

ndomin 参考资料

概要: 简单非支配排序 (nondominated-sorting)。

描述:

该函数利用简单非支配排序法构造种群的非支配集，即简单地遍历种群所有个体，计算每个个体被多少个其他个体支配，根据支配该个体的个体数计算并返回种群个体适应度及非支配个体的索引。若传入 `exIdx` 参数，则会对非可行解的个体进行排除。

语法:

```
[FitnV, frontIdx] = ndomin(ObjV)
[FitnV, frontIdx] = ndomin(ObjV, exIdx)
```

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。
`exIdx` 是一个可选参数，它是 $1 \times n$ 的 `numpy` 的 `array` 类型行向量，代表种群中对应非可行解的个体下标。它使程序能够避免受到非可行解对非支配排序的影响
函数返回经过简单非支配排序后的种群个体适应度列向量 `FitnV` 以及非支配个体在种群中的索引：`frontIdx`，它是一个 `numpy` 的 `array` 类型行向量。

- 简单非支配排序的算法流程如下：
1. 遍历种群所有个体，计算各个个体被多少个种群中的其他个体支配 (保存在 `snp` 集合中)，
 2. 根据第 2 步得到的 `snp` 集合，其中为 0 的元素对应的个体即为当代种群的非支配个体，
 3. 个体的适应度 = 种群个体数 / (支配该个体的个体数 + 1)。
- 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)
使用简单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] = ndomin(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,)。