

# Geatpy 总览

## 1. Geatpy 层次结构

Geatpy 是简单封装的开放式进化算法框架，可以方便、自由地与其他算法或项目相结合。其层次结构如下图：

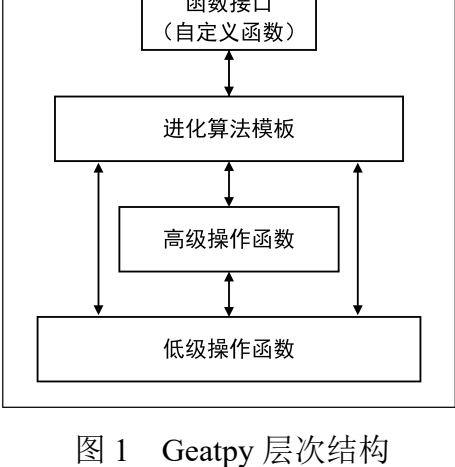


图 1 Geatpy 层次结构

中心是进化算法模板。它是为解决实际优化问题而编写的模板函数或脚本。里面清晰地定义了相关的变量的描述以及展现了遗传算法的详细流程。Geatpy 本身自带几种用于解决常见问题的进化算法模板，你可以根据实际需求修改或重构模板。

函数接口是指用户自定义的优化问题中的目标函数以及罚函数。进化算法模板可以调用该函数接口来计算种群个体的目标函数值。

此外，你也可以不用函数接口，直接把函数定义写在进化算法模板内部，虽然这样不利于扩展和接入到其他项目中，但对于学习进化算法以及做一些相关的基础性实验或者解决一些简单问题也很方便。

下面第 2 点展示了 Geatpy 的函数调用关系。中心是进化算法模板，它调用高级的运算函数 (selecting, recomb, mutate)。高级函数进一步调用相关的低级运算函数 (即实现选择、交叉、变异等底层算法的函数)。这种层级调用关系使 Geatpy 的结构十分清晰，更重要的是，你可以自定义更多低级运算函数来轻松自由地扩展 Geatpy。

**特别注意：**自定义的进化算法模板最好不要与 Geatpy 自带的模板名称重复，否则会造成难以处理的冲突问题。

## 2. Geatpy 调用关系

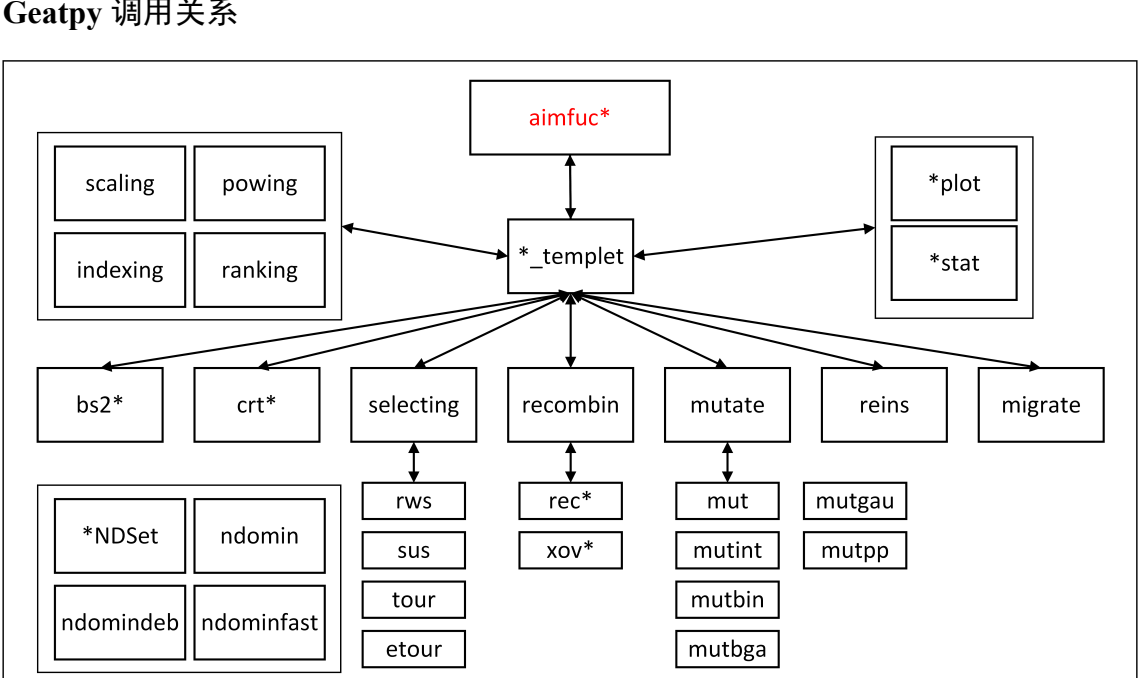


图 2 Geatpy 调用关系

图中黑色的表示 Geatpy 中内置的函数 (当然用户也可以自定义与之类似的函数)；红色的表示是用户自定义的。其中，templet\* 是进化算法模板，\* 表示名字是多样的；aimfuc\* 是优化目标函数；crt\* 用于创建种群染色体矩阵 (简称“种群矩阵”)；bs2\* 用于二进制/格雷码种群的解码；ranking, powing, scaling, indexing 均是用于计算种群适应度的函数；migrate 是种群迁移函数；selecting、recombin、mutate 分别是高级的选择、重组和变异函数，对应下方的都是低级的选择、重组和变异函数；reins 是重插入操作函数；\*plot 是以 plot 名字结尾的用于可视化输出函数。\*stat 是以 stat 名字结尾的统计函数，\*NDSet 是与多目标优化帕累托最优解集的处理有关的函数。

上述结构中，aimfuc\*, templet\*, treplot, replot 等都是可以自定义的，自定义时必须遵循相关的数据结构 (详见“Geatpy 数据结构”章节)。

## 3. Geatpy 主要功能

以下是 Geatpy 的内核函数，其详细用法可以参见“Geatpy 函数”章节。

### 3.1 与初始化种群相关的函数

- crtfld(生成区域描述器)
- crtbp(创建简单离散种群、二进制编码种群)
- crtbp(创建整数型种群)
- crtpp(创建排列编码种群)
- crtrp(创建实数型种群)
- meshrng(网格化变量范围)

### 3.2 进化迭代

当完成了种群的初始化后，就可以进行遗传进化迭代了。这部分是在进化算法模板里编写的。迭代过程中包括：

- 调用 ranking 或 scaling 等计算种群适应度。
- 调用 selecting 进行选择操作 (也可以直接调用低级选择函数)。
- 调用 recomb 进行重组操作 (也可以直接调用低级重组函数)。
- 调用 mutate 进行重组操作 (也可以直接调用低级变异函数)。
- 调用 reins 进行重插入生成新一代种群。
- 调用 migrate 进行种群迁移 (可以增加种群的多样性)。

### 3.3 适应度计算

- ranking(基于等级划分的适应度分配计算)
- scaling(线性尺度变换适应度计算)
- indexing(指数尺度变换适应度计算)
- powing(幂尺度变换适应度计算)

### 3.4 选择

selecting 是高级选择函数，它调用下面的低级选择函数：

- sus(随机抽样选择)
- rws(轮盘赌选择)
- tour(锦标赛选择)
- etour(精英锦标赛选择)

### 3.5 重组

重组包括了交叉。recombin 是高级的重组函数，它调用下面的低级重组函数：

- recdis(离散重组)
- recint(中间重组)
- reclin(线性重组)
- xovdp(两点交叉)
- xovdprs(减少代理的两点交叉)
- xovmp(多点交叉)
- xovpm(部分匹配交叉)
- xovsh(洗牌交叉)
- xovshrs(减少代理的洗牌交叉)
- xovsp(单点交叉)
- xovsprs(减少代理的单点交叉)

### 3.6 突变

mutate 是高级的突变函数，它调用下面的低级突变函数：

- mut(简单离散变异算子)
- mutbga(实数值变异算子)
- mutbin(二进制变异算子)
- mutgau(高斯突变算子)
- mutint(整数值变异算子)
- mutpp(排列编码变异算子)

### 3.7 重插入

reins 是重插入函数，它将育种个体重插入到父代种群中，生成新一代种群。

### 3.8 种群迁移

当使用多种群设计时，可用 migrate 函数实现种群中的个体迁移。

### 3.9 染色体解码

对于二进制/格雷编码的种群，我们要对其进行解码才能得到其表现型。

- bs2int(二进制/格雷码转整数)
- bs2rv(二进制/格雷码转实数)

### 3.10 可视化

- frontplot(多目标优化帕累托前沿绘图函数)
- sgaplot(单目标进化动态绘图函数)
- treplot(单目标进化跟踪器绘图)

### 3.11 多目标相关

- ndomin(简单非支配排序)
- ndomindeb(Deb 非支配排序)
- ndominfast(快速非支配排序)
- redisNDSet(基于聚集距离的帕累托最优子集筛选)
- upNDSet(更新帕累托最优集)

### 3.12 模板相关

- sga\_real\_templet(单目标进化算法模板 (实值编码))
- sga\_code\_templet(单目标进化算法模板 (二进制/格雷编码))
- sga\_permut\_templet(单目标进化算法模板 (排列编码))
- sga\_new\_real\_templet(改进的单目标进化算法模板 (实值编码))
- sga\_new\_code\_templet(改进的单目标进化算法模板 (二进制/格雷编码))
- sga\_new\_permut\_templet(改进的单目标进化算法模板 (排列编码))
- sga\_mpc\_real\_templet(基于多种群竞争进化单目标编程模板 (实值编码))
- sga\_mps\_real\_templet(基于多种群独立进化单目标编程模板 (实值编码))
- moea\_awGA\_templet(基于适应性权重法 (awGA) 的多目标优化进化算法模板)
- moea\_rwGA\_templet(基于随机权重法 (rwGA) 的多目标优化进化算法模板)
- moea\_nsga2\_templet(基于改进 NSGA-II 算法的多目标优化进化算法模板)
- moea\_q\_sorted\_new\_templet(改进的快速非支配排序法的多目标优化进化算法模板)
- moea\_q\_sorted\_templet(基于快速非支配排序法的多目标优化进化算法模板)