

复现“用于 MOF 阵列优化的遗传算法库”开发文档

一、复现文献：

Gustafson J A, Wilmer C E. Intelligent Selection of Metal-Organic Framework Arrays for Methane Sensing via Genetic Algorithms[J]. ACS SENSORS, 2019, 4(6): 1586-1593

语言：Python

版本：3.8.0

二、对文献中所使用遗传算法的分析

2.1. 遗传算法运算关键参数和运算流程

2.1.1. 关键参数设置

种群大小 = 20

MOF 阵列大小 = 用户自定义

候选 MOF 列表 = range(1 : 50)

在竞争中胜出的顶部染色体个数 = $20 \times 20\% = 4$ （在文献作者源码中是 2）

在竞争中胜出的幸运染色体个数 = $20 \times 20\% = 4$ （在文献作者源码中是 2）

单独 MOF 对应值：随机生成的大小为“MOF 阵列大小”的 list 型数据（原因参见 2.2.3.）

2.1.2. 运算流程设计

第一步：创建初始种群；

第二步：修复初始种群中重复的染色体和染色体上重复的基因；

第三步：开始遗传运算的循环。

循环第一步：计算种群中每个染色体的适值；

循环第二步：按照适值大小将适值和种群列表对应排序；

循环第三步：保存当前种群和适值至文件“generations.csv”，保存每一代最大适值和平均适值至文件“Fitness_in_generations.csv”；

循环第四步：选择父代；

循环第五步：产生子代：选择使用“交叉”+“变异”或单独“变异”策略；

循环第六步：修复种群中重复的染色体和染色体上重复的基因；

2.2. 遗传算法各部分的具体实现方法

2.2.1. 创建初始种群的实现方法

输入变量：种群大小、MOF 阵列大小、候选 MOF 列表；

返回值：初始种群列表。

运算流程：

- 1、根据 MOF 阵列大小和候选 MOF 列表，使用 random.sample 函数随机生成初始种群的个体，并将每个个体内的基因从小到大排列；
- 2、重复生成直至个体数量与种群大小相等。

2.2.2. 修复种群中重复的染色体和染色体上重复的基因

输入变量：种群大小、种群列表、候选 MOF 列表；

返回值：种群列表。

运算流程：

- 1、修复种群列表中每个染色体的重复基因；
- 2、删除种群列表中重复的染色体，使用遍历比较的方法；
- 3、使用 random.sample 函数随机生成的个体，将种群列表补充到要求的种群大小；

4、重复 2、3 直至没有重复个体出现。

2.2.3. 适值计算的实现方法

适值函数改动说明: 为使程序得以运行, 并达到减少开发时间, 提高 GA 成功率的目的, 将每个 MOF 与一个数对应, 问题转化为, 在 50 个数中寻找一个元素乘积最大的特定大小组合的问题。该问题的优点在于可以事先设置真正的元素乘积最大组合, 便于对迭代结果的验证。

输入变量: 种群列表、单独 MOF 对应值;

返回值: 适值列表。

计算方法:

- 1、生成一个长度为“MOF 阵列大小”的 list 型数据做适值列表。
- 2、依次提取种群列表中的个体;
- 3、根据个体中的基因值, 提取“单独 MOF 对应值”中的对应值;
- 4、将所有值相乘, 得到当前个体的适值;
- 5、依次保存适值到适值列表的对应位置。

2.2.4. 选择父代的实现方法

输入变量: 种群列表、在竞争中胜出的顶部染色体个数、在竞争中胜出的幸运染色体个数;

返回值: 父代列表 (赋给了种群列表, 完成了优胜劣汰)。

运算流程:

- 1、选择适值最好的“在竞争中胜出的顶部染色体个数”个染色体幸存;
- 2、在最好的“在竞争中胜出的顶部染色体个数”个染色体以外, 使用 random.sample 函数随机选择“在竞争中胜出的幸运染色体个数”个染色体幸存;

2.2.5. 产生子代的实现方法

1、交叉+变异实现方法

输入变量: 父代列表 (实则为种群列表)、种群大小、变异率、候选 MOF 列表;

返回值: 种群列表。

运算流程:

- a) 建立一个父代列表的深度拷贝;
- b) 从拷贝中使用 random.sample 函数随机选择两个作为交叉对象;
- c) 将两个交叉对象的元素合并, 并去除重复项, 形成小基因库;
- d) 使用 random.sample 函数随机选择小基因库里的“MOF 阵列大小”个基因形成一个新的染色体个体;
- e) 对新产生的个体应用变异, 变异方法与变异实现方法 f 步骤相同, 但只针对当前新产生的个体产生变异;
- f) 重复 b、c、d、e 直到父代和子代个体数之和, 等于“种群大小”。

2、变异实现方法

输入变量: 父代列表 (实则为种群列表)、种群大小、变异率、候选 MOF 列表

返回值: 种群列表

运算流程:

- a) 建立一个父代列表的深度拷贝;
- b) 使用 random.choice 函数随机从深度拷贝中选取一个染色体作变异对象;
- c) 建立一个变异对象的浅拷贝, 称作子代; (因为染色体是 list, 所以不需要深度拷贝)
- d) 从候选 MOF 列表中挑选出不在变异对象里的 MOF, 建立一个小基因库;
- e) 建立一个小基因库的浅拷贝; (实际为引用, 但在这里效果没有差别)

- f) 依次扫描子代的基因作变异;
 - a) 使用 random.random 函数给当前基因产生一个 0-1 之间的随机变量;
 - b) 判断该随机变量是否小于变异率
 - i. 如果是:
 - 1. 使用 random.choice 随机从小基因库浅拷贝处获取一个基因, 并使用该基因替换当前基因;
 - 2. 从小基因库的浅拷贝中删去上一步随机选出的基因;
 - 3. 判断小基因库的浅拷贝是否为空, 如果是则将小基因库浅拷贝的值刷新为不包含替换基因后的子代基因的候选 MOF 列表;
 - ii. 如果否: 不变异。
- g) 将变异完成的子代基因由小到大排序;
- h) 重复 b、c、d、e、f、g 直到父代和子代个体数之和, 等于“种群大小”。