软件全称： 基于代理模型的遗传算法优化工具箱软件

软件简称： **saopy**

软件版本： **V1.0**

用 户 手 册

上海交通大学

目　　录

一 概况

（一）开发硬件环境

CPU: 2x10-core Intel Xeon e5-2680 v2
内存: 128GB
主板: ASUS Z9PE-D8

（二）运行硬件环境

CPU: 任意。推荐多核，例如 2x10-core Intel Xeon e5-2680 v2
内存：推荐>=16GB

（三）开发操作系统

Windows 7、Windows 10

（四）开发环境 / 开发工具

PyCharm

（五）运行平台 / 操作系统

Windows 7 或 Windows 10

（六）软件运行支撑环境 / 支持软件

numpy>=1.16.0
matplotlib>=3.0.0
scipy>=1.0.0
geatpy=2.2.3
pytorch>=1.3.1

（七）编程语言及版本

Python 3.6

（八）程序量

4700 行

（九）开发目的

自动实现整个基于代理模型和遗传算法的优化流程

（十）面向领域 / 行业

任何需要使用代理模型优化的领域，例如航空航天

（十一） 主要功能

自动实现整个基于代理模型和遗传算法的优化流程。在优化中能在样本密度较低和函数梯度变化较大的区域添加新的样本点，提高模型精度。同时提供一种动态调整加点个数的方法，在保证优化效果的同时大幅减少计算量。

（十二） 技术特点

使用面向对象的编程方法实现，各个模块之间相互独立，同时可以相互调用，方便代码的维护和使用。

二 系统简介

（一）概述

本优化工具箱可以自动实现整个基于代理模型和遗传算法的优化流程。在优化中能在样本密度较低和函数梯度变化较大的区域添加新的样本点，提高模型精度。同时提供一种动态调整加点个数的方法，在保证优化效果的同时大幅减少计算量。

（二）应用范围和对象

任何需要使用代理模型优化的领域，例如航空航天

（三）系统特色

使用面向对象的编程方法实现，各个模块之间相互独立，同时可以相互调用，方便代码的维护和使用。

（四）界面设计

目前 V1.0 版本没有 GUI 操作界面，直接通过在主程序(main.py)中修改参数进行需要的操作。后面版本可以增加 GUI 操作界面。

（五）主要功能简介

整个优化工具箱的流程图如图 1 所示，下面将详细介绍每个模块的功能。

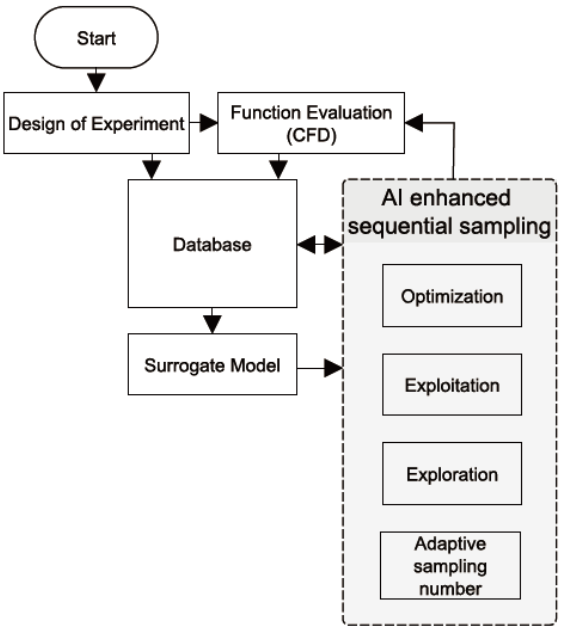

图 1 saopy 优化工具箱流程图

实验设计方法模块（Design of Experiment）用于生成初始样本点，目前提供三种生成样本点的方法：full factorial, random LHS, optimal LHS。模块有两个输入参数，分别是样本点个数和维数，根据所选择的生成样本点的方法，使用对应的 begin sampling 方法，生成归一化的样本点，然后根据给定的参数的范围，反归一化（inverse normalize）后得到最后的样本点 X，同时会生成.csv 文件，方便其他模块的调用。此外，还提供了样本点可视化的方法，使用 plot 方法，可以输出样本点的散点图，更加直观的显示样本点分布情况。

函数计算模块（Function Evaluation），包括标准测试函数模块（Benchmark Function），及 CFD 计算模块。此模块的输入为实验设计模块生成的样本点 X.csv，或者可以使用其他方法或者人工生成的样本点，然后使用 calculate 方法计算得出每个样本点对应的函数值 y，并生成.csv 文件，方便其他模块的调用。其中标准测试函数模块还具有画出标准函数图像的功能，方便和之后使用代理模型生成的响应面进行对比。CFD 计算模块中会调用 CATIA 生成模型，调用 Pointwise 画网格，调用 CFD++进行计算，这些功能都通过 Python 及各个软件的脚本实现自动化完成。用户可以模仿这个模块的方法定义自己想要的函数计算方法。

代理模型模块（Surrogate Model）一共提供三种代理模型方法：人工神经网络（ANN），径向基函数（RBF），和 Kriging。此模块的输入为样本点 X.csv 和通过函数计算模块得到的对应的函数值 y.csv。首先会对 X 和 y 进行归一化，然后根据选择的代理模型，使用对应的方法进行训练，训练完成后会将代理模型保存下来，方便其他模块调用。同时，此模块还具任意维数响应面的可视化功能，例如在画 5 维响应面时，任意选择两个参数作为变量，其余参数默认固定为参数范围的中间值（也可以指定特定值），这样参数两两组合，一共会画出 10 个响应面。另一个子模块是交叉验证模块（Cross Validation）。此模块的功能是将所有数据平均分成 n 份（n 为输入参数 fold number），选择 n-1 份作为训练集，1 份作为测试集，一共会做 n 次不同的训练，最后得到所有测试集的均方根误差（RMSE）的平均值，作为评判代理模型精度的一个标准。同时会输出误差最大的一个样本点，方便后续模块（Exploitation）调用。由于 n 次不同的训练之间互不影响，所以这些训练可以并行进行，此模块同样提供了这个功能，根据输入的并行进程数（parallel process number）并行训练，大大提高了运行速度。此外，此模块还提供两种将数据平均分成 n 份的方法，最简单的是随机分成 n 份（random），这个方法运行速度快，但是其缺点是，如果测试集的样本和训练集非常接近，那么得到的均

方根误差就会较小，反之，则会较大，不能非常好的反应出模型的真实性能。另一种方法是测试集样本均匀分布的方法（optimal testing data），这个方法的实现过程和实验设计模块中的 optimal LHS 一样，这样可以更好的反应模型的性能。最后 plot 功能可以将所有训练集和测试集的真值和代理模型预测值进行对比。

遗传算法优化模块（Optimization）基于现有的开源工具箱 Geatpy 建立。Geatpy 是一个高性能遗传算法工具箱，并提供一个高度模块化的面向对象的算法框架，可用于求解单目标优化、多目标优化等问题。此模块的功能是求解代理模型的最优值，同时也能求解具有解析式的测试函数的最优值，用于对比验证。它的输入是代理模型模块中保存下来的已经训练好的代理模型（surro model），遗传代数（generation），种群规模（population），和参数优化范围（X range）。交叉概率和变异概率使用默认值 0.7 和 0.025。优化完成后会保存最优值 y.csv 和对应的设计变量 X.csv（对于单目标优化会保存最优值；对于多目标优化会保存 Pareto front 上的一系列值，其总数为种群规模），方便其他模块调用。优化完成后，使用 plot 功能可以画出最优值随遗传代数收敛曲线，用于确定合适的遗传代数，由于代理模型或者测试函数的计算时间几乎可以忽略不计，所以推荐使用较大的遗传代数（默认使用 5000），保证优化的收敛。

Exploitation 的核心思想是，对于函数非线性程度较大的区域或者说梯度变化较大的区域，代理模型往往不能较好的进行拟合，需要在这些区域加入更多的样本点才能提高代理模型在这些区域的拟合精度。对于有解析表达式的函数，可以使用数值的方法求出梯度变化较大的区域，而实际问题是没有函数表达式的，无法求出梯度，所以需要换一种思路。由于代理模型在这些区域的拟合精度较低，可以通过 RMSE 反应出来，所以可以在 RMSE 较大的点的附近区域加点。Exploitation 模块首先将代理模型中 RMSE 最大的一个样本点和需要加点的个数作为输入，然后求出到这个样本点最近的一个样本点的距离，之后生成一个长度在这个距离范围内的随机向量，将 RMSE 最大的样本点加上这个向量就得到了这个样本点附近的随机的一个点，当这个点超出了参数的边界范围，就会舍弃它，重新生成一个，最后将这些点反归一化后输出为.csv 文件，方便其他模块调用。

Exploration 的核心思想是，对设计空间的各个区域给与同等的重视程度，尽可能使样本点均匀的充满整个设计空间，所以会在样本密度较低的区域加点。Exploration 模块与 Exploitation 的一个区别在于 Exploration 不需要代理模型作为输入，只需要样本点 X.csv 作为输入。Intersite distance 可以用来衡量设计空间中样本的密度，intersite distance 越大就代表样本密度越低。具体实现方法是将设计空间划分网格，计算每个网格节点到所有样本点的最小距离作为 intersite distance，当网格足够密时，就可以画出整个设计空间的 intersite distance 的云图。这里划分网格的方法类似于实验设计方法，目前提供两种方法 full factorial 和 random LHS，和实验设计模块中的一致。对于维数较高的问题，使用 full factorial 网格节点个数会成指数增加，需要大量计算时间，所以推荐使用 random LHS。在得到所有网格节点的 intersite distance 后，就可以选择 intersite distance 最大的网格节点，作为需要增加的新的样本点。由于增加新的样本点后，整个设计空间的 intersite distance 分布会发生变化，所以每加一个点后需要重新计算 intersite distance。最后同样将这些点反归一化后输出为.csv 文件，方便其他模块调用。

有了上述两种加点方法，再将遗传算法优化得到的最优的点也作为新的样本点加入进来，与原始的所有样本点组成新的数据库，重新建立代理模型，由此开始新的一轮优化，如此循环迭代多次，直到得到收敛的结果。这其中存在的一个问题是每次迭代过程中 Exploitation 和 Exploration 的加点个数是人为指定的，那么是否能够根据迭代过程中代理模型精度的改善和优化迭代收敛的情况来动态调整加点的个数呢？最终的目标，一是尽可能加快收敛速度，即减小迭代次数，因为每次迭代之间是串行的，相邻两次迭代生成的新的样本点只能串行计算

（每次迭代中的样本点可以并行计算），这样可以减少需要的总的优化时间。二是尽可能减少新加的样本点的个数，即减少所需增加的函数计算次数。动态加点模块（Adaptive sampling number）可以实现这两个目标，首先需要给定一个最大允许加点个数（最小加点个数默认为1），这个可以根据实际情况具有的预算和优化时间的限制来确定，通常来说，加点个数越多，收敛速度越快，优化时间越短，但是计算成本也越大。默认初始加点个数为（最大允许加点个数+1）/2。当前后两次迭代 best so far 有改进时，代表目前优化还没有收敛，需要加更多的点来加快收敛速度，以满足第一个目标，所以加点个数增加；当前后两次迭代 best so far 没有改进时，代表目前优化可能已经开始收敛，可以适当减小加点个数，以满足第二个目标。目前增加或者减少加点个数的速度是线性的，默认从初始加点个数开始，连续 5 次增加或者减少加点个数后达到最大或者最小加点个数。在得到加点个数之后，会调用 Optimization，Exploitation 和 Exploration 模块，得到 3 个模块各自所加的点，然后去掉重复的点，输出新加的样本点 new X.csv，方便其他模块调用。

三　系统安装说明

本软件目前是 Python 开源代码，可以使用任意 Python IDE 运行，推荐使用 PyCharm。需要安装的 Python 库，见（六）软件运行支撑环境 / 支持软件。

四　系统功能及操作步骤

（一）系统启动过程

demo 文件夹中为测试函数验证算例，main.py 为主程序。以 demo\single_obj_demo\ackley_2D_ANN\main.py 为例，将第一行 rootpath 修改成用户 saopy-master 文件夹所在路径，然后运行 main.py 就自动开始整个优化流程。

（二）系统的操作过程及界面

以 demo\single_obj_demo\ackley_2D_ANN\main.py 为例。lower_bound, upper_bound 为参数的下上界。plot_y_range 为函数图像或者代理模型响应面的 y 的范围和间隔。如果不知道具体 y 的范围，可以使用这种形式 plot_y_range = []，会自动根据 y 的最大和最小值画图。total_iter 为总迭代次数。

initial sampling plan 部分，number = 40 为初始样本点个数; dimension = 2 为维数。

function evaluation 部分为用初始样本点 X_new.csv 通过函数计算得到 y0_new.csv 的过程，这里是测试函数的例子，用户可以在这部分定义自己需要的函数计算方法。

get best ANN architecture with minimum RMSE 部分为获得最优 ANN architecture 部分，max_layers 为最大层数，max_neurons 为最大每层神经元个数（当前版本每层神经元个数相同），step 为神经元个数间隔，num_fold 为交叉验证份数，parallel_num 为并行计算进程数（用户可以根据自己电脑 cpu 性能修改该参数，推荐该参数和 cpu 核数相同）。获得最优 ANN architecture 后，会将这个 architecture 保存下来（best_ANN_arch），后面每次迭代都会使用这个 architecture。

cross validation and get max error point 为交叉验证和代理模型训练部分。parallel_num = 3 为并行计算进程数; num_fold = 3 为较差验证份数。

adaptive sequential sampling 部分为遗传算法优化，以及添加新的样本点部分。DOE_num=40 为初始样本点个数（要和 initial sampling plan 中的一致）,max_num=40 为最大每次迭代

Exploitation 和 Exploration 分别加点个数。number_optimized_X=5 就是遗传算法优化的种群规模。对于单目标优化，会将每次最优的点作为新的样本点添加到 database 中；对于多目标优化，会将 Pareto front 上所有的点作为新的样本点添加到 database 中，所有的点的个数就为 number_optimized_X，也就是种群规模。

f.plot，best_surro.plot，ass1.plot 为结果可视化功能，是非必要流程，用户在定义自己的问题时可以选择将这几行注释掉，不显示这些可视化结果。

（三）系统小结

这里以 demo\single_obj_demo\ackley_2D_ANN\main.py 为例说明运行流程，用户可以根据需要修改参数，以及在 function evaluation 中定义自己需要的函数计算方法。