

reins 参考资料

概要： 在种群中重插入育种个体

描述：
该函数将育种个体重插入到父代种群中， 生成新一代种群。

语法：
Chrom = reins(Chrom, SelCh)
Chrom = reins(Chrom, SelCh, SUBPOP)
Chrom = reins(Chrom, SelCh, SUBPOP, Select)
Chrom = reins(Chrom, SelCh, SUBPOP, Select, INSR)
Chrom = reins(Chrom, SelCh, SUBPOP, Select, INSR, FitnVCh)
Chrom = reins(Chrom, SelCh, SUBPOP, Select, INSR, FitnVCh, FitnVSel)
[Chrom, ObjV] = reins(Chrom, SelCh, SUBPOP, Select, INSR, FitnVCh, FitnVSel,ObjVCh, ObjVSel)
[Chrom, ObjV, LegV] = reins(Chrom, SelCh, SUBPOP, Select, INSR, FitnVCh, FitnVSel, ObjVCh, ObjVSel, LegVCh, LegVSel)

详细说明：
reins 将育种个体插入到当前种群中， 用代替父代某些个体并返回重插入后的新一代种群。
Chrom 为父代种群矩阵， 其每行代表一个个体的染色体。
SelCh 为选择、交叉、变异等操作后得到的育种种群矩阵， 术语上称作” 育种种群”， 其每行对应一个育种个体。
SUBPOP (可选参数) 表示子种群的数量， 若缺省或设为 None， 则默认是 1。
Select (可选参数) 指明育种个体替代父代个体的选择方法：
0 为均匀选择；
1 为基于适应度的选择；
如果 Select 缺省或为 None， 则默认为 0。
在基于适应度的选择中， 适应度强的个体被用于替换父代的适应度差的个体。
INSR (可选参数) 表示选择重插入的育种个体数占全部育种个体数的比率 (即选择了百分之多少的育种个体)。如果缺省或设为 None， 则默认为 1.0。
FitnVCh 是一个保存着父代种群的个体对应的适应度值的列向量。
对基于适应度的重插入 (即当 Select 为 1 时)， FitnVCh 发挥作用。
FitnVSel 为一个保存着育种种群的个体对应的适应度值的列向量。
如果所有育种个体的数量大于重插入到种群中的育种个体数量， 则 FitnVSel 发挥作用。此时将按育种个体的适应度从大到小的顺序选择插入。
ObjVCh 是一个保存着父代种群的个体对应目标函数值的矩阵 (可以是多目标)。
ObjVSel 为一个保存着育种种群的个体对应的目标函数值的矩阵 (可以是多目标)。
LegVCh 是一个保存着父代种群的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。
LegVSel 是一个保存着育种种群的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。
注意： 当给 reins 函数传入 ObjVCh 参数时， 也要传入 ObjVSel， 即不能缺省。此时函数将不仅返回重插入后的种群矩阵， 还会返回重插入后种群的目标函数值矩阵。
同理， 当给 reins 函数传入 LegVCh 参数时， 也要传入 LegVSel， 即不能缺省。此时函数将不仅返回重插入后的种群矩阵， 还会返回重插入后种群的目标函数值矩阵。
一旦传入 LegVCh 和 LegVSel， 要求也要传入合法的 ObjVCh 和 ObjVSel。
此外， 可以将目标函数值当作适应度传入本函数中， 但传入前要乘上 maxormin， (maxormin 为最大最小化标记， 它为 1 时表示这是个最小化目标， 为-1 时表示这是个最大化目标)。

特别注意：
本函数是根据 FitnVCh 和 FitnVSel 来进行重插入的， 与 ObjVCh 和 ObjVSel 无关， 因此在调用本函数前， 不需要对传入的 ObjVCh 和 ObjVSel 乘上’maxormin’(最大最小化标记)， 对于返回的 ObjV， 也不需要乘上’maxormin’ 进行还原。

应用实例：
现有四个变量， 范围分别是 [-10,10]、 [-5,5]、 [-3,3]、 [-1,1]。 创建一个含有这 4 个变量的 6 个个体的实数值种群 Chrom， 同时再创建一个含有 2 个个体的实数值种群 SelCh 来重插入到 Chrom 中。

```
FieldDR = np.array([[ -10,-5,-3,-1],[10, 5, 3, 1]]) # 创建区域描述器
Chrom = crtrp(6, FieldDR) # 创建含有6个个体的种群， 把它看作父代种群
# 创建列向量来存储父代种群个体的目标函数值
FitnVCh = np.array([[21,22,23,16,15,24]]) . T
SelCh=crtrp(2, FieldDR) #
    创建含有2个个体的种群， 看成是待重插入的育种种群
# 把育种个体重插入到父代种群中
Chrom = reins(Chrom, SelCh, 1, 1, 1, FitnVCh)
```

插入前父代种群如下：

$$\text{Chrom} = \begin{pmatrix} 1.46122027\text{e} - 01 & 3.45234379\text{e} + 00 & 2.31583857\text{e} + 00 & 6.91123313\text{e} - 01 \\ 5.67085869\text{e} + 00 & 3.52743074\text{e} - 01 & 1.57330911\text{e} + 00 & 2.97135778\text{e} - 03 \\ 5.19726307\text{e} + 00 & 2.67165148\text{e} + 00 & 7.85408841\text{e} - 01 & 1.89785150\text{e} - 01 \\ 2.25048911\text{e} + 00 & 4.45678441\text{e} + 00 & 1.92489047\text{e} + 00 & 6.05092404\text{e} - 01 \\ 1.49160644\text{e} + 00 & 3.71236655\text{e} + 00 & 2.69765077\text{e} + 00 & 3.12855563\text{e} - 01 \\ 5.00117627\text{e} + 00 & 4.05129548\text{e} + 00 & 2.78183093\text{e} + 00 & 1.33010496\text{e} - 01 \end{pmatrix}$$

待插入的育种种群如下：

$$\text{Selch} = \begin{pmatrix} 1.08190019 & 3.99550597 & 0.72815683 & 0.31596068 \\ 3.49844636 & 0.00448962 & 0.63786374 & 0.94370521 \end{pmatrix}$$

重插入得到的新一代种群如下：

$$\text{Chrom} = \begin{pmatrix} 1.46122027\text{e} - 01 & 3.45234379\text{e} + 00 & 2.31583857\text{e} + 00 & 6.91123313\text{e} - 01 \\ 5.67085869\text{e} + 00 & 3.52743074\text{e} - 01 & 1.57330911\text{e} + 00 & 2.97135778\text{e} - 03 \\ 5.19726307\text{e} + 00 & 2.67165148\text{e} + 00 & 7.85408841\text{e} - 01 & 1.89785150\text{e} - 01 \\ 3.49844636\text{e} + 00 & 4.48961622\text{e} - 03 & 6.37863745\text{e} - 01 & 9.43705208\text{e} - 01 \\ 1.08190019\text{e} + 00 & 3.99550597\text{e} + 00 & 7.28156829\text{e} - 01 & 3.15960684\text{e} - 01 \\ 5.00117627\text{e} + 00 & 4.05129548\text{e} + 00 & 2.78183093\text{e} + 00 & 1.33010496\text{e} - 01 \end{pmatrix}$$

对比重插入前后的 Chrom 矩阵， 可以看出重插入前目标函数值最大 (遵循” 目标函数值越大适应度越小” 的约定) 的两个个体在重插入过程中被育种个体替换了。