

# Y-GAMA 指标调整向导帮助

ver 3.0

## 目 录

（一）启动向导 .....	1
（二）流程简介 .....	1
1. 设置问题的参数 .....	2
2. 设置问题的约束 .....	2
3. 设置问题的目标 .....	2
注：约束与目标的区别 .....	2
（三）操作流程 .....	2
1. 设置参数 .....	3
2. 添加其他参数 .....	3
3. 设置约束 .....	4
4. 设置目标 .....	5
5. 启动优化及结果展示 .....	6
（四）再次编辑和保存读取 .....	8
1. 再次打开指标调整向导 .....	8
2. 保存和读取 .....	9
3. 卡片编辑指标调整向导 .....	10
附录：常见问题 .....	10
1. 什么是算法？ .....	10
2. 算法内部如何运行？ .....	11
3. 什么时候用算法？ .....	11
4. 算法求解问题所需要的时间： .....	12

5.	什么是最新的 Online Learning 算法? .....	12
6.	算法选择建议及参数设置? .....	13
	(1) 选择建议.....	13
	(2) 参数设置: .....	13
7.	启动了计算, 明明在计算, 但优化界面不显示计算结果.....	14
8.	用算法计算了很久也没有效果? (即对算法的优化参数设置建议) .....	14
	(1) 错误举例: .....	14
	(2) 正确举例: .....	14

## （一）启动向导

可以通过点击“轴线网格”标签下的“指标调整”按钮来启动指标调整向导：

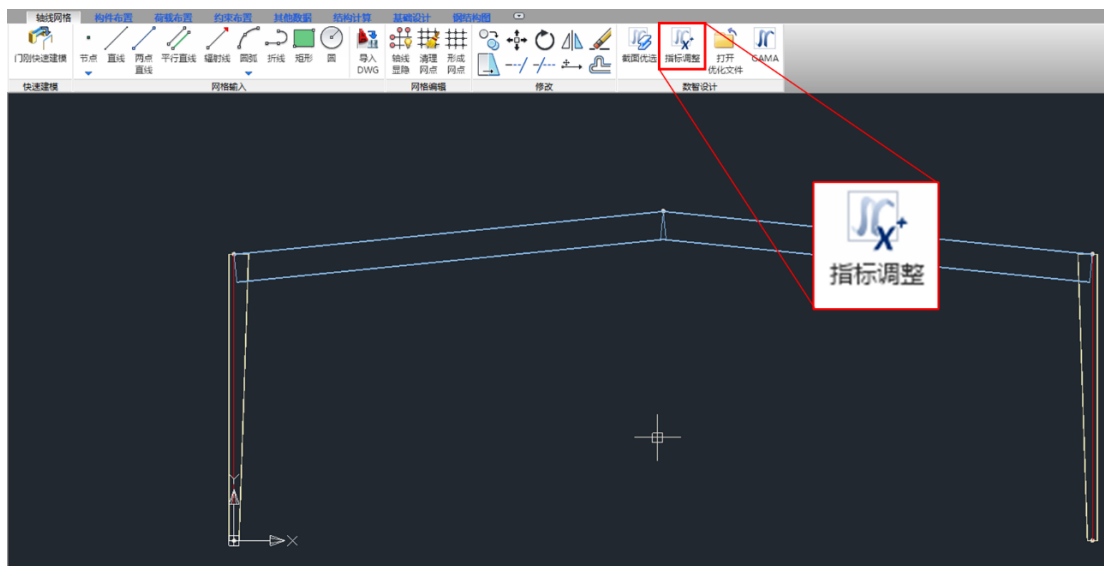


图 1.1

## （二）流程简介

算法的原理可以参考“指标调整常见问题”中内容，在此不做赘述。

门刚的指标调整向导主要针对于通过智能算法来调整构件截面以使得模型的指标在满足特定限制的情况下，最大化或者最小化某项（或某几项）指标”的特定应用场景，其中，“模型的指标”包括：

- 结构总质量
- 材料用量（型钢）
- 周期
- 工程造价
- 构件超限数量
- 门刚梁、柱、支撑应力比限值
- 门刚柱顶位移与高度比

整个使用流程大致分为三个步骤：

1. 设置问题的参数

设置需要调整的构件及智能算法在调整它们时可选的候选截面；

2. 设置问题的约束

在智能算法调整模型过程中，模型需要满足的指标；

3. 设置问题的目标

在对模型计算结果评价中，需要关注的指标（即“哪个模型是较好的模型”）。

注：约束与目标的区别

约束是满足即可，例如是结构的某些指标。

目标是越小越好，例如在满足结构指标(约束)的条件下，哪个模型是最好的；目标可以是经济性，也可以是结构某项指标。关键在于希望哪一项**越少越好**。

(三) 操作流程

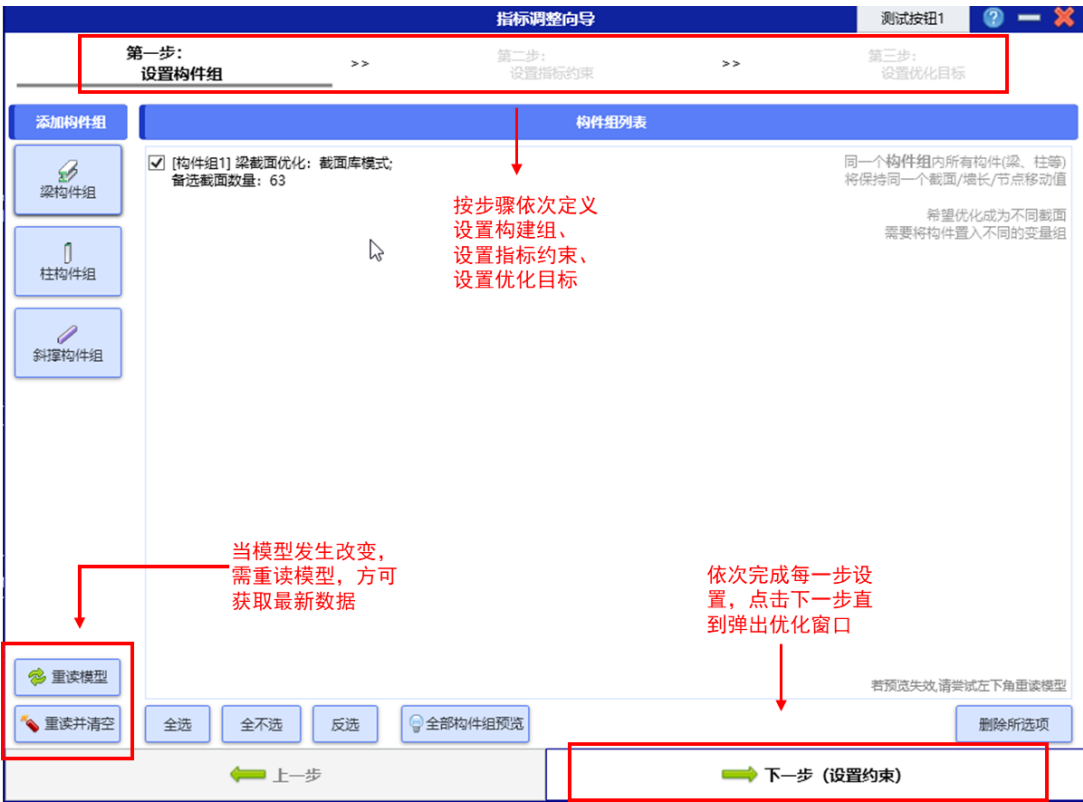


图 3.1

## 1. 设置参数

### 1.1 选择需调整的设计构件

每一个参数是由一组别调整的“设计构件”和一组“备选截面”组成的。设置参数时，首先需要选择参数所对应的构件类型（图 3.2 ①）。

然后点击**添加构件组**来打开添加参数窗口。首先根据图 3.2 中 ②~④选择“设计构件”，然后通过⑤选择这部分构件的候选截面。注意在指标调整中，一组“设计构件”始终保持为一个截面：

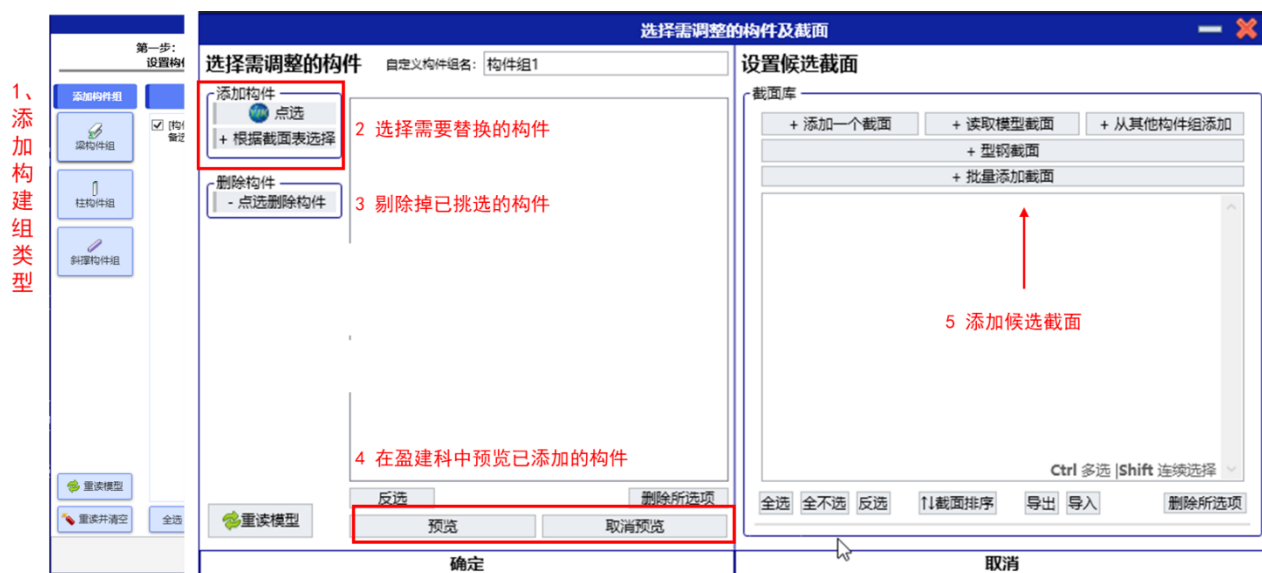


图 3.2

### 1.2 设计构件的候选截面

当选择**截面库**模式时，可以使用按钮来添加自定义截面，或者是从现有盈建科模型中读取在模型中已经建立的截面类型。

候选截面会赋予到“设计构件中”，计算过程中每次挑选一个截面。即一组“设计构件”始终保持为一个截面。

## 2. 添加其他参数

一个参数添加完毕后，可以继续添加其他的参数。

参数之间是互相独立存在的，算法在调整截面时，会将挑选其中一个候选截面赋予给这“设计构件”。即一个参数内的所有“设计构件”经过算法调整后会是同一个截面。

### 3.设置约束

智能算法在进行模型评价时，会倾向于选择所有约束都满足的模型，正确地选择约束可以让算法帮设计师朋友们快速排除不符合需求的模型。因此，过于严格的约束可能导致算法始终找不到符合要求的模型。

关于算法的说明，可以参考“常见问题目录”中内容，在此不再赘述。

门刚模块的指标约束类型与 GAMA 主模块的略有不同，增加了门刚柱应力比限值、门刚支撑应力比限值、门刚梁应力比限值、门刚注定位移与高度比。



图 3.3

在“设置指标约束”窗口中，左侧是挑选约束，可以对指标类型、具体指标、约束行为等进行详细设置，设置完毕后，可以单击中间的“添加”按钮来将挑选好的约束添加到算法的约束行为中，成为“已设置约束”。



图 3.4

## 4. 设置目标

由于出现多个满足所有约束的模型时，算法将挑选“目标”更优秀的模型为最佳模型，并且使用其参数来规划下一次模型的生成参数。

所以，目标可以说是结构优化的总纲领；而约束是满足即可。

与约束窗口类似，左侧挑选目标之后，通过中间的“添加”按钮来进行设置。与约束不同的是，目标无需进行详细的约束行为设置（大于小于等于），只需要选择“优化方向”，即尽量往大了调、尽量往小了调，以及尽量往某一个值去贴近。



图 3.5

当选择“尽量接近指定值”作为目标时，需要额外指定具体的趋近值，并且右侧附有“？”帮助按钮，可以查看具体含义。

需要注意，Y-GAMA 内置的智能算法均支持设置多个目标。但目标数量过多（3 个及以上）会极大程度上地增加计算复杂度，也会降低智能算法的评价函数效率。除非必要，尽量设置较少数量的目标（2 个及以下），将其余需要控制的指标值设置成为约束。

勾选**删除中间过程计算结果**将删除中间过程的计算结果，在迭代次数很多的情况下，此举可以节省储存空间。

设置目标完成后，可以点击左下角“完成设置”按钮来完成向导，打开优化器界面。



图 3.6

## 5.启动优化及结果展示

### 5.1 启动优化

设置好目标，点击确定后，就弹出优化器界面，如下图 3.7。这部分截面主要分三部分内容：选择算法、设置对应参数、启动优化。



1、选择对应算法，可通过点击“算法帮助”按钮确定算法

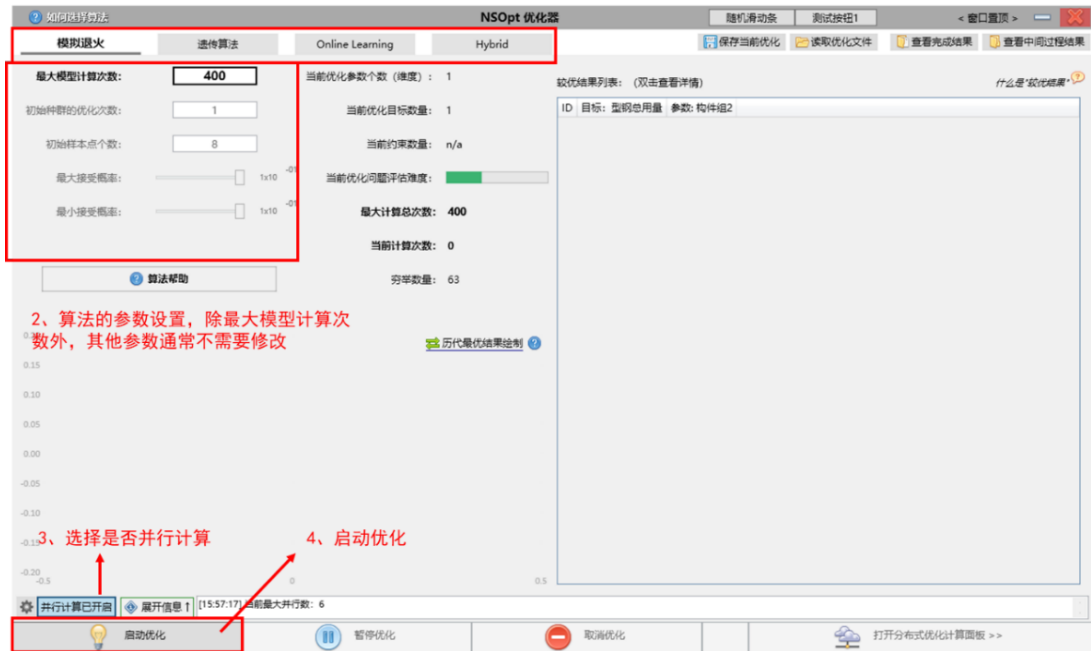


图 3.7

关于算法的选择，可以参考“附录：常见问题”中对应内容。

## 5.2 计算结果展示

计算结果分为左侧“图形展示区”和右侧较优结果“数据展示区”。

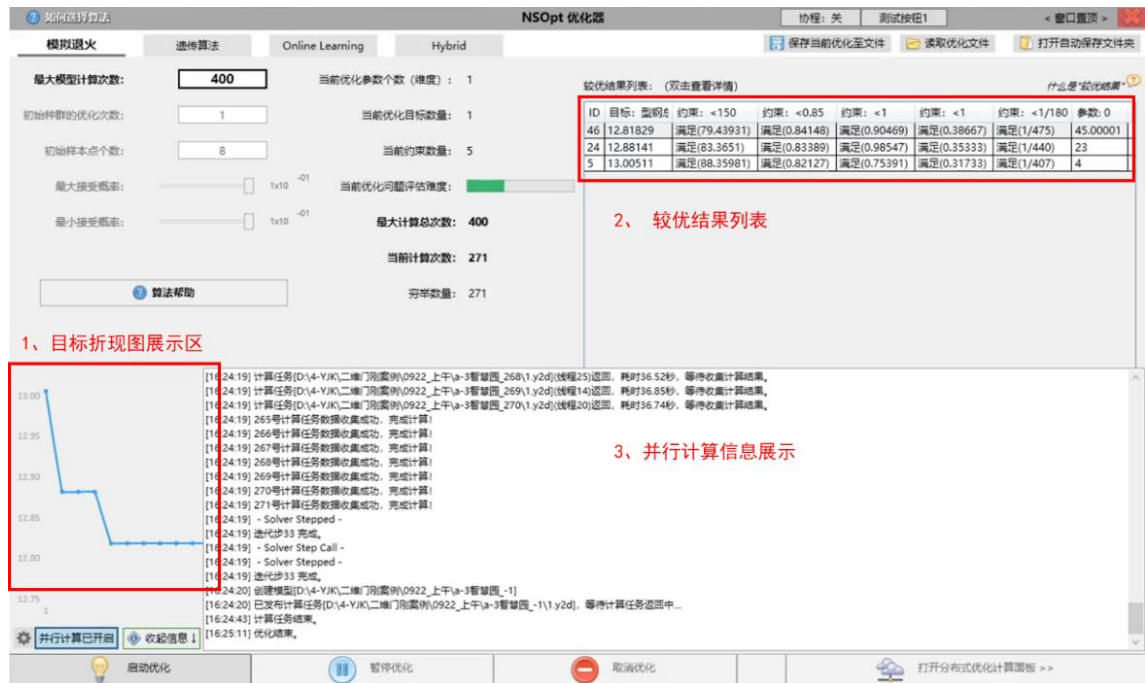


图 3.8

数据展示区中，ID 为计算的模型编号，目标、参数、约束为对应模型的结果。

如 ID46 为第 46 个模型。对应目标为型钢总质量，12.82t。



图 3.9

5.3 在计算却不显示结果:

参考“优化算法常见问题”第 7 条。

5.4 文件保存



图 3.10

(四) 再次编辑和保存读取

1. 再次打开指标调整向导

优化向导完成后，关闭弹出的“优化器”窗口，如下图 4.1，会回到 Y-GAMA 模块的画布上，双击鼠标滚轮中键，即可以看到由“指标调整向导”自动创建的一系列卡片，如下图 4.2。

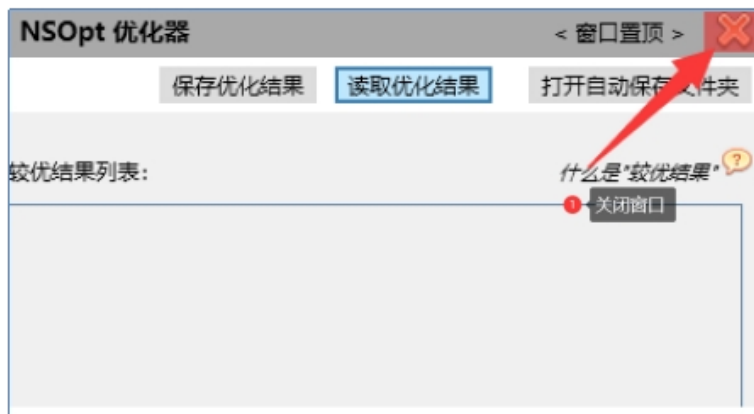


图 4.1

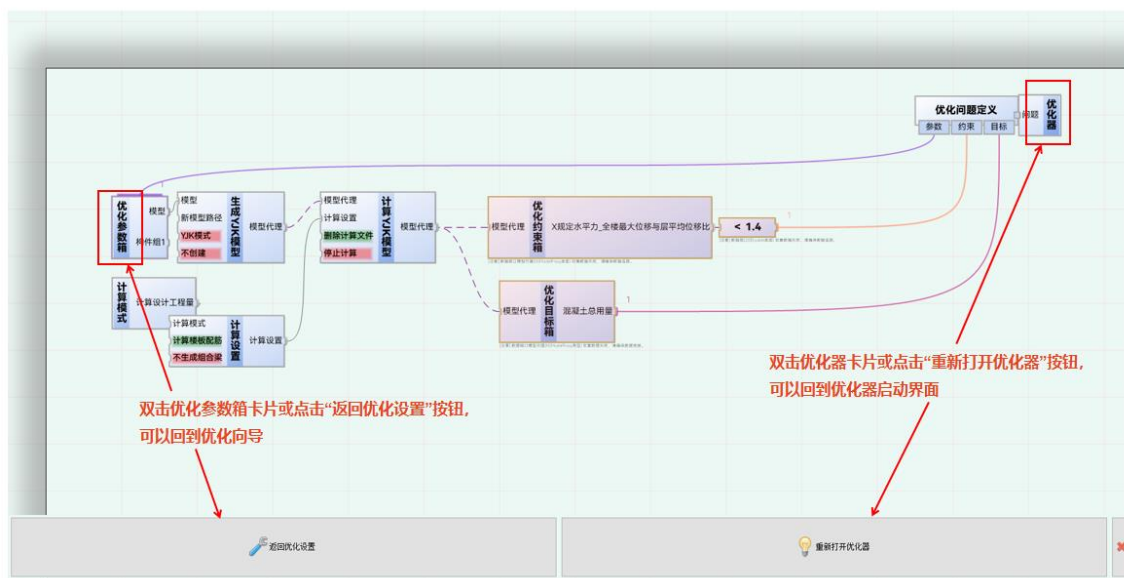


图 4.2

再次打开指标调整向导，可以双击画布的“优化参数箱”卡片或点击“返回优化设置”按钮，回到指标调整向导中，如上图 4.2。之前设置的参数、约束、目标均会保存在内。

再次启动指标调整，双击画布“优化器”卡片或点击“重新打开优化器”按钮，回到优化器启动界面，如上图 4.2。

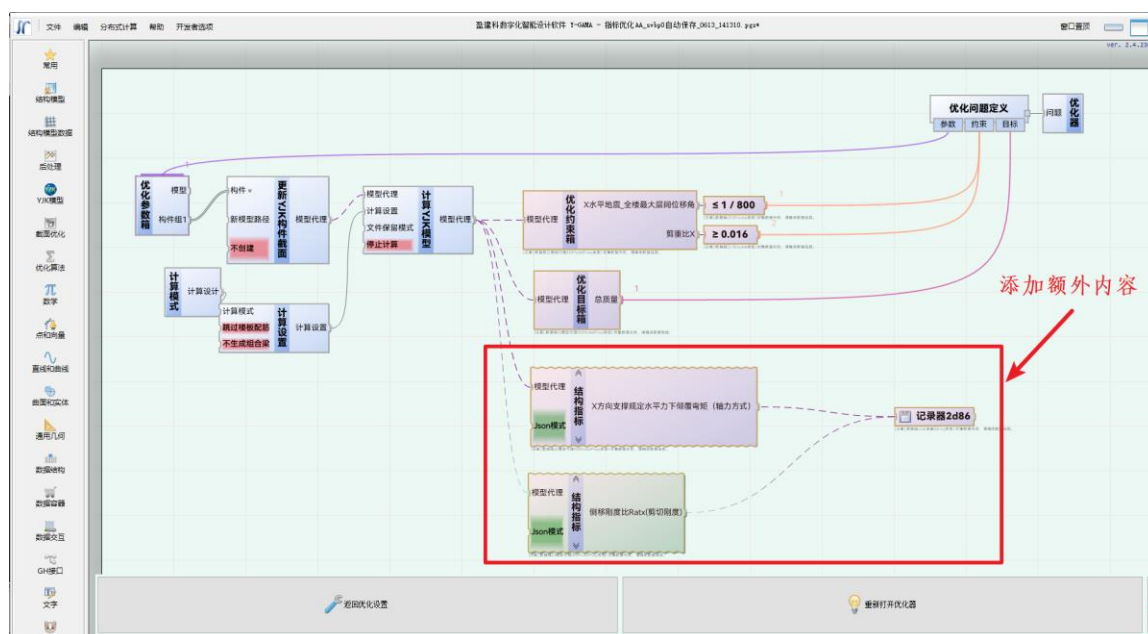
## 2. 保存和读取

当弹出画布界面后，截面优选参数均会保存在卡片组中。也就是说，只需保存画布上该文件（扩展名为.pgiz），再次打开 Y-GAMA 时，通过画布读入对应的.pgiz 文件即可获取所有的数据。



### 3. 卡片编辑指标调整向导

对于习惯使用卡片的用户，可以在优化向导生成的卡片基础上进行编辑。如果还有其他未能在指标调整向导中设置的约束和目标，可以在该画布上继续添加“结构指标”卡片来读取额外的数据，并连入“优化问题定义”卡片中，进行进一步的定制化。



总而言之，指标调整向导的流程保持了“参数化”流程打造，您可以在此基础上任意添加内容，比如添加额外的计算、实现结果写出文件等等。

## 附录：常见问题

### 1. 什么是算法？

首先说明，YGAMA 没有有限元求解器，而是调用 YJK 进行有限元求解。

可以认为，算法是一种智能的自动求解“问题”的方法。

一个“问题”，一般由三部分组成：

①问题的变量，如梁柱截面尺寸，为参数；

②问题需满足的条件，如结构的大指标，为约束；

③由于在变量(梁柱截面变化)范围内，可能存在很多满足条件(结构指标)的结果，实际只需要一个最优结果，如经济性最优；这个结果为目标。

## 2. 算法内部如何运行？

具体来说，算法计算时首先会“随机”生成第一代模型（具有一定数量），记录此时的参数并计算，算法内部对计算结果评分，满足约束且目标越小（或更大）为较优模型；根据评分，选择更为合理的参数生成下一代模型，如此反复。不断挑选出满足约束条件下，目标更小（或更大）的模型。

关于约束和目标的区别，可以简单的这样区分：

**约束为满足即可；目标为越小越好。根据这个原则设置约束和目标。**

**例 1：**大部分模型，要求指标满足束即可，因此指标为约束。

在满足约束的模型中，要找一个造价最低的，因此目标为造价。

**例 2：**一个模型调整了很久，周期比指标总是不过，要求调过指标，这个时候也不关心经济性。

因此周期比要求越小越好，是目标；其他指标满足即可，为约束。

## 3. 什么时候用算法？

算法是一种智能的自动求解“问题”的方法。可以很容易的想到，还有一种求解“问题”的方法——穷举法。

当模型较小，“问题”的总计算次数为几百次时，求解问题只需要几小时。

当求解“问题”的穷举数量为几千~几百万之间时，想通过一个更快速的方法找到一个满意的结果，此时就使用算法。

	模拟退火	遗传算法	Online Learning
性能	<ul style="list-style-type: none"> <li>• 局部搜索能力强</li> <li>• 收敛快</li> </ul>	全局搜索能力强	计算次数更少
计算成本	低	相对较高	极低
适用场景	在已知解附近搜索更优解	无法判断变量与目标之间明确的关系	变量与目标之间存在较明确的关系

#### 4. 算法求解问题所需要的时间：

上文已经提到，Y-GAMA 本身没有有限元求解器，而需要启动 YJK 完成模型计算，然后通过算法不断迭代修改模型。

根据经验，当穷举数量为几千~几百万之间时，元启发算法(退火、遗传、粒子群)约需要计算 0.1%~2%乘以穷举量的模型，基本可以找到一个满意的较优模型。穷举量越大，取值越小。机器学习算法（onlinelearning）所需的计算量更少。

举例：

200w 穷举，计算模型数量为：200w $\times$ 0.1%=2000 个模型。

2000 穷举量，计算模型数量为：2000 $\times$ 2%=40 个模型。

假设计算一个模型需要 5min，上述两种情况，分别需要 2000 $\times$ 5=1w 分钟，40 $\times$ 5=200 分钟。

以上均为建立在合理的优化参数条件下的经验值，不排除特殊的可能性。

这种时候也可以采用 Online Learning 算法以及分布式计算来降低计算时间（调用局域网内安装有 Y-GAMA 的计算机一起算）。

#### 5. 什么是最新的 Online Learning 算法？

Online Learning 算法基于在线学习框架的算法，对比元启发式算法（退火、遗传、粒子群）收敛较快，可以减少大部分的计算量。该算法并不会上传用户数据。

此算法限值条件：①现阶段在目标与约束数量之和小于 7 个，且优化参数数量小于 10 个时，新算法有较好的收敛性。否则算法本身所需的计算时间（除模型计算外的计算时间）有明显的增加，后续会更新以满足更复杂的问题。②在找到最优解后，还会继续进行求解。

## 6. 算法选择建议及参数设置？

### (1) 选择建议

建议使用 Online Learning 机器学习算法，对比其他三个算法有速度优势。也可以尝试采取其他算法。

其他算法的选择：

①退火算法：适用于简单的离散或连续问题的求解。穷举次数不超过 500 时，退火的效率较高，另外退火算法有较好的局部搜索能力。

②遗传算法：通用的算法。对参数、约束、目标和问题穷举次数没有明显要求。如果一个问题用其他算法找不到比较好的解但确定参数范围内一定有满足约束，结果更优的模型，那么可以用遗传算法来计算，但需要计算的数量多一些。

③粒子群算法：与遗传算法类似，如果优化问题更接近一个连续函数性能会好一些其他情况建议直接用遗传算法。

### (2) 参数设置：

所有的算法，必须要设定的参数只有最大迭代次数这一个参数（下图红框位置），其他参数都可以使用默认值，这个参数决定最多计算的模型个数。

模拟退火	遗传算法
初始样本点个数: <input type="text" value="8"/>	种群规模: <input type="text" value="20"/>
初始种群的优化次数: <input type="text" value="1"/>	最大迭代次数: <input type="text" value="20"/>
最大迭代次数: <input type="text" value="400"/>	

粒子群	Online Learning
粒子数量: <input type="text" value="20"/>	初始样本点个数: <input type="text" value="8"/>
最大迭代次数: <input type="text" value="20"/>	最大迭代次数: <input type="text" value="50"/>



## 7. 启动了计算，明明在计算，但优化界面不显示计算结果

一般情况下，算法在算完一代模型后才会右侧“数据展示区”显示第一个结果，一代模型数量一般为8个（模拟退火）、40个（退火及粒子群）、8个（Online Learning）。

## 8. 用算法计算了很久也没有效果？（即对算法的优化参数设置建议）

从上面我们知道了如何选择算法；接下来我们看下，如何提升算法求解问题的效率。

从“4. 算法求解问题所需要的时间”可以知道，用算法求解问题主要消耗的时间为基数(穷举量) $\times$ 比例(约为0.1%~2%)，除了选择合理的算法外（见第5项），最有效的是降低穷举计算量。

在使用算法的时候，需要告诉算法哪些构件需要调整，因此有不同的分组，每组有不同的候选截面。

### (1) 错误举例：

① 某项目设置了9组修改截面，每组设置一个型钢库作为候选截面（共200种截面）。

**错误原因：**穷举数量为 $200^9=512$  亿亿个模型。即使是0.1%的计算比例，在这么多模型中找到一个较优模型，无异于大海捞针。

② 某项目主梁和次梁都放到指标调整中调整，主梁8组，每组4个截面；次梁6组，每组4个截面。穷举量： $4^{(8+6)}=2.6$  亿次。

**错误原因：**指标调整主要针对指标优化，而次梁不抗侧力。因此次梁调整可以使用截面优选（内力反算的优化策略），经过几次迭代就可以快速找到次梁最优截面。此时的总计算量约为 $4^8 \times 0.1\% = 75$  次。

### (2) 正确举例：

某项目，位移比、周期比等指标易超限，但基本满足规范。设计师根据经验选定了8组对指标影响较大的构件，每组给了4个候选截面，根据惯性矩比从小到大排序。

由于这个优化问题属于不同构件之间相互组合，因此采用遗传算法。最终计算了130个模型，找到满意的结果，计算比例为 $130/4^8=0.2\%$ 。