

天线智能优化软件使用说明书

中国地质大学（武汉）

2020 年 1 月

目录

一、	引言.....	3
1.1	编写目的.....	3
1.2	软件用途.....	3
二、	软件运行环境配置.....	4
2.1	python 软件的安装.....	4
2.2	Ansys2018 环境配置.....	10
2.3	并行计算环境配置.....	10
三、	软件安装.....	11
四、	软件功能介绍.....	13
五、	软件操作说明.....	14
5.1	新建工程.....	14
5.2	工程设置.....	15
5.2.1	几何参数设置.....	17
5.2.2	算法参数设置.....	18
5.2.3	性能参数设置.....	20
5.3	工程运行.....	22
5.3.1	开始优化.....	23
5.3.2	算法输出.....	23
5.4	其它功能.....	27
5.4.1	视图显示.....	27
5.4.2	关联匹配.....	27
5.4.3	添加问题.....	27
5.4.4	添加算法.....	29
5.4.5	帮助.....	33

一、引言

1.1 编写目的

编写本使用说明书的目的是充分叙述本软件所能实现的功能及其运行环境，以便使用户了解本软件的使用范围和使用方法，并为软件的维护和更新提供必要的信息。

1.2 软件用途

本软件是一款基于演化优化算法和电磁仿真软件 Ansys 的天线优化设计软件。本软件建立了一个统一的多目标多约束天线优化问题模型，并可通过多目标演化算法同时优化多个目标，可自由选择优化问题约束和目标个数和类型。用户可根据需求灵活的进行优化目标设置，并可根据多目标优化结果，从最优解集中灵活的选择适合于工程应用的天线模型进行实际制作。本软件具有简单易操作的显示界面，通过界面能够方便的建立天线优化工程，构建天线优化问题，。此外，本软件还具有灵活的算法扩展和问题扩展模块，用户可轻松实现向软件中添加新的天线问题模型和优化算法。

二、软件运行环境配置

本软件底层算法部分利用 python 编写，在进行个体评估时还需调用外部电磁仿真软件 Ansys。因此，在利用本软件进行天线优化设计时，所在计算机上还需安装有 python 解释器和 Ansys 电磁仿真软件。Python 版本需为 python2(暂不支持 python3 版本)，Ansys 版本为 HFSS18.0。

2.1 python 软件的安装

第一步：双击运行 python 安装包中的安装程序 python-2.7.3.msi，如图 1.1.1-1.1.3。

名称	修改日期	类型	大小
Python27	2019/12/25 10:46	文件夹	
python-2.7.3.msi	2005/1/22 0:00	Windows Install...	15,496 KB

图 2.1.1 python 安装包

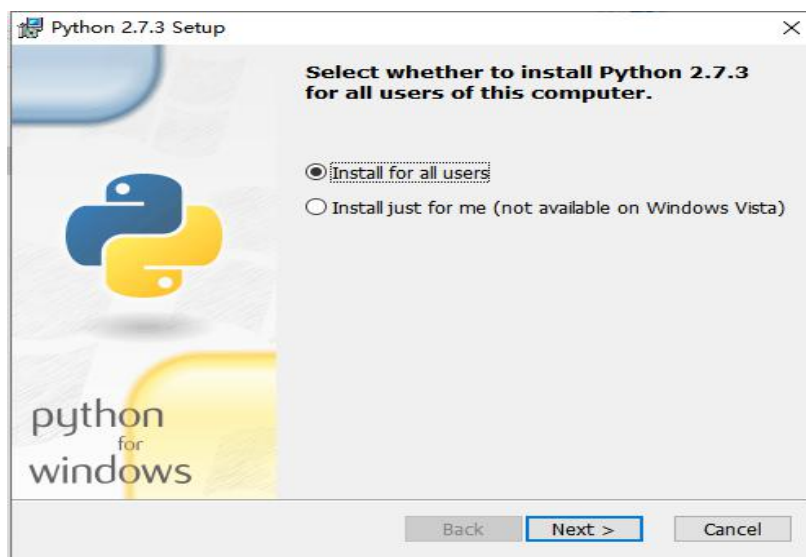


图 2.1.2 python 安装步骤 1

点击 next

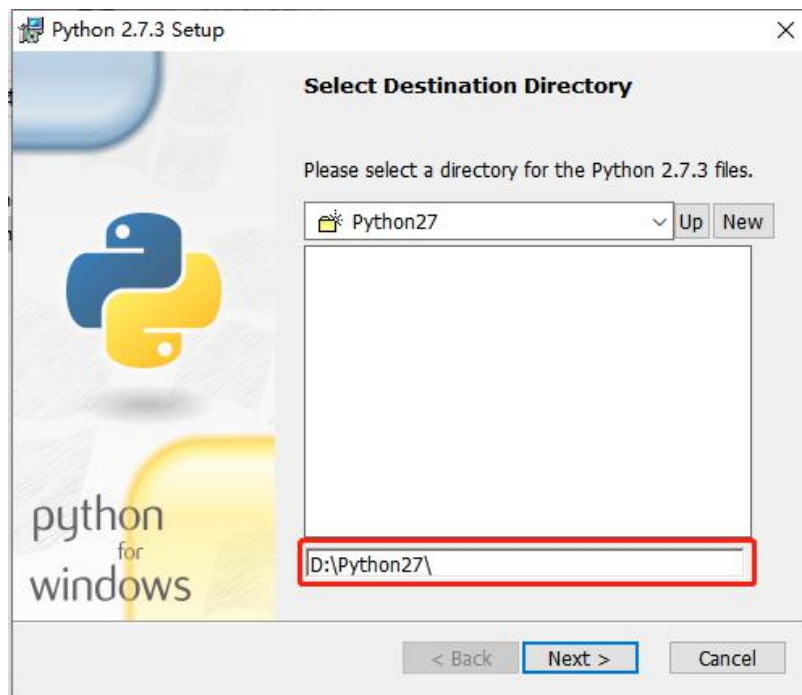


图 2.1.3 python 安装步骤 2

此处可选择软件的安装目录，然后根据提示点击 next 直至安装完成。

第二步：将 python 安装包下 Python27 目录中的文件拷贝至上一步 python 软件的安装目录。

名称	修改日期	类型	大小
Python27	2019/12/25 10:46	文件夹	
python-2.7.3.msi	2005/1/22 0:00	Windows Install...	15,496 KB

图 2.1.4 python 安装步骤 3

名称	修改日期	类型	大小
DLLs	2019/12/25 10:43	文件夹	
Doc	2019/12/25 10:43	文件夹	
include	2019/12/25 10:43	文件夹	
Lib	2019/12/25 10:45	文件夹	
libs	2019/12/25 10:45	文件夹	
Scripts	2019/12/25 10:45	文件夹	
tcl	2019/12/25 10:46	文件夹	
Tools	2019/12/25 10:46	文件夹	
LICENSE.txt	2012/4/10 23:34	文本文档	40 KB
matplotlib-wininst.log	2014/3/29 11:36	文本文档	107 KB
NEWS.txt	2012/4/10 23:18	文本文档	304 KB
numpy-wininst.log	2014/3/29 11:22	文本文档	95 KB
python.exe	2012/4/10 23:31	应用程序	26 KB
pythonw.exe	2012/4/10 23:31	应用程序	27 KB
README.txt	2012/4/10 23:18	文本文档	54 KB
Removematplotlib.exe	2014/3/29 11:36	应用程序	192 KB
Removenumpy.exe	2014/3/29 11:22	应用程序	192 KB
Removescipy.exe	2014/3/29 11:36	应用程序	192 KB
scipy-wininst.log	2014/3/29 11:36	文本文档	161 KB
w9xpopen.exe	2012/4/10 23:31	应用程序	49 KB

全部复制

图 2.1.5 python 安装步骤 4

名称	修改日期	类型	大小
DLLs	2019/12/25 11:02	文件夹	
Doc	2019/12/25 11:02	文件夹	
include	2019/12/25 11:02	文件夹	
Lib	2019/12/25 11:02	文件夹	
libs	2019/12/25 11:02	文件夹	
Scripts	2019/12/15 21:56	文件夹	
tcl	2019/12/25 11:02	文件夹	
Tools	2019/12/25 11:02	文件夹	
LICENSE.txt	2012/4/10 23:34	文本文档	40 KB
matplotlib-wininst.log	2014/3/29 11:36	文本文档	107 KB
NEWS.txt	2012/4/10 23:18	文本文档	304 KB
numpy-wininst.log	2014/3/29 11:22	文本文档	95 KB
python.exe	2012/4/10 23:31	应用程序	26 KB
pythonw.exe	2012/4/10 23:31	应用程序	27 KB
README.txt	2012/4/10 23:18	文本文档	54 KB
Removematplotlib.exe	2014/3/29 11:36	应用程序	192 KB
Removenumpy.exe	2014/3/29 11:22	应用程序	192 KB
Removescipy.exe	2014/3/29 11:36	应用程序	192 KB
scipy-wininst.log	2014/3/29 11:36	文本文档	161 KB
w9xpopen.exe	2012/4/10 23:31	应用程序	49 KB

此处为python安装目录

图 2.1.6 python 安装步骤 5

将上一步复制的文件拷贝到 python 安装目录，替换文件夹下已有的文件。

第三步：添加 python 安装路径到系统环境变量。

在 python 安装目录中，右击 python.exe 文件，拷贝文件所在位置路径。

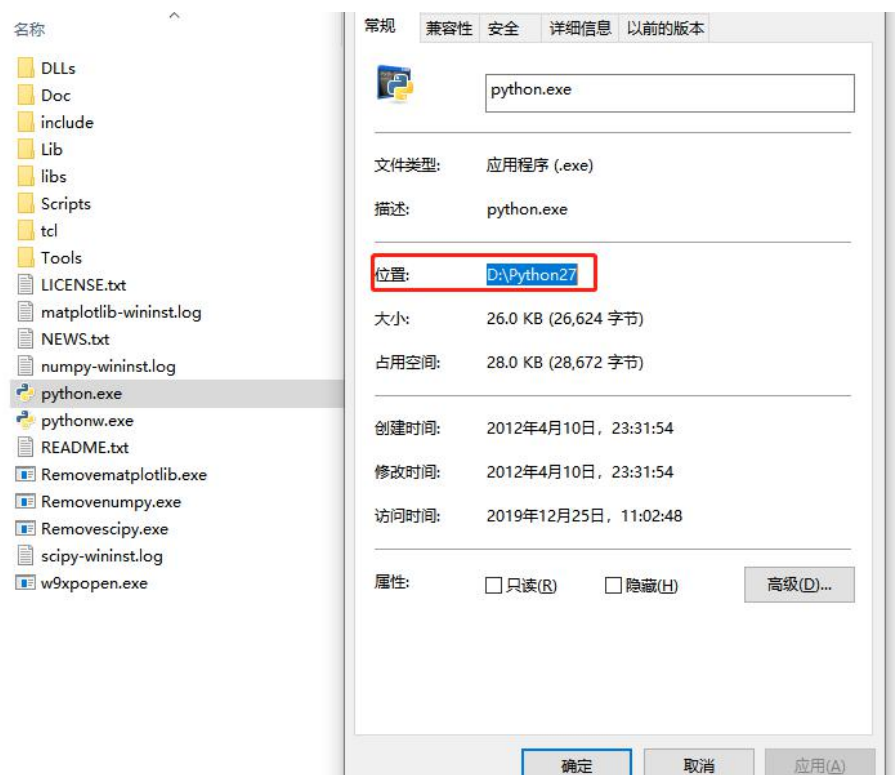


图 2.1.7 添加 python 环境变量步骤 1

然后右击“我的电脑”，点击“属性”，进入系统控制面板主页。



图 2.1.8 添加 python 环境变量步骤 2

点击高级系统设置

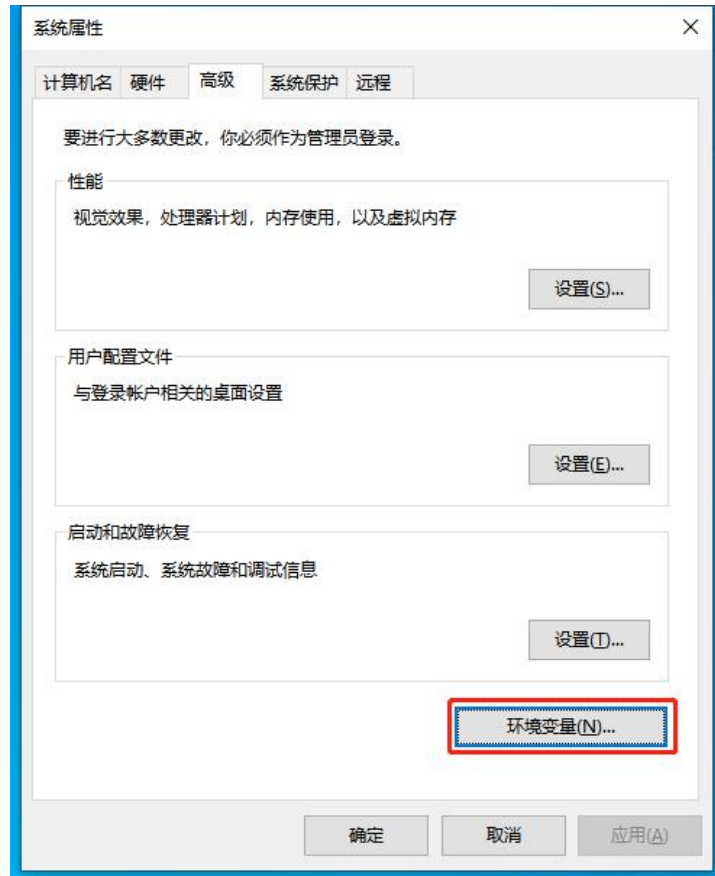


图 2.1.9 添加 python 环境变量步骤 3

点击环境变量

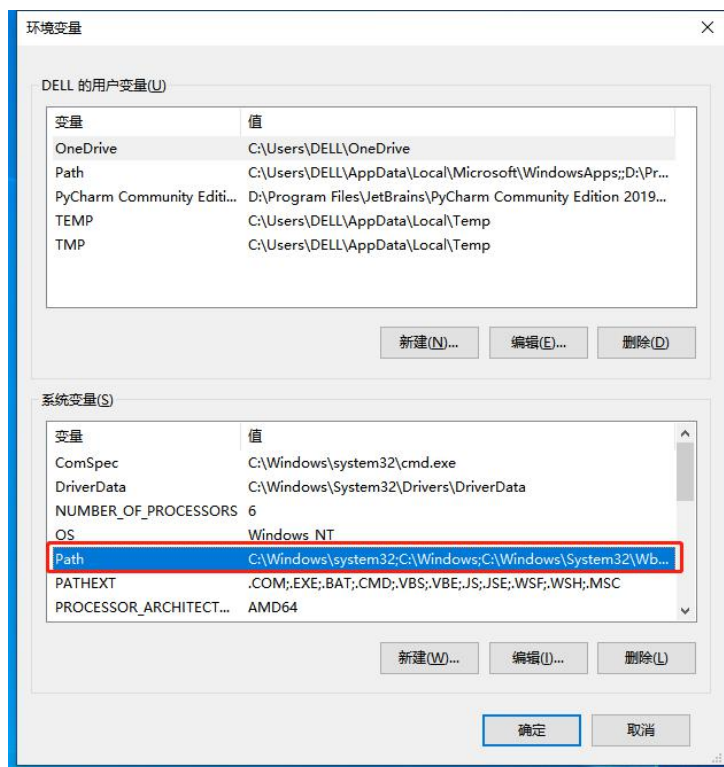


图 2.1.10 添加 python 环境变量步骤 4

在系统变量中双击 Path

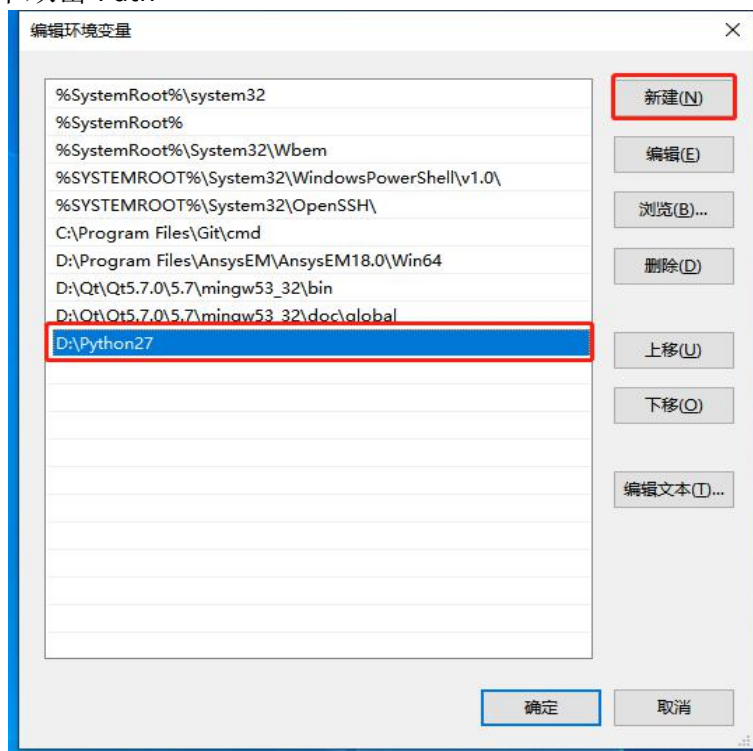


图 2.1.11 添加 python 环境变量步骤 5

点击新建，写入 python 的安装路径，点击确认。至此，python 运行的环境配置完成。

第四步：python 环境配置检查。可通过一下步骤检查配置是否成功。同时按下“win+R”，出现如下运行窗口：

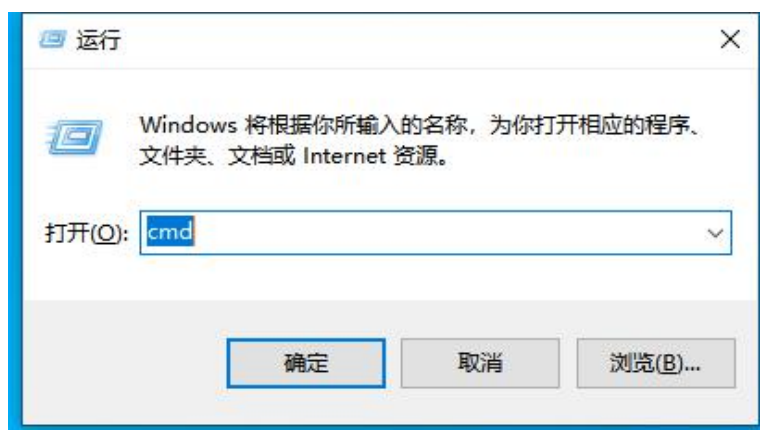


图 2.1.12 检查 python 环境变量步骤 1

输入 cmd，点击确定，进入系统命令行窗口。输入 python，若返回如下相关信息，则说明 python 环境已配置成功。

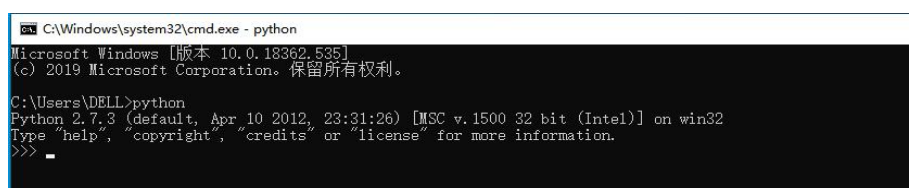


图 2.1.13 检查 python 环境变量步骤 2

2.2 Ansys2018 环境配置

安装破解后仿照 python 方式添加环境变量。

2.3 并行计算环境配置

本软件支持并行计算，并行计算的实现是利用 MPI 进行多进程并行计算，软件采用的 MPI 版本是 MPICH2，因此，要实现并行计算，需在电脑上配置 MPICH2 并行计算环境。

三、 软件安装

双击运行软件安装包 AntennaDO-setup.exe 文件，按照提示信息操作即可完成安装。

选择软件安装目录

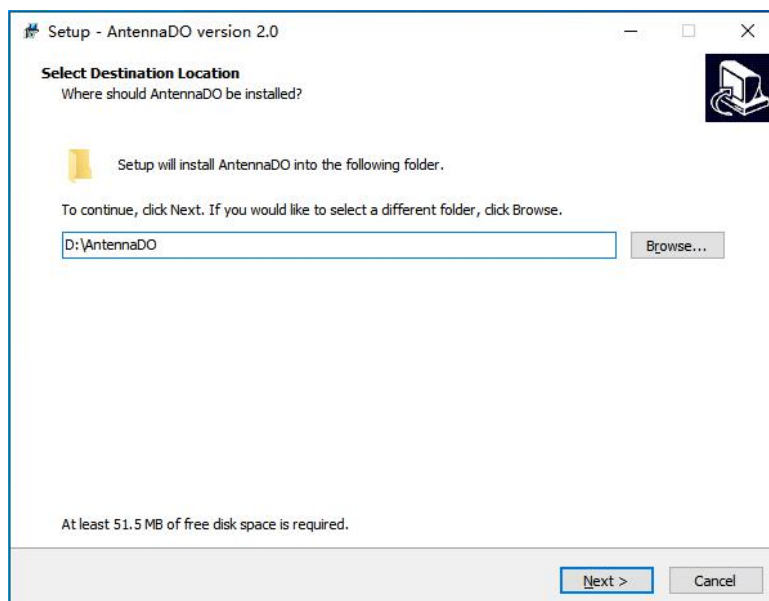


图 3.1 软件安装步骤 1

选择是否创建桌面快捷方式，默认不勾选

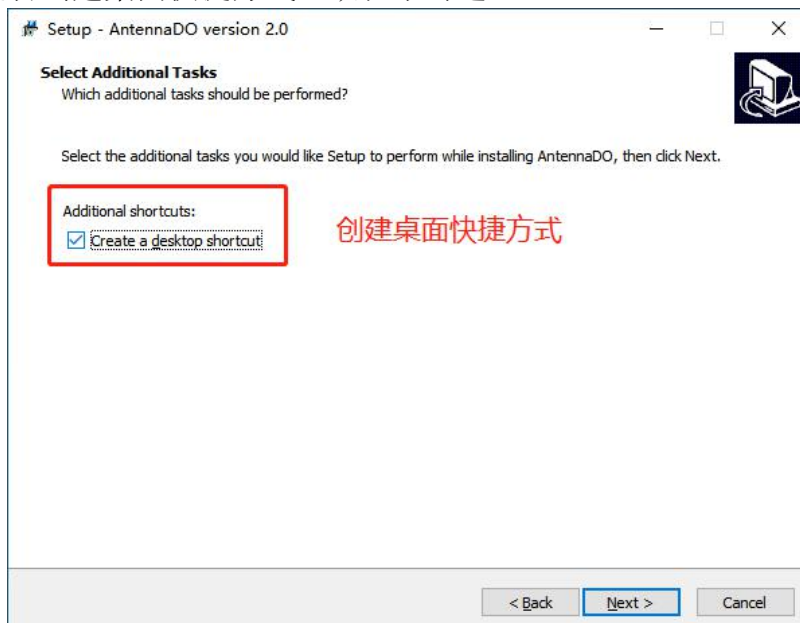


图 3.2 软件安装步骤 2

安装，点击 Install，等待安装完成。

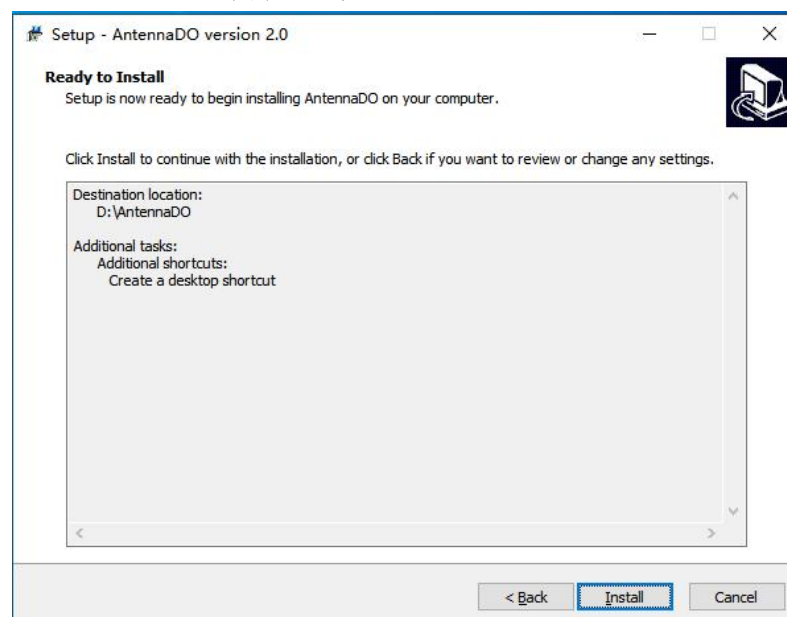


图 3.3 软件安装步骤 3

安装完成，是否立即启动软件，默认勾选

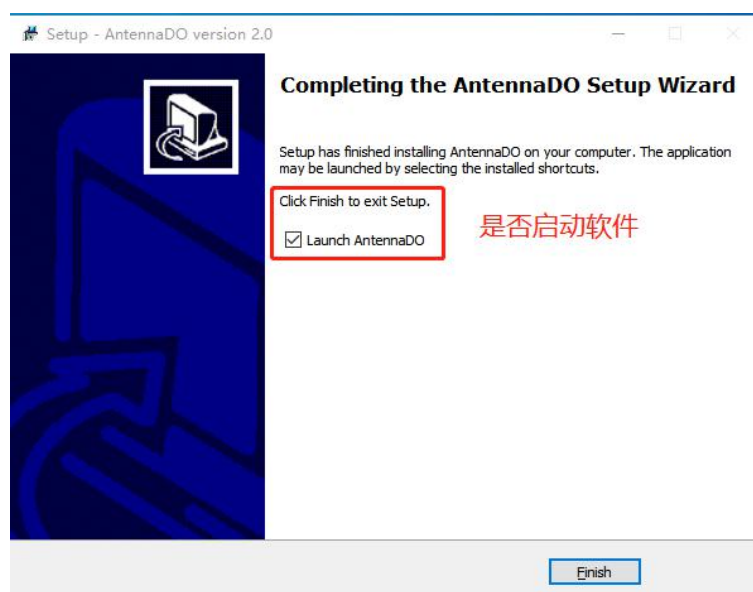


图 3.4 软件安装步骤 4

四、软件功能介绍

运行软件之后，会出现如图所示界面。

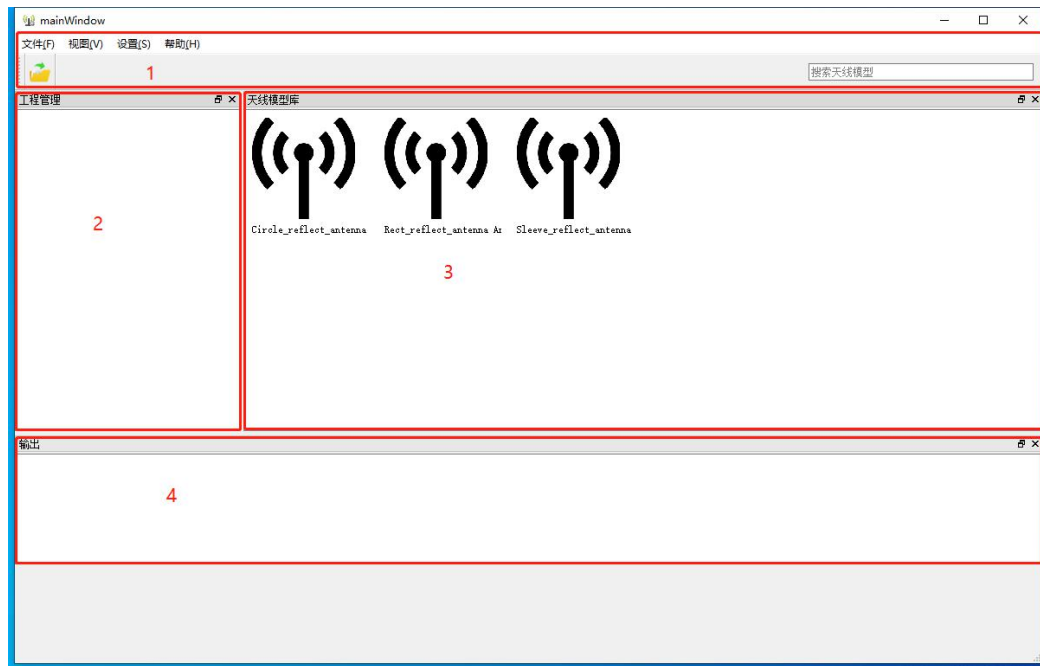


图 4.1 软件主界面

整个界面主要包含四个部分。

1、菜单栏部分。此处主要是软件的一些基本操作选项包含四项，分别是文件（用于打开工程文件）、视图（用于设置界面显示布局）、设置（用于设置算法和问题间的关联以及添加天线问题和优化算法）和帮助（打开软件的说明文档，以获取帮助信息）。

2、项目管理。此处显示天线优化工程建立之后，与该工程相关的一些设置。

3、天线模型库。此处显示的是软件天线库中已添加的天线问题模型，通过点击其中的问题模型图标可建立相应的天线优化工程。

4、输出。在工程运行时，此处将以文字的形式输出优化进度信息。此外，优化过程中，软件会在另一个窗口中以图形方式输出优化的收敛情况。

五、 软件操作说明

5.1 新建工程

(1)、找到所需优化的天线模型，可通过两种方式查找：a、当天线模型库中已添加的模型数量较少时，可直接在天线模型库界面中找到对应的天线模型。b、当天线模型库中已存在较多天线模型时，可在天线模型库右上方的搜索栏中输入天线问题名称来搜索相应的模型。此处为模糊搜索，当在搜索栏中输入关键字后，天线模型库中将只显示名称包含该关键字的天线模型。例如：建立一个天线模型为 Circle_reflect_antenna 的优化工程，我们可直接在显示界面中找到该模型，如图。

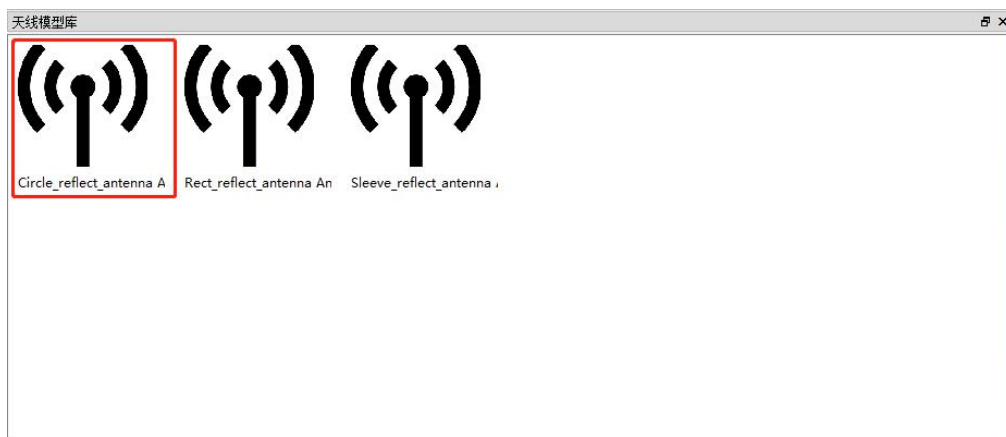


图 5.1.1 天线模型库界面
或在搜索栏中输入问题模型名称关键字，结果如图

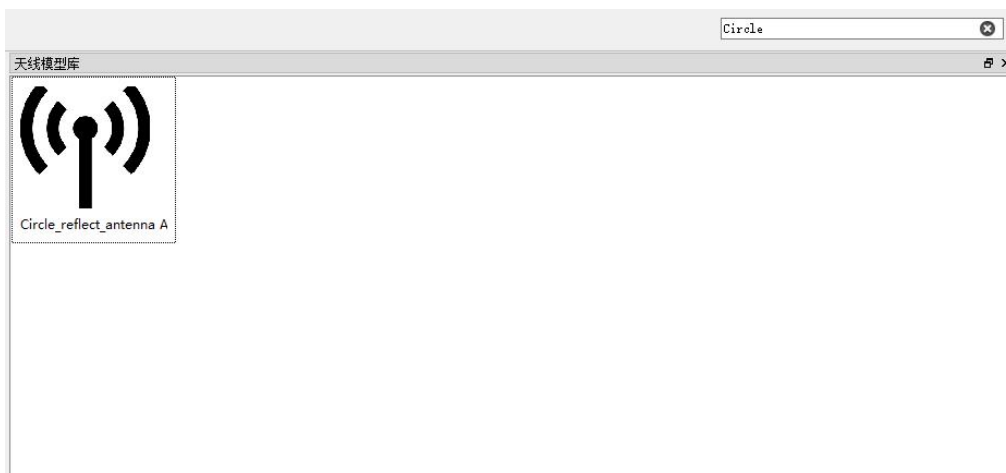


图 5.1.2 通过搜索栏寻找天线模型

(2)、建立天线优化工程。双击问题模型图标，会出现如下图所示菜单选项，点击“新建工程”即可进入新建工程下一步。按照提示点击下一步，直到出现如下窗口。在此处分别设置工程名称和工程所在路径，以及是否将当前工程路径设为默认路径，若选择，则下次新建工程时，新的工程文件将默认存储在此路径下。

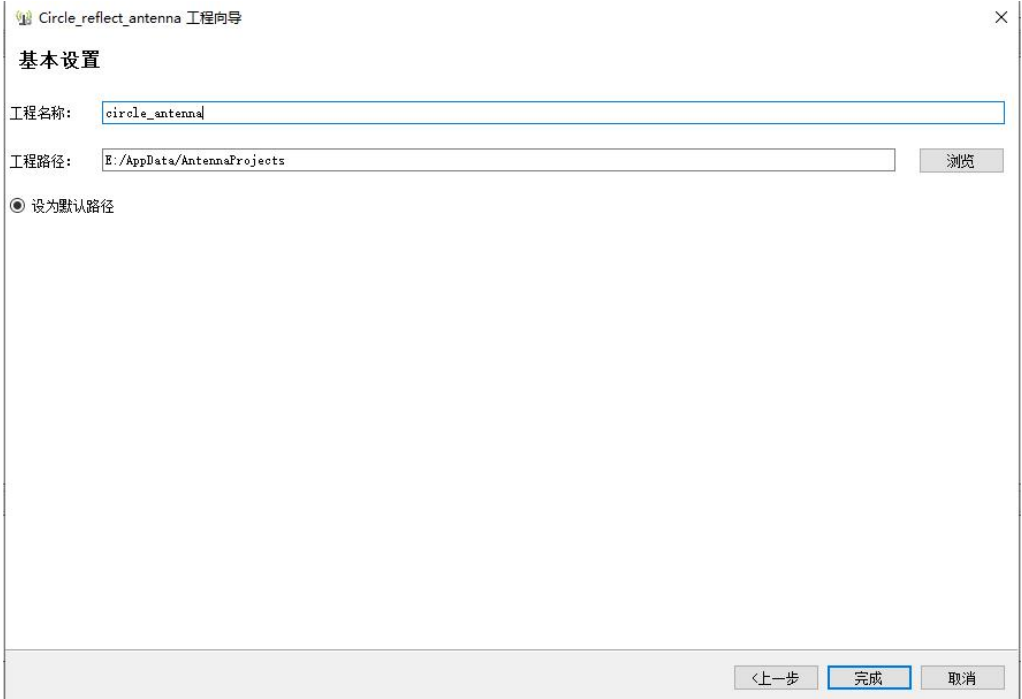


图 5.1.3 新建工程名称和路径设置

5.2 工程设置

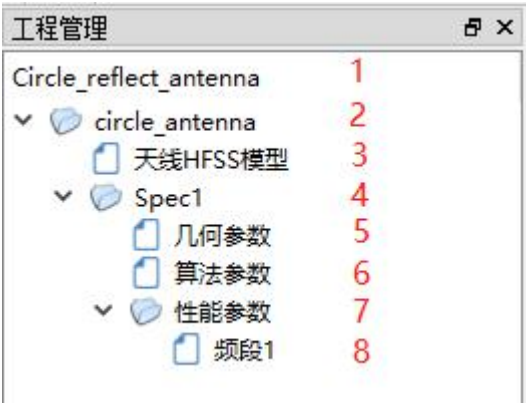


图 5.2.1 工程管理列表

在建立好优化工程之后，在工程管理栏会出现如图所示的树形工程管理信息。图中各部分的含义如表 5-2-1。

表 5-2-1 工程管理列表各项含义

序号	含义
1	天线模型名称
2	优化工程名称
3	双击可打开天线模型对应的 hfss 模型
4	优化问题的案例名称，右击可进行删除或复制操作。
5	天线的几何参数设置，即待优化天线结果变量。双击可进行相应参数的设置和修改。
6	优化算法相关的参数设置，双击可进行相应参数的设置和修改。
7	天线优化时的性能参数设置，其下可包含多个频段，每个频段对应一组参数设置，右击可增加频段设置。
8	某一频段对应的天线优化性能参数设置。双击可进行相应参数的设置和修改。

除了通过新建工程来打开工程管理项，本软件还可通过打开已有的工程文件新建打开。点击菜单栏中的打开文件图标，如图所示

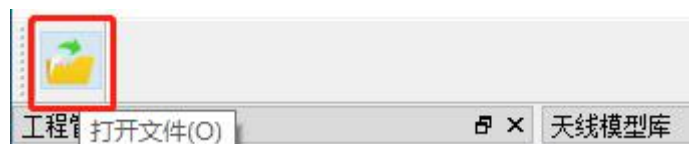


图 5.2.2 打开工程文件步骤 1

或者按下“Ctrl+O”快捷键，进入已建立的工程目录，找到后缀为.rel 的文件，选择并打开。

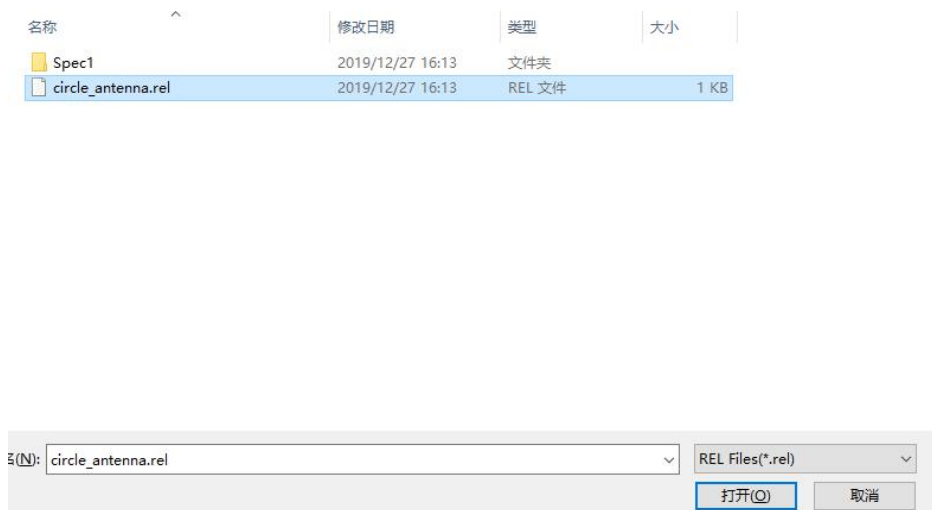


图 5.2.3 打开工程文件步骤 2

在打开天线优化工程后，开始进行天线优化之前，需要进行相应的参数设置，一般包括天线几何结构参数设置、优化算法参数设置和天线性能参数设置。

5.2.1 几何参数设置

双击几何参数，将出现如图所示界面。

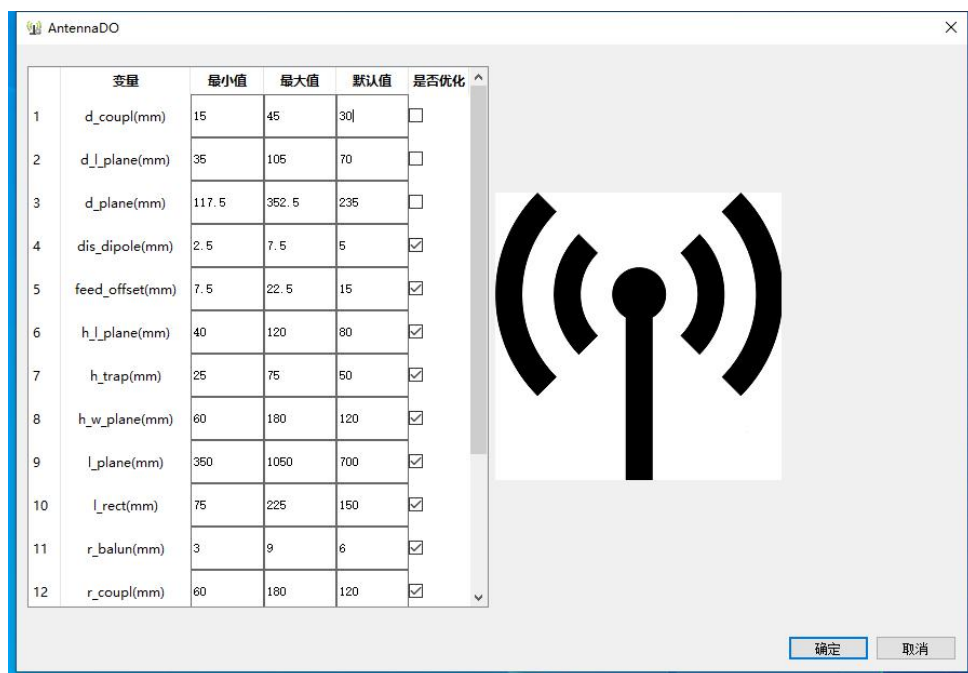


图 5.2.4 天线几何参数设置界面

其中变量一栏表示所有可供优化的天线结构参数，括号中为参数的单位。最小值

和最大值表示变量优化的范围区间，用户可手动输入数值设置。默认值表示，当某个变量不优化时，将取此值作为变量的固定值。最后一栏为是否优化选项，用户可根据需求勾选需要优化的变量。在对结构参数设置完成之后，点击确定按钮，可保存设置内容。

（注意：在构建 HFSS 天线模型时，对于长度相关变量，需统一单位，建议用 mm）

5.2.2 算法参数设置

在进行算法参数设置之前，需要先对问题和和算法进行关联匹配。操作如下。点击设置中的关联匹配，如图

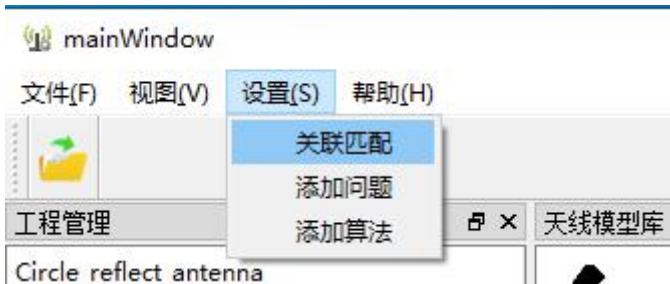


图 5.2.5 关联匹配设置步骤 1

出现如下图所示关联匹配设置界面

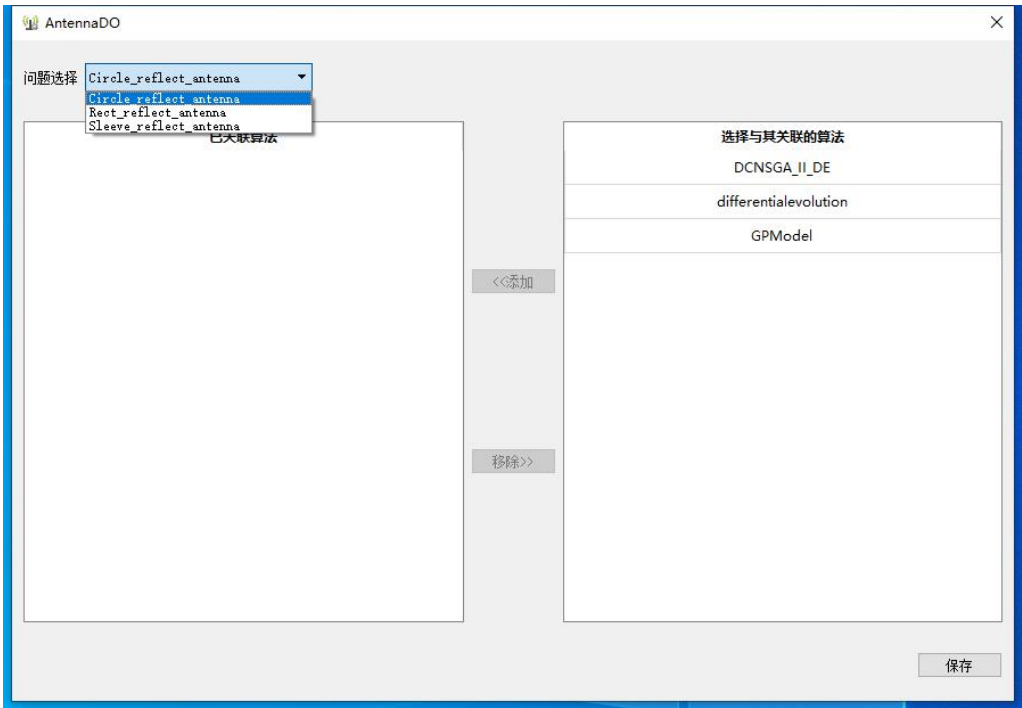


图 5.2.6 关联匹配设置步骤 2

点击问题选择栏的下拉列表，可在列表中选择天线模板库中存在的天线问题。界面左侧已关联算法中显示当前问题已关联的算法（只有与问题关联的算法才能用于求解该问题），界面右侧选择与其关联的算法中显示了当前软件算法库中存在的优化算法。点击选中需要关联的算法，然后点击中间的添加按钮，可将该算法

关联到对应问题。若需要将问题已关联的算法取消关联，在左侧已关联算法中点击选中该算法，再点击中间的移除按钮，便可取消关联。问题算法关联设置完成之后，点击保存，即可保存设置。例如，对 Circle_reflect_antenna 问题关联 DCNSGA_II_DE、和 differentialevolution 两种算法，设置结果如图所示

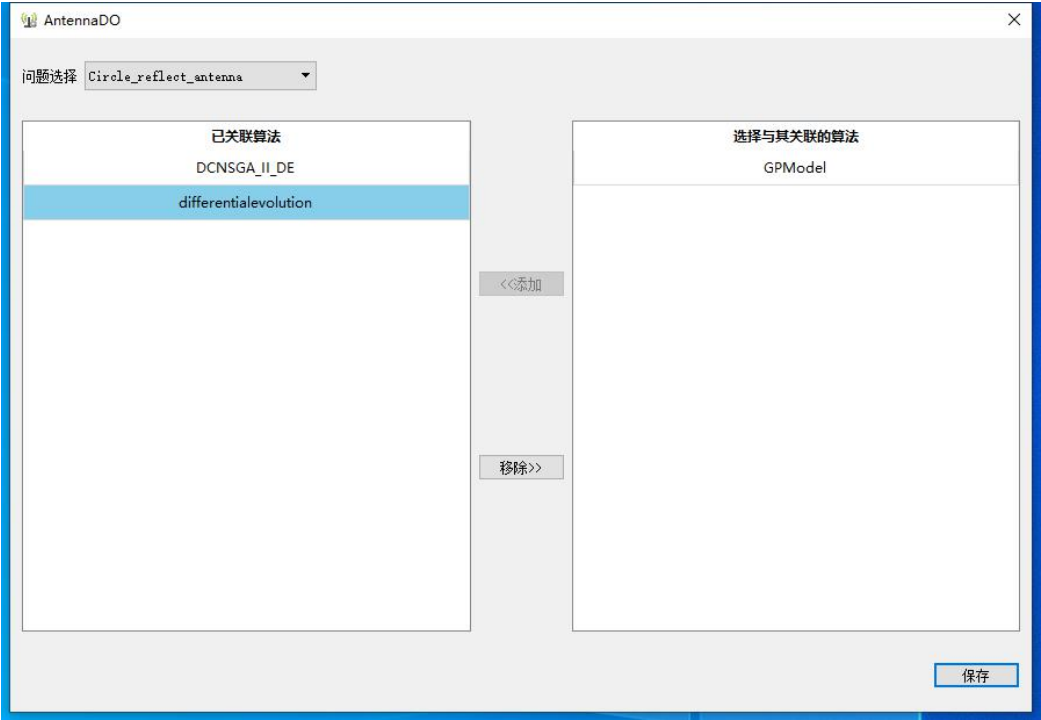


图 5.2.7 关联匹配设置步骤 3

设置好问题和算法的关联匹配后，双击主界面【工程管理】栏下的算法参数，将出现如图所示参数设置界面。

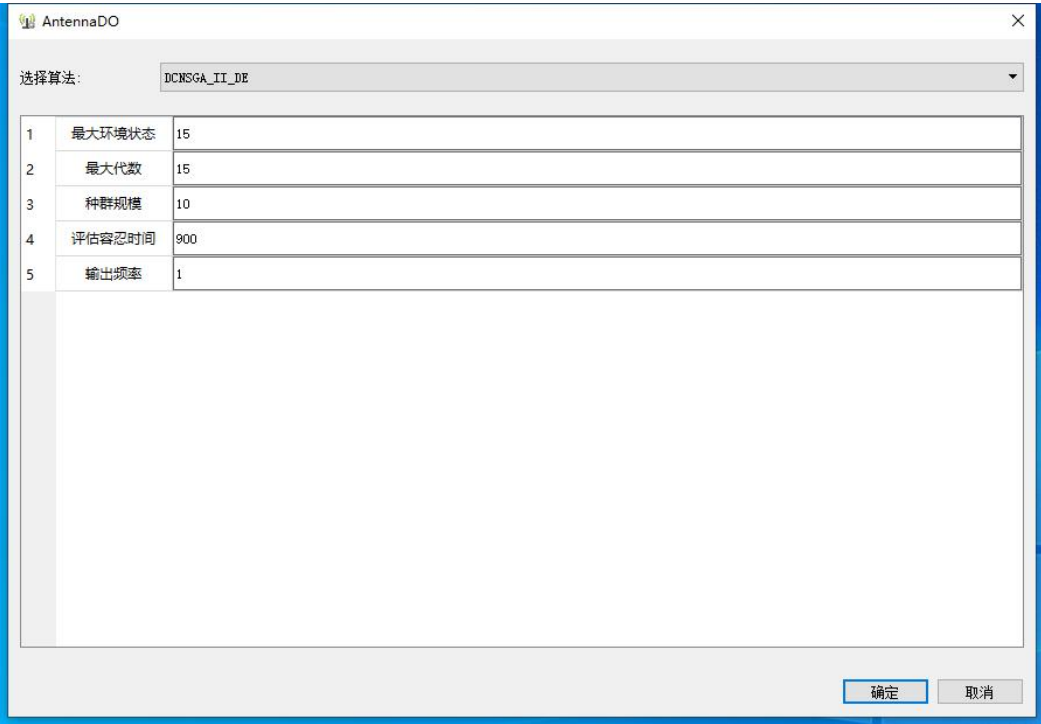


图 5.2.8 算法参数设置

在【算法选择】栏可从已关联的算法中选择要进行优化求解的算法，点击下拉列表可进行选择，默认为问题关联的第一个算法。界面下半部分列表中为算法的参数设置，均可进行手动输入编辑，且内容不能为空。其中最后两行“评估容忍时间”和“输出频率”是所有算法都有的参数。“评估容忍时间”表示单个个体评估容忍时间，单位为秒，当天线仿真时间超过这个时间，该个体被视为无效解。“输出频率”表示算法输出优化进度间隔的代数。如输出频率设置为 5，则每隔 5 代，在界面输出栏将输出当前的优化信息。

5.2.3 性能参数设置

双击【频段】栏，会出现如下界面。性能参数设置分为三个部分，分别为仿真计算范围，约束设置和目标设置。

天线仿真参数设置

仿真计算范围设置

约束设置

目标设置

频率范围设置

	频段下限 (MHz)	频段上限 (MHz)	频点个数	扫频方式	极化方式
1	300	600	3	等差	总增益

辐射范围设置

	起始 θ	终止 θ	θ 步长	起始 φ	终止 φ	φ 步长
1	-180	180	90	0	180	90

确认

确定

取消

图 5.2.9 天线仿真计算范围设置界面

1) 仿真计算范围设置

其中包含两部分：频率范围设置和辐射范围设置。
在频率范围设置中，可在频段上下限和频点个数栏中直接输入数字来进行设

置。在扫频方式中可选择等差或等比来设置扫频范围中的频点取样方式。在极化方式中可选择总增益、左旋、右旋、水平或垂直来设置增益的计算方式。在辐射范围设置中，可手动输入电磁仿真需要计算的远场辐射的角度范围，其中角度步长输入值必须为正。

点击确认按钮后，确认仿真计算范围。**注意：只有确认仿真计算范围之后，才能对后面约束和目标中与辐射范围相关的轴比、增益进行设置。**

2) 约束设置

约束设置界面如图，分为四部分：增益设置、轴比设置、回拨损失设置和结构参数设置。在未点击仿真计算范围设置中的“确认”按钮之前，增益设置和轴比设置是不可编辑的。因为回拨损失与辐射范围无关，所以该项无需确认仿真计算范围即可进行编辑。

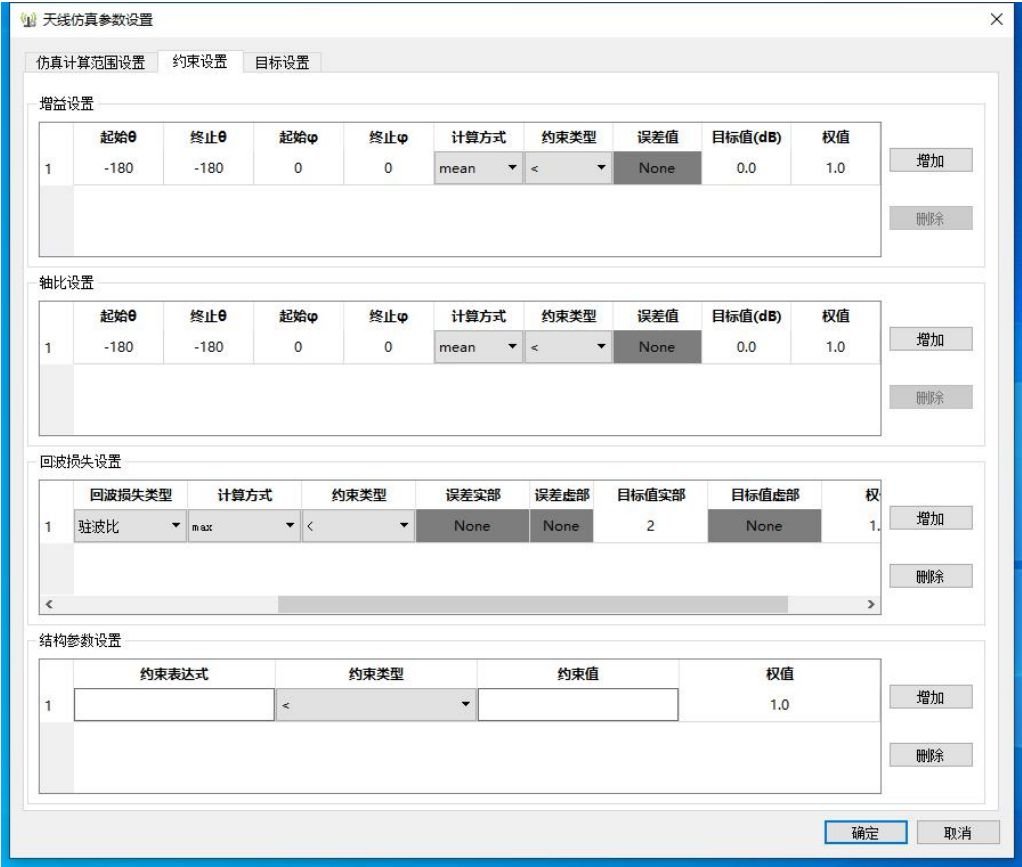


图 5.2.10 约束设置界面

其中增益和轴比设置方式相同。前四栏为计算的辐射范围设置，可直接输入数值进行设置，注意输入范围值必须在仿真计算范围设置中设置的辐射范围内。点击输入栏右侧的下拉符号，会提示出可供选择的数值，如图所示，若输入其他数值，将会在界面下方提示错误。



图 5.2.10 约束设置中的增益设置

在计算方式中，可选择进行约束计算的方式。可供选择的方式有 mean、max、min 和 rms。分别表示利用所设置范围内的平均值、最大值、最小值和均方根值来计算目标值。在约束类型中，有>、<、delta 可供选择，其中 delta 表示不等式约束。只有当约束类型设为 delta 时，才会用到后面的误差值，误差值表示进行等式约束时，允许的误差范围。目标值可直接手动输入，其单位为 dB，表示约束需要满足的目标。当有多个约束设置时，可根据各约束的重要程度设置相应的权值。

回拨损失设置中无辐射计算范围的角度设置。第一列为序号，1_1 表示端口 1 的第一条设置（由于目前只支持单端口，所以此处最多只有一条设置）。Z0 实部和 Z0 虚部分别表示需要匹配的端口阻抗的实部值和虚部值。在回拨损失类型中，可选择计算回波损失的类型。包括驻波比、S11(dB)、和阻抗，其中 S11 的单位是 dB。

结构参数用于设置对天线结构参数的约束。可在约束表达式栏中输入要进行约束的表达式（注意：表达式中的参数必须是在【几何参数】设置中被选中进行优化的参数）。提示：在设置该项约束时可双击【天线 HFSS 模型】栏，打开天线模板的 HFSS 模型查看天线结构，或双击【几何参数】栏，打开结构参数设置，查看已选择的待优化结构参数。

3) 目标设置

此部分为算法的优化目标设置，可进行多目标设置，在目标设置中至少需设置一个优化目标。（注意：当设置多个优化目标时，需选择多目标算法进行求解。）此部分的设置内容与设置方式与约束设置类似，不同的是在目标设置中，最后一栏为约束类型，其中包含“max”和“min”两个选项。“max”表示目标最大化优化，“min”表示目标最小化优化。此外，在目标设置界面最下端，是鲁棒性目标设置。该目标表示天线在各频点、角度上辐射特性的稳定性。其辐射范围设置默认与前面设置的辐射范围设置相同。用户可根据需求勾选是否优化此目标。

5.3 工程运行

5.3.1 开始优化

在界面左侧【工程管理】栏下右击当前工程的案例名，在出现的菜单栏中点击运行，即可开启工程运行。如图所示

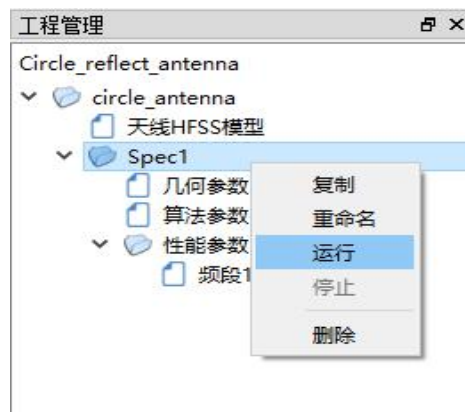


图 5.3.1 启动运行

5.3.2 算法输出

1) 界面输出

算法开始运行后，在主界面的下端的【输出】栏中将输出算法运行信息。如图

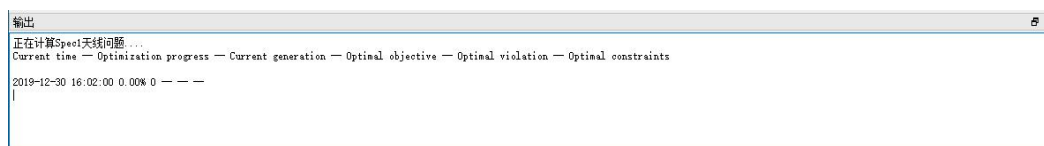


图 5.3.2 算法运行时主界面输出内容

当采用单目标算法优化时，其输出格式为：

“当前时间—优化进度—当前优化代数—当前代最优目标值—当前代最优个体的违约值—当前代最优个体的约束值”

当采用多目标算法优化时，其输出格式为：

“当前时间—优化进度—当前优化代数—当前环境变量”

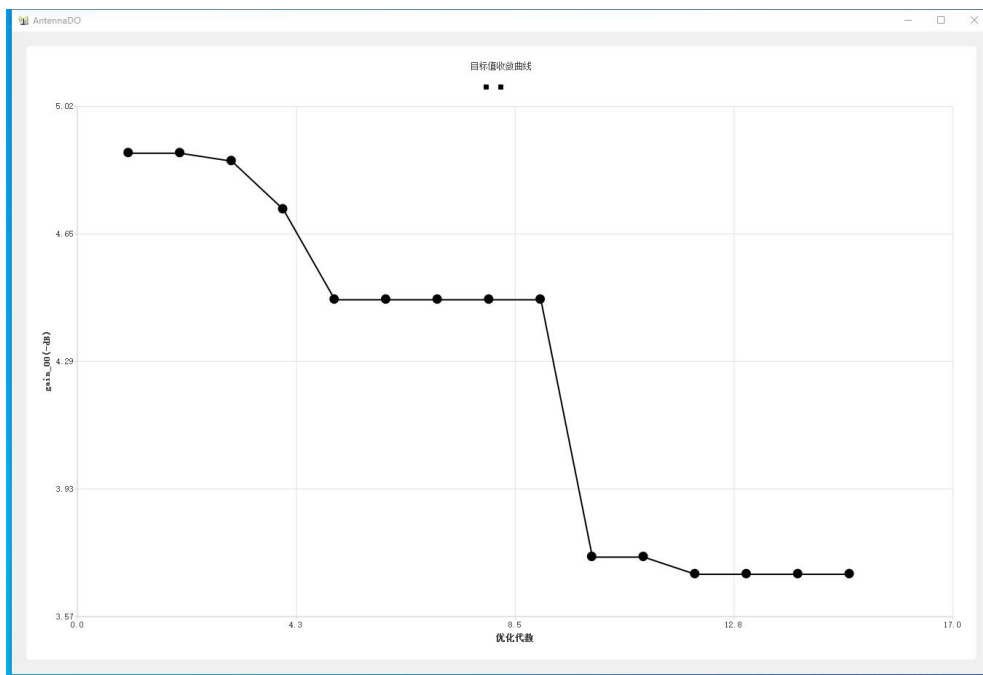


图 5.3.3 算法运行时两目标结果可视化输出

在进行单目标优化时，每一代最优解对应的目标值将以曲线的形式展现在图形中，如图 5.3.3。其中横坐标表示演化代数，纵坐标表示每一代中最优个体的目标值，该目标为进行【性能参数】设置时设置的目标。图中黑色的点代表对应代数的目标值，双击最近一代对应的点，可打开相应的天线模型，查看天线的特性。（注意：之前的结果一定不会优于最近一代的最优解，所以本软件只保留了最近一代的最优解，因此，双击图中前面代数的个体对应的点将无法打开对应的天线模型。）

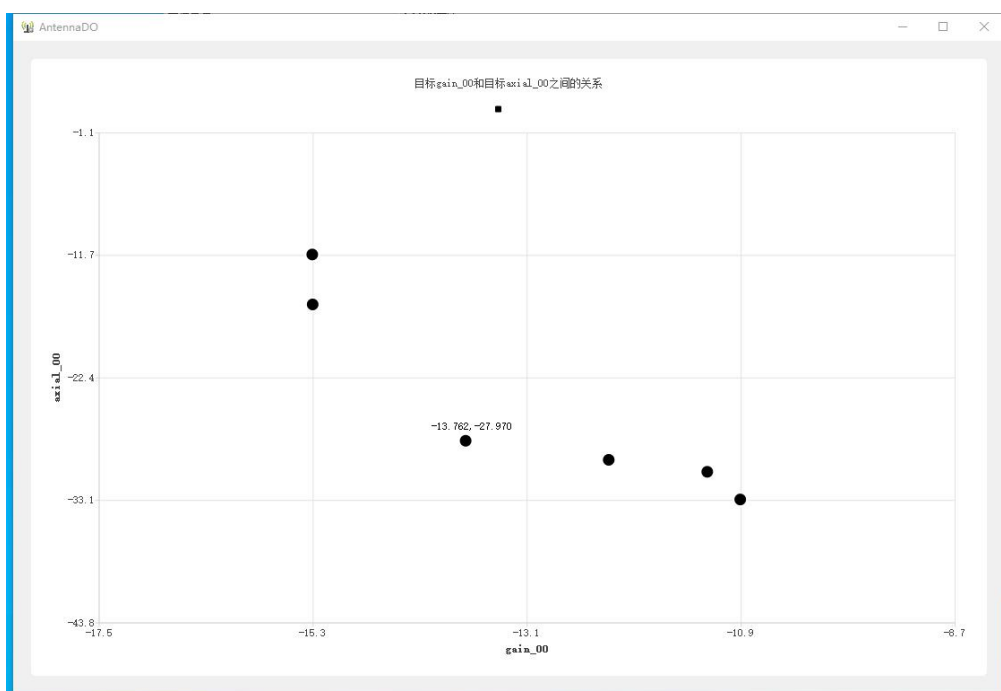


图 5.3.4 算法运行时两目标结果可视化输出

在进行多目标优化时，会以散点图的形式输出运算到当前代的最优解集的 **parato** 前沿。如图所示，图中横纵坐标分别表示进行【性能参数】设置时设置的两个目标增益和轴比的目标值，坐标轴标签后面的两个数值分别表示频段标号和同一频段内该目标的标号，如“**gain_00**”表示第一个频段设置中设置的第一个增益目标。将鼠标移动至其中的某个散点上，在该点上方将显示该点的坐标。双击该点，将会打开该点对应解的 **HFSS** 天线模型。用户可通过鼠标选定图中的指定矩形区域，从而放大显示比例，点击鼠标右键，显示区域将会以鼠标所在位置为中心缩小显示比例。（提示：当图像中无点显示时，可适当缩小显示比例，找到点所在位置，再放大显示）

对于所设目标数目大于 2 个的情况，**parato** 前沿显示时，将会以各目标两两之间的目标值散点图形式显示。如 3 个目标将会同时显示 3 个二位散点图，4 个目标显示 6 个二维散点图。三个目标输出如图

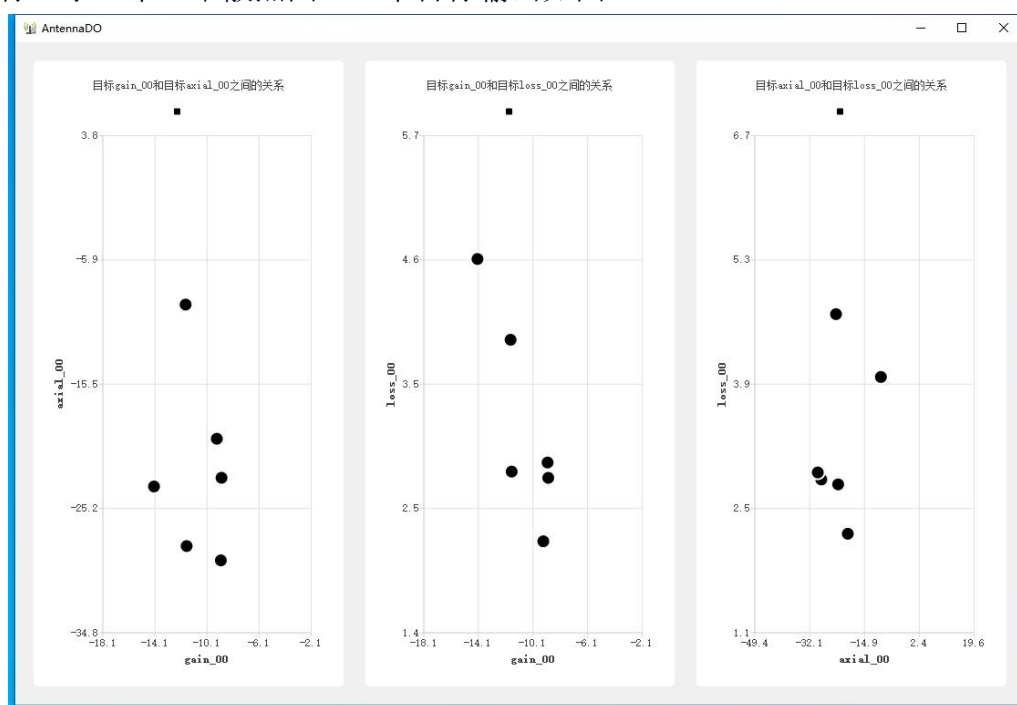


图 5.3.5 算法运行时三目标运行结果可视化输出

右击当前运行的案例名，在出现菜单中点击“停止”，算法将停止运行。下次从新运行该工程时将会从中断出继续运行。

2) 文件输出

算法运行后，在工程文件夹中（建立工程时，设置的路径），存放着工程相关文件和工程输出文件。如图，其中后缀为“.rel”的文件为工程文件，可通过此文件打开工程。后缀为“.xml”的文件为工程设置数据文件。后缀为“.aedt”和“.vbs”的文件与启动天线模型相关。新建工程之后，这些文件会自动生成。

名称	修改日期	类型	大小
Spec1	2020/1/3 21:56	文件夹	
Spec1_1	2020/1/7 9:39	文件夹	
circle_0103_03.rel	2020/1/3 20:53	REL 文件	1 KB
circle_0103_03.xml	2020/1/6 21:39	XML 文档	2 KB
Circle_reflect_antenna.aedt	2019/11/26 16:27	Ansoft Electronic...	125 KB
Circle_reflect_antenna.vbs	2020/1/3 20:53	VBScript Script ...	4 KB

图 5.3.6 工程文件夹中文件

在该文件夹下，还会有以“Spec”开头的文件夹，每一个文件夹对应工程中的一个优化案例，文件夹中存放对应案例的优化输出文件。“Spec”文件夹中的内容如图，其中“global_conf.json” 文件为进行天线优化的全局配置信息，另一个 json 文件为天线问题的配置文件，界面上的设置信息将通过这两个文件传递到软件底层算法进行运算。除此之外，还有两个保存输出文件的文件夹 “outfilepath” 和 “outhfsspath”。

outfilepath	2020/1/3 21:56	文件夹	
outhfsspath	2020/1/3 21:56	文件夹	
Circle_reflect_antenna_conf.json	2020/1/3 20:56	JSON 文件	7 KB
global_conf.json	2020/1/3 22:15	JSON 文件	1 KB

Spec文件夹内容

图 5.3.7 当前案例 Spec 文件夹内容

“outhfsspath” 文件夹中存放当前代个体对应天线的 HFSS 模型文件，“outfilepath” 文件夹中内容如图，各文件的作用如表

表 5-3-1 输出文件夹中各文件作用

文件名	文件作用
OptimalIndividual	文件夹中存放当前最优个体对应天线模型文件和仿真结果文件
result	文件夹中存放上一代所有个体对应天线的仿真结果文件
temp	文件夹中存放当前父代个体对应天线仿真结果文件，临时存放当前子代个体对应天线仿真结果文件
BestIndividuals.json	保存最优解信息，用于界面输出
BreakGeneration	保存断点信息，用于中断续算
convofpop	保存种群状态信息，用于判断算法是否收敛
errOut.log	若算法运行出错，将保存错误信息，正常则为空。
popInfo	种群信息，用于断点续算时初始化种群
spot.json	保存算法环境信息，目前只有利用 DCNSGA_II_DE 算法时会有

OptimalIndividual	2020/1/7 9:39	文件夹	
result	2020/1/7 9:39	文件夹	
temp	2020/1/7 9:42	文件夹	
BestIndividuals.json	2020/1/7 9:39	JSON 文件	10 KB
BreakGeneration	2020/1/7 9:39	文件	1 KB
convofpop	2020/1/7 9:39	文件	10 KB
errOut.log	2020/1/6 21:39	文本文档	0 KB
Generation.json	2020/1/7 9:39	JSON 文件	1 KB
logOut.log	2020/1/6 21:39	文本文档	0 KB
popInfo	2020/1/7 9:39	文件	11 KB
serverOut.log	2020/1/7 8:20	文本文档	33 KB
spot.json	2020/1/7 9:39	JSON 文件	1 KB

图 5.3.8 输出文件夹中文件内容

（注意：在软件运行时请勿让输出文件夹一直打开，否则可能造成软件运行错误，在查看相关结果之后应及时关闭）

5.4 其它功能

5.4.1 视图显示

在主界面的上部菜单栏，点击【视图】栏，可选择主界面显示的区域内容。如图 5.4.1



图 5.4.1 主界面视图选择

5.4.2 关联匹配

“关联匹配”的相关介绍见 4.2.2 节算法参数设。

5.4.3 添加问题

用于扩展问题模板库。点击主界面上端菜单栏中【设置】栏中的“添加问题”，选择所要添加的天线问题的 HFSS 模型文件，即可在模型库中添加新的天线优化

问题模型。例如：若用户已经建立好天线的 HFSS 模型文件，并命名为 “Rect_reflect_antenna.aedt”，如图，点击设置菜单栏中的“添加问题”，选择 “Rect_reflect_antenna.aedt” 文件，点击“打开”。随后，为该天线添加结构示意图，用于界面显示，如图 5.4.3，若不添加自定义的结构示意图，将采用软件默认图片表示。

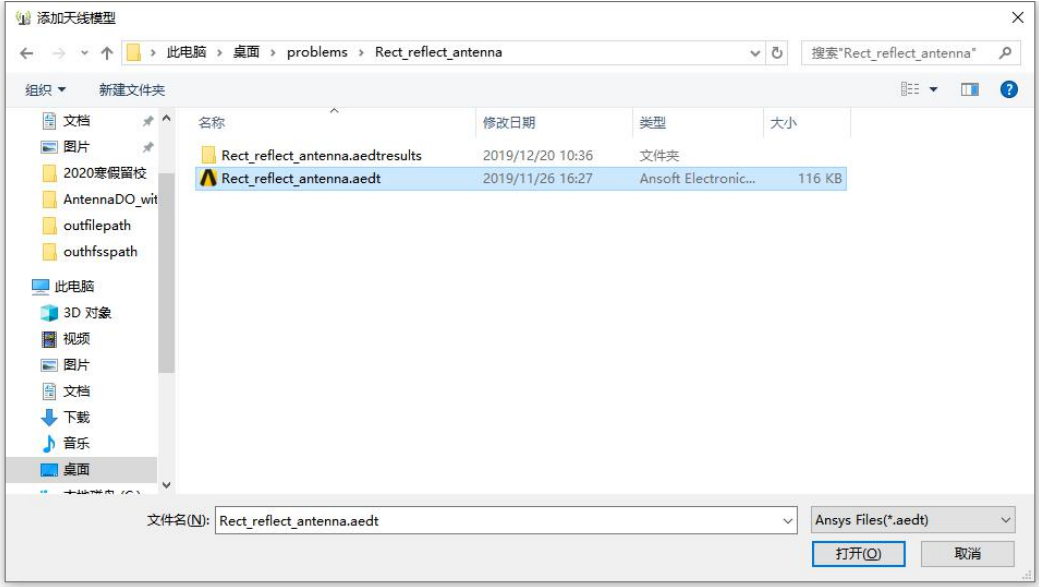


图 5.4.2 添加天线问题模型

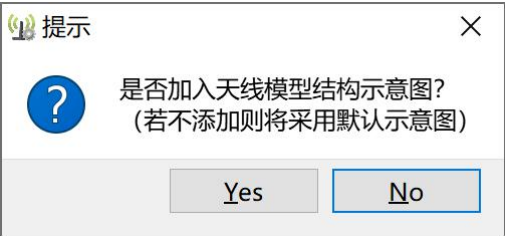


图 5.4.3 添加天线结构示意图

添加成功后，新添加的问题模型将在天线模型库中显示。如图

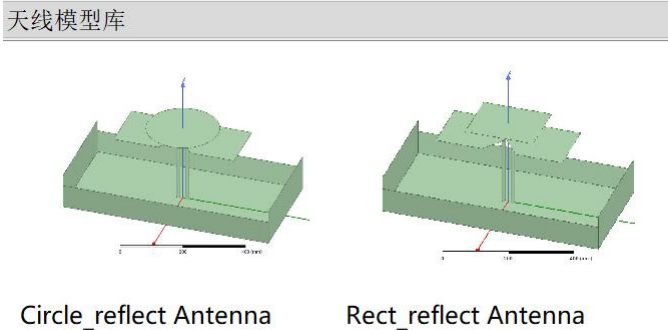


图 5.4.4 添加天线问题模型成功

5.4.4 添加算法

本软件支持算法扩展，用户可将自己编写的优化算法加入到软件算法库中，但要求所添加的算法需按照本软件规定的算法格式进行编写。用户编写新算法时可参照以下步骤进行：

在软件的安装目录中，存在一个“DEA4AD”的文件夹，里面存放着软件运行的底层代码。**注意：用户不要随意更改其中的内容，以免软件不能正常运行。**在路径“\DEA4AD\trunk\algorithm”中存放着软件中以及存在的算法文件。如图



图 5.4.5 算法模板所在文件夹

其中“ALGORITHM_XX”为算法模板文件夹，用户添加新算法时，可先拷贝该文件夹，然后对副本进行修改。**注意：不要再原始文件上修改，以免算法模板文件丢失。**在算法模板中包含一下文件。

名称	修改日期	类型	大小
ALGORITHM_XX.py	2020/1/7 16:01	Python File	4 KB
ALGORITHM_XX_conf.json	2019/12/30 20:29	JSON 文件	1 KB
ALGORITHM_XX_conf.py	2019/12/31 16:17	Python File	1 KB
ALGORITHM_XX_tool.py	2020/1/7 16:00	Python File	1 KB

图 5.4.6 算法模板包含文件

算法编写步骤：

- 1) 修改算法文件夹和其中对应文件的命名。命名格式为：“ALGORITHM_XX”，“ALGORITHM”表示算法的名称，“XX”表示算法的类型，是多目标算法还是单目标算法。建议用“MO”表示多目标算法，“SO”表示单目标算法。
- 2) 设置算法参数。用记事本打开“ALGORITHM_XX_conf.json”文件，其中内容如图，其中包含两种常用算法的参数设置，演化的“最大代数”和“种群规模”（即每一代中的个体数目）。若所加算法中有其它可变参数，可在此文件中添加。新添加的参数需保持格式一致。其中“note”是对该变量的解释说明，英文单词表示该参数在算法中的变量名，对应的数字为参数值。如图

```

{
  "generation": {
    "generation": "20",
    "note": "最大代数"
  },
  "popsize": {
    "popsize": "10",
    "note": "种群规模"
  }
}

```

图 5.4.7 算法 json 文件参数设置内容

```

{
  "generation": {
    "generation": "20",
    "note": "最大代数"
  },
  "popsize": {
    "popsize": "10",
    "note": "种群规模"
  },
  "new_para": {
    "new_para": "0.5",
    "note": "新加入的参数"
  }
}

```

图 5.4.8 算法 json 文件新增参数设置示例

注意：两项之间需用半角逗号隔开，最后一项后不用。在“ALGORITHM_XX_conf.json”文件主要用于界面与底层算法之间的交互，在 json 文件中设置好算法参数后，还需打开“ALGORITHM_XX_conf.py”，进行相应代码的编写，该文件主要用于读取算法参数。其中内容如图


```
#!/usr/local/bin/python
# -*- coding: utf-8 -*-
import os, sys

import json
WORKING_DIR = os.path.dirname(__file__)
f = None
try:
    file_name = os.path.split(__file__)[-1].split(".")[0]
    f = file(WORKING_DIR + r'/' + file_name + '.json')
    s = json.load(f)
    ###最大代数
    generation=int(s["generation"]["generation"])
    ###种群规模
    popsize=int(s["popsize"]["popsize"])
    #=====从json中读取算法参数=====#
except Exception,e:
    raise Exception('There are some errors in the algorithm configuration file:d
                    'please check the json file. Details:'+str(e.message)+'.')
finally:
    f.close()
```

图 5.4.9 算法参数配置 conf.py 文件内容

文件中已写好参数读取相关代码，并将默认参数“最大演化代数”和“种群规模”读取到变量“generation”和“popsize”中，若算法有新的参数设置，可仿照以上格式，添加相应代码。如图

```
#!/usr/local/bin/python
# -*- coding: utf-8 -*-
import os, sys

import json
WORKING_DIR = os.path.dirname(__file__)
f = None
try:
    file_name = os.path.split(__file__)[-1].split(".")[0]
    f = file(WORKING_DIR + r'/' + file_name + '.json')
    s = json.load(f)
    ###最大代数
    generation=int(s["generation"]["generation"])
    ###种群规模
    popsize=int(s["popsize"]["popsize"])
    ###新加参数
    new_para = float(s["new_para"]["new_para"])
except Exception,e:
    raise Exception('There are some errors in the algorithm configuration file:d
                    'please check the json file. Details:'+str(e.message)+'.')
finally:
    f.close()
```

图 5.4.10 算法参数配置 conf.py 文件新加参数示例

3) 编辑算法主程序。在算法主程序中，主要包含 run(), init(), loop(), leave()等函数。

- a) 其中 `run()` 函数是算法的接口函数，程序通过调用此函数来调用算法。`run()` 函数内容如图，若新算法中有新加入的算法参数，只需在图中红色方框处加入新参数读取代码即可。若没有加入新参数，该函数代码无需修改。

```
def run(problem_initialize, evaluator, outputfreq = 1, current_g=0, MaxK = 1, K = 1, condition = lambda x : False):
    """
    这是函数的入口，所有的算法只需要实现该接口即可用框架进行计算
    :param problem_initialize: 问题传入的参数列表，详见问题的problem_initialize接口
    :param evaluator: 问题的evaluate函数接口，用来计算一个个体适应值。对于普通问题就是计算函数值，对于天线问题就是调用电磁仿真软件计算
    :param outputfreq: 输出频率，默认1，由global.conf.OUTPUT_FREQ传入
    :param current_g: 当前代数，默认0，当断点续算的情况下，为中断的代数
    :param MaxK: 特定算法才使用
    :param K: 特定算法才使用
    :param condition: 终止条件
    :return:
    """
    # 参数读取，如果有其它参数可自行添加
    generation = conf.generation
    popsize = conf.popsize
    new_para = conf.new_para
    # 判断问题传入参数的合法性
    if problem_initialize[0] != 0 and problem_initialize[5] != 0:
        # 初始化种群
        init(popsize, problem_initialize, evaluator)
        # 差分算法的主循环
        loop(generation, current_g, outputfreq, condition)
        return leave()
    else:
        raise Exception('Problem is not correct: objectives=0 or genecount=0, please check the problem')
```

图 5.4.11 算法接口函数内容

- b) 算法第一步是进行初始化，该部分代码如图，其中第一部分为输出环境初始化，用到了本软件自带的工具包 `output.py`；第二部分为问题规模初始化，读取问题模块传入的问题规模信息，保存到对应的全局变量中；第三部分为生产初始化种群，该部分调用了算法相关的工具包“`ALGORITHM_XX_tool.py`”，该工具包中默认包含算法的初始化种群函数，用户可根据需求自行添加工具函数，然后在算法主程序中调用，调用方式为：`tool.func_name()`。一般情况下，初始化函数 `init()` 无需修改。

```
def init(popSize, problem_initialize, evaluator):
    output_init_output()
    global _size, _genCount, _evaluator, _pop, _trans_pop, _upper, _lower, _constraints_num, _objectives_number
    _size, _genCount, _evaluator, _upper, _lower = popSize, problem_initialize[0], evaluator, problem_initialize[1], problem_initialize[2]
    _constraints_num, _objectives_number = problem_initialize[4], problem_initialize[5]
    _pop = tool.get_feasible_init_pop(_size, _gencount, _evaluator, _upper, _lower)
```

图 5.4.12 算法初始化函数内容

- c) 初始化之后，算法将进入主循环部分。主循环 `loop()` 函数的内容如图，添加新算法时，只需在循环体出编辑算法循环部分的代码。

```
def loop(generation, current_g, outputfreq, condition):
    """
    算法的主循环
    :param generation: 代数
    :param current_g: 当前代数
    :param outputfreq: 输出频率
    :param condition: 终止条件
    """
    while (True):
        # =====循环体=====
        g = g + 1
        # 调用output模块输出信息
        output.output(_pop, g, generation, outputfreq)
        if g == generation:
            break
        if condition(_pop):
            break
```


图 5.4.13 算法主循环内容

- d) 最后一部分为算法离开部分，函数 `leave()` 内容如图，算法优化结束时，返回最优解的目标值。

```
def leave():  
    _pop.sort(cmp = poptools.compare)  
    _evaluator(_pop[0])  
    return _pop[0]['objectives'] #返回最优解信息
```

图 5.4.14 算法离开函数

完成以上步骤之后，新算法即编写完成。然后需将新写的算法添加到软件的算法库。添加方式如下：点击主界面上端菜单栏【设置】中的“添加算法”，选择所要添加优化算法所在的文件夹，选择添加后，会出现如图所示提示框，选择新加入算法是否为多目标算法。若新添加算法为多目标算法，点击是，若为单目标算法，点击否。完成之后，点击【设置】中“关联匹配”项，可查看算法是否加入到软件算法库。

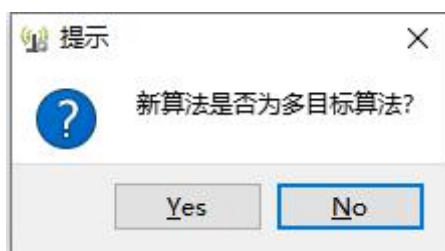


图 5.4.15 添加新算法

5.4.5 帮助

点击主界面上部菜单栏中的【帮助】，然后点击子菜单中的“使用说明”，可代开本软件的使用说明文档。