

数据可视化

本章介绍 Geatpy 的可视化功能。

当前 Geatpy 提供 3 个可视化输出的库函数: `trcplot`,`sgaplot` 和 `frontplot`。分别是单目标进化算法的可视化输出、单目标优化进化过程动画输出, 以及多目标进化算法的可视化输出。

值得注意的是, 在进行可视化输出之前, 最好在 Python 控制台执行以下语句:

matplotlib qt5

使得绘制的图形在窗口中显示, 否则动画效果就无法显示。

1. trcplot

该函数根据传入的进化追踪器来实现种群进化全过程中一些参数的可视化。比如绘制各代种群最优个体的目标函数值、适应度值、种群个体的解等。其语法如下:

```
trcplot(pop_trace, labels)
trcplot(pop_trace, labels, titles)
trcplot(pop_trace, labels, titles, save_path)
```

其中 `pop_trace` 是一个进化追踪器, 每一列代表一个参数, 如第一列代表个体最优目标函数值等, 每一行对应一个“代”, 比如第一行对应的是第一代种群的最优个体等等。

因为 `pop_trace` 包含许多信息, 我们可以根据这些信息绘制一张或多张图片。因此需要使用 `labels` 和 `titles` 来控制需要画多生张图、每张图包含多少个变量。

因此 `labels` 和 `titles` 都是 list 类型的列表。`labels` 指代了图片中有哪些变量, 这些变量的图例名称是什么。例如:

1. 假设 `pop_trace` 有 2 列, 含义分别是'a' 和'b', 则 `labels = [['a'], ['b']]`, 表示要画 2 张图, 每张图画 2 个变量, 图例分别是'a' 和'b'。

2. 假设 `pop_trace` 有 2 列, 含义分别是'a' 和'b', 则 `labels = [['a', 'b']]`, 表示要画 1 张图, 图中有 2 个变量, 图例分别是'a' 和'b'。

3. 假设 `pop_trace` 有 3 列, 含义分别是'a', 'b' 和'c', 则 `labels = [['a'], ['b', 'c']]`, 表示要画 2 张图, 第一张图有 1 个变量, 图例是'a'; 第二张图有 2 个变量, 图例是'b' 和'c'。

注: `labels` 的元素总数必须等于 `pop_trace` 的列数

`titles` 表示各图片的标题, 元素设为空字符串则表示不显示图片标题。

如 `titles = ['a', '']`, 表示有 2 张图片, 标题分别是 a 和不设标题

注: `len(titles)` 必须等于 `len(labels)`

案例:

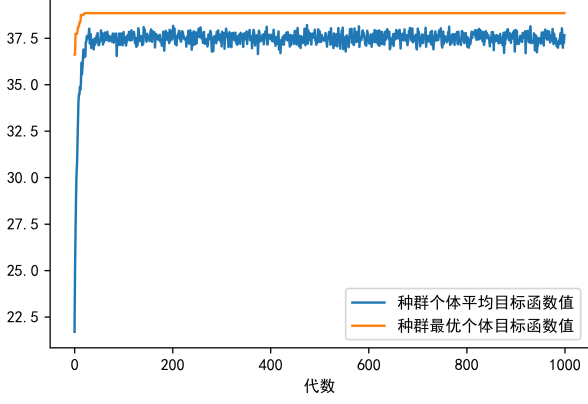
在 `sga_code_templat` 算法模板中, 有这样的一行代码:

```
ga.trcplot(pop_trace, [['种群个体平均目标函数值',
                        '种群最优个体目标函数值']])
```

其中 `pop_trace` 是一个 n 行 2 列的矩阵, 第一列代表种群个体平均目标函数值, 第二列代表种群最优个体目标函数值。因此传入 `trcplot` 绘图函数的参数 `labels` 设为 `[['种群个体平均目标函数值', '种群最优个体目标函数值']]`, 表示要画 1 张图, 这张图中同时绘制“种群个体平均目标函数值”以及“种群最优个体目标函数值”。

`titles` 参数是缺省的, 因此绘图将不显示标题。

绘图结果如下:



2. sgaplot

该函数根据传入的数据集和进化代数 `gen` 绘制动态图或进化结束后的静态图。其语法如下:

```
newAx = sgaplot(ValueSet, Label, saveFlag)
newAx = sgaplot(ValueSet, Label, saveFlag, ax)
newAx = sgaplot(ValueSet, Label, saveFlag, ax, gen)
newAx = sgaplot(ValueSet, Label, saveFlag, ax, gen, title)
newAx = sgaplot(ValueSet, Label, saveFlag, ax, gen, title, save_path)
```

其中 `ValueSet` 是一个 numpy 的 array 类型的列向量, 一般传入该函数前是进化记录器 `pop_trace` 的某一列数据。其实际含义由 `Label` 确定。

`Label` 是一个字符串, 代表数据集 `ValueSet` 的含义

`saveFlag` 是布尔类型的标记, 表示是否要保存图片。当要绘制动画时, 必须设为 `False`。

`ax` 是可选参数, 在绘制动画的时候需要传入。其代表上一帧的动画。当画第一帧时, 其值为 `None`。

`gen` 是可选参数, 表示当前进化代数, 默认为 `None`。该参数没有缺省或为非 `None` 时, 图片将绘制动态图。

`title` 是可选参数, 表示图形的标题名称。

`save_path` 是 string 类型的可选参数, 表示保存图片的路径。

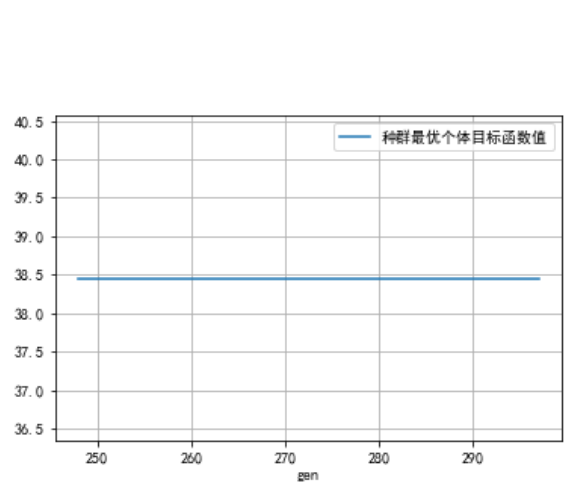
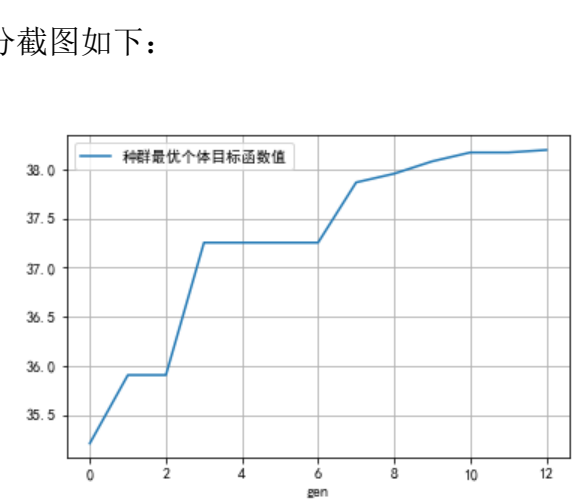
`newAx` 代表新的图形, 是更新后的 `ax`。

案例:

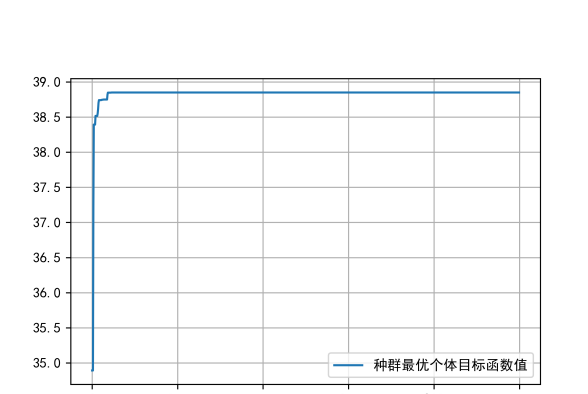
在解决一个单目标优化问题时绘制进化过程中的各代种群最优个体的目标函数值的变化动态图。并且在进化结束后绘制一个进化全过程的各代最优个体的目标函数值静态图。

```
...
ax = None # 存储上一帧图片
# 开始进化
for gen in range(MAXGEN)
...
# 进化操作
...
# 记录最优个体
bestIdx = np.max(FitnV)
pop_trace[gen,1] = ObjV[bestIdx] # 记录当代目标函数的最优值
...
# 绘图
ax = ga.sgaplot(pop_trace[:, [1]], '种群最优个体目标函数值',
False, ax, gen)
# 进化结束
ga.sgaplot(pop_trace[:, [1]], '种群最优个体目标函数值',
True) # 绘制最终的帕累托前沿图
# end
```

运行结果动画部分截图如下:



进化结束后绘制的静态图为:



3. frontplot

该函数根据帕累托最优集矩阵 `NDSet` 绘制 2 维或 3 维的目标函数值散点图。并且可以在种群进化过程中绘制动态的帕累托最优解的 2 维或 3 维动画。

语法:

```
newAx = frontplot(NDSet, saveFlag)
newAx = frontplot(NDSet, saveFlag, ax)
newAx = frontplot(NDSet, saveFlag, ax, gen)
newAx = frontplot(NDSet, saveFlag, ax, gen, title)
newAx = frontplot(NDSet, saveFlag, ax, gen, title, save_path)
```

其中 `NDSet` 为存储帕累托最优解的目标函数值矩阵, 一列对应一个目标函数值, 其列数必须为 2 或 3。

`saveFlag` 是布尔类型的标记, 表示是否要保存图片。当要绘制动画时, 必须设为 `False`。

`ax` 是可选参数, 在绘制动画的时候需要传入。其代表上一帧的动画。当画第一帧时, 其值为 `None`。

`gen` 是可选参数, 表示当前进化代数, 默认为 `None`。当该参数为非 `None` 时, 图片将在图例部分显示当前 `gen` 的值以及前沿点数。

`title` 是可选参数, 表示图形的标题名称。

`save_path` 是 string 类型的可选参数, 表示保存图片的路径。

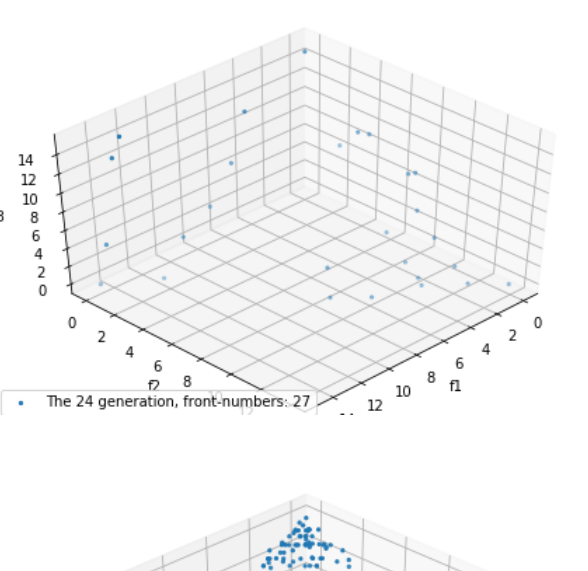
`newAx` 代表新的图形, 是更新后的 `ax`。

案例:

在解决 3 目标优化问题时绘制进化过程中的帕累托前沿的动画, 以观察搜索到的帕累托前沿的动态变化。并且在进化结束后绘制一个最终的帕累托前沿图。

```
...
ax = None
# 开始进化
for gen in range(MAXGEN)
...
# 进化操作
...
# 更新帕累托最优集NDSet
...
# 绘图
ax = ga.frontplot(NDSet, False, ax, gen + 1)
# 进化结束
ga.frontplot(NDSet, True) # 绘制最终的帕累托前沿图
# end
```

运行结果动画部分截图如下:



最终的前沿面静态图为:

