

ndominfast 参考资料

概要: 快速非支配排序 (nondominated-sorting fast)。

描述:

快速非支配排序算法与简单非支配排序法类似，但不同的是它不会遍历种群的全部个体，而是遍历排除了被支配个体后的种群，每趟遍历都对整个种群中的被支配个体进行淘汰。函数最后根据种群个体的非支配情况计算并返回种群个体适应度及非支配个体的索引。

语法:

```
[FitnV, frontIdx] = ndominfast(ObjV)
```

```
[FitnV, frontIdx] = ndominfast(ObjV, FitnV)
```

详细说明:

ObjV 是种群个体的目标函数矩阵，每一行对应一个个体，每一列对应一个目标。

FitnV 是一个可选参数，表示用其他算法求得的种群个体适应度列向量，当缺省或为 None 时，表明没有用其他算法求种群个体的适应度，此时该函数将根据个体被支配的情况计算适应度。

函数返回经过快速非支配排序后的种群个体适应度以及非支配个体在种群中的索引：frontIdx，它是一个 array 类型的行向量。

注意: Geatpy 的非支配排序均遵循最小化目标的约定。

应用实例:

考虑一个两个目标的优化问题，设种群规模为 20，这 20 个个体的目标函数值如下：
(9,1),(7,2),(5,4),(4,5),(3,6),(2,7),(1,9),(10,3),(8,5),(7,6),(5,7),(4,8),(3,9),(10,5),(9,6),(8,7),
(7,9),(10,6),(9,7),(8,9)

使用快速非支配排序法计算该种群的非支配个体：

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

得到的非支配个体索引 frontIdx 为：0 1 2 3 4 5 6

即前 7 个个体是种群中的非支配个体。

再次提醒的是，frontIdx 是 numpy 的 array 类型的行向量，行向量和行矩阵的关系在“Geatpy 数据结构”章节中有详细描述。可以使用 print(frontIdx.shape) 来查看 frontIdx 的规格，可以发现结果为 (7,)，若是行矩阵的话，输出的是 (1,7) 而不是 (7,)。