scaling 参考资料

概要： 线性尺度变换适应度计算

描述:

该函数对目标函数值 ObjV 进行线性变换，

遵循"最小适应度为 0"的约定（特殊情况除外）。

语法:

FitnV = scaling(ObjV)

FitnV = scaling(ObjV, LegV)

FitnV = scaling(ObjV, LegV, Smul)

FitnV = scaling(ObjV, LegV, Smul, SUBPOP)

详细说明:

ObjV 为一个保存着个体对应的目标函数值的列向量。

LegV 是一个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。

Smul(可选参数) 线性变换的决定上界的常数。若缺省或为 None, 则默认为 2。

SUBPOP(可选参数) 为子种群的数量, 要求能够被种群个体数整除。若缺省或为 None，默认值为 1。

FitnV 为记录着种群个体适应度值的列向量。

算法:

该函数计算出由 Smul 的值决定上界的适应度值即 $F' = aF + b$。

其中 F 为直接把目标函数值 ObjV 作为适应度值的函数：F(个体) = ObjV

F' 为对 F 进行线性尺度变换后的适应度值函数，且 F' 要满足以下两个条件：

1. 为了保证适应度在平均值的个体在下一代的期望复制数为 1，因此要使得原适应度的平均值 Favg = 新适应度的平均值 F'avg。

2. 为了控制适应度最大的个体在下一代中的复制数，因此设定变换后的适应度最大值等于原适应度平均值的。指定倍数，即：F'max = Smul * Favg

经过线性变换后，原适应度高的少数个体的适应度等比例缩小，同时适应度较差的个体的适应度等比例扩大，从而有利于种群的多样性。

注意: 为遵循" 目标函数值越大适应度越小" 的约定，算法中先对 ObjV 取相反数。

根据上述条件进行线性变换后，原适应度最小的个体可能因为这种变换而导致适应度小于 0，此时我们要特殊处理这种情况，采用另一种变换方式：

1. 变换后适应度再加 1 使最小适应度为 1，即：F'min = 1

2. 不再要求 F'avg = Favg, 而是设置斜率 dF'/dF = Favg / (Favg - Fmin)

算法处理了以下 3 种特殊情况：

1. 当种群的所有个体的目标函数值都相等时，此时算法中作为分母的 delta 会为 0，因此假如遇到这种情况，则将 FitnV 设置为全 1 的列向量

2. 有可能上述情况下 delta 的值相当的小，但不为 0，此时也要将 FitnV 设置为 1 的列向量。

3. 对于传入的 ObjV 均小于 0 的情况，要先进行平移变换，使所有的 ObjV 都 >=0。

4. 如果种群某些个体的目标函数值为 nan 或 None（即不合法），而可行性列向量又没有作标记，此时函数将对这些额外的非可行解做出标记，更新 LegV。

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:

根据目标函数值 ObjV，利用线性尺度变换求其对应的适应度：

```
ObjV = np.array([[1], [2], [3], [4], [5], [10], [9], [8], [7], [6]])
LegV = np.array([[1], [1], [1], [1], [1], [ 1], [1], [1], [1], [1]])
FitnV = scaling(ObjV, LegV) # 进行线性尺度变换适应度计算
```

得到适应度如下：

$$\mathrm{FitnV} = \begin{pmatrix} 10 \\ 9 \\ 8 \\ 7 \\ 6 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$

参考文献:

[1] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley Publishing Company, January 1989.