” 算法, 返回包含种群个体适应度的列向量。  
遵循” 目标函数值越大适应度越小” 的约定。

语法: `FitnV = ranking(ObjV)`  
`FitnV = ranking(ObjV, RFun)`  
`FitnV = ranking(ObjV, RFun, SUBPOP)`

详细说明: `ObjV` 是一个保存着个体对应的目标函数值的列向量。  
`RFun` 是一个可选参数:  
1. 如果 `RFun` 是一个在 [1, 2] 范围内的 1\*1 的矩阵, 则函数采用线性排序, 此时 `RFun` 代表选择压力;  
2. 如果 `RFun` 是一个 1\*2 的 array, 则:  
`RFun[0]`: SP 是一标量, 指定了选择压力;  
`RFun[1]`: RM 指定排序方式:  
当 `RM = 0` 为线性排序, 此时 `RFun[1]` 要在 [1,2] 范围内;  
当 `RM = 1` 为非线性排序, 此时 `RFun[2]` 要在 [1,len(`ObjV`)-2] 范围内。  
3. 如果 `RFun` 是一个长度等于 `ObjV` 长度的行矩阵, 则表示对 `ObjV` 每一行的适应度值计算, 此时 `RFun` 通常是一个元素递增的行向量;  
4. 如果 `RFun` 是缺省或者设为 `None`, 则默认采用线性排序且选择压力为 2。  
`SUBPOP` 表示子种群的数量, 是一个可选参数。`SUBPOP` 要求能够被种群个体数整除。当该参数缺省时, 子种群的数量默认值为 1。关于 `Geatpy` 子种群的有关概念详见 `migrate` 函数的参考资料。

算法说明: 该函数实现了基于等级划分的适应度分配算法 (算法描述详见” 进化算法介绍” 中的” 适应度计算” 章节)。  
值得注意的是, 由于矩阵 `FitnV` 未被排序, 所以可以反映初始输入矩阵 `ObjV` 的顺序。

应用实例: 考虑有 10 个个体的种群, 其当前目标值 `ObjV` 如下情况。

```
ObjV=np.array([[ 1],[ 2],[ 3],[ 4],[ 5],[10],[ 9],[ 8],[ 7],[ 6]])
```

(1) 使用线性排序和选择压力为 2, 求适应度:

```
RFun = np.array([[2,0]])
FitnV = ranking(ObjV, RFun)
```

得到 `FitnV`:

$$\text{FitnV} = \begin{pmatrix} 2 \\ 1.77777778 \\ 1.55555556 \\ 1.33333333 \\ 1.11111111 \\ 0 \\ 0.22222222 \\ 0.44444444 \\ 0.66666667 \\ 0.88888889 \end{pmatrix}$$

(2) 采用非线性排序和选择选择压力为 2, 求适应度:

```
RFun = np.array([[2,1]])
FitnV = ranking(ObjV, RFun, 1)
```

得到 `FitnV`:

$$\text{FitnV} = \begin{pmatrix} 2 \\ 1.66331464 \\ 1.38330779 \\ 1.15043805 \\ 0.95677023 \\ 0.38065341 \\ 0.45770463 \\ 0.55035244 \\ 0.66175385 \end{pmatrix}$$

参考文献: [1] D. Whitley, “The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best” , Proc. ICGA 3, pp. 116-121, Morgan Kaufmann Publishers, 1989.