

selecting 参考资料

概要： 高级选择函数

描述：
此函数执行通用选择，处理多个种群，选择过程实际上调用了低级选择函数。

语法：
SelCh = selecting(SEL_F, Chrom, FitnV)
SelCh = selecting(SEL_F, Chrom, FitnV, GGAP)
SelCh = selecting(SEL_F, Chrom, FitnV, GGAP, SUBPOP)
[SelCh, ObjVSel] = selecting(SEL_F, Chrom, FitnV, GGAP, SUBPOP, ObjV)
[SelCh, ObjVSel, LegVSel] = selecting(SEL_F, Chrom, FitnV, GGAP, SUBPOP, ObjV, LegV)

详细说明：
函数 select 从种群 Chrom 中选择优良个体，并将选择的个体返回到新种群 SelCh 中。
SEL_F 为包含低级选择函数名称的字符串如 ‘rws’。
Chrom 为包含当前种群的染色体矩阵，每一行对应一个个体的一条染色体。
FitnV 为包含种群 Chrom 中个体的适应度值的列向量。
GGAP (可选参数) 表示代沟，表示被选择的种群的比率, 默认值为 1.0。允许代沟大于 1，表示子代种群个体数大于父代种群个体数。
SUBPOP (可选参数) 表示子种群的数量。如果 SUBPOP 缺省或设为 None 时, 默认的 SUSPOP=1。Chrom 中的所有子种群个体数量是相等的。
ObjV 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)。
LegV 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。
SelCh, ObjVSel, LegVSel 的含义分别和 Chrom, ObjV, LegV 相同，前者是后者经过选择后得出的。

算法说明：
selecting 检测输入参数的一致性并调用低级选择函数，遍历所有子种群进行选择操作。

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

应用实例：
考虑以下具有 8 各个体的种群 Chrom，其适应度为 FitnV。

```
Chrom=np.array([[1,11,21],
                [2,12,22],
                [3,13,23],
                [4,14,24],
                [5,15,25],
                [6,16,26],
                [7,17,27],
                [8,18,28]])

FitnV = np.array( [[1.50,1.35,1.21,1.07,0.92,0.78,0.64,0.5]]) .T
SelCh = selecting('sus',Chrom,FitnV)  #使用随机遍历抽样sus选择个体
```

得到 SelCh:

$$\text{SelCh} = \begin{pmatrix} 1 & 11 & 21 \\ 4 & 14 & 24 \\ 3 & 13 & 23 \\ 1 & 11 & 21 \\ 2 & 12 & 22 \\ 6 & 16 & 26 \\ 4 & 14 & 24 \\ 7 & 17 & 27 \end{pmatrix}$$