

redisNDSets 参考资料

概要: 基于拥挤距离的帕累托最优子集筛选 (re-choose base on crowding-distance)。

描述:

该函数实现了基于拥挤距离计算的帕累托最优子集的筛选, 筛选出符合数量的分布性较好的帕累托最优解子集。

语法:

```
[NDSetsSub, NDSetsObjVSub] = redisNDSets(NDSets, NDSetsObjV)
[NDSetsSub, NDSetsObjVSub] = redisNDSets(NDSets, NDSetsObjV, NUM)
```

详细说明:

NDSets 是筛选前的帕累托最优解集, 它是 `numpy array` 类型的矩阵, 每一行代表一个帕累托最优解的控制变量的值。有多少个控制变量, NDSets 就有多少列。

NDSetsObjV 是筛选前的帕累托最优集的目标函数值矩阵, 它也是 `numpy array` 类型的, 每一行代表各个帕累托最优解的目标函数值。有多少个目标, NDSetsObjV 就有多少列。

NUM 是可选参数, 表示需要筛选出的子集包含多少个帕累托最优解。当缺省或数值不大于当前帕累托最优解集 NDSets 的行数 (即帕累托最优解的个数) 时, 不进行筛选, 直接返回传入的 NDSets 和 NDSetsObjV。

NDSetsSub 和 NDSetsObjVSub 分别是筛选后的 NDSets 和 NDSetsObjV。

调用 `redisNDSets` 函数是实现 NSGA2 算法必不可少的一步。当然, 在其他多目标优化算法中, 也可以调用该函数, 以增强帕累托最优解的分布性。

特别注意:

本函数是根据传入参数 NDSetsObjV 来计算拥挤距离的, 且遵循“最小化目标”的约定, 但是, 在对 NDSetsObjV 进行排序时, 无论是从小到大排序还是从大到小排序, 对拥挤距离的计算是无影响的, 因此, 不需要对传入的 NDSetsObjV 乘上 'maxormin' (最大最小化标记)。也不需要返回的 NDSetsObjVSub 其乘上 'maxormin' 进行还原。

参考文献:

[1] ARAVIND SESHADRI. A FAST ELITIST MULTIOBJECTIVE GENETIC ALGORITHM: NSGA-II.