

# AOE 网络设计说明书

<https://github.com/lingbai-kong>

同济大学

2020 年 7 月

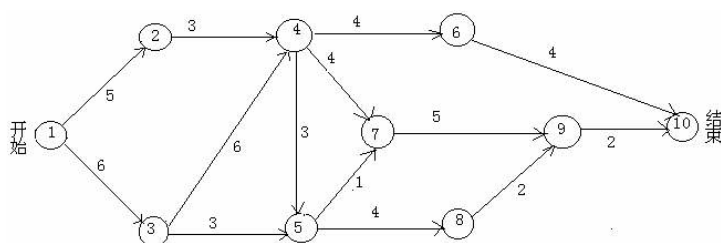
## 第二部分 综合应用设计说明

### 2.1 题目

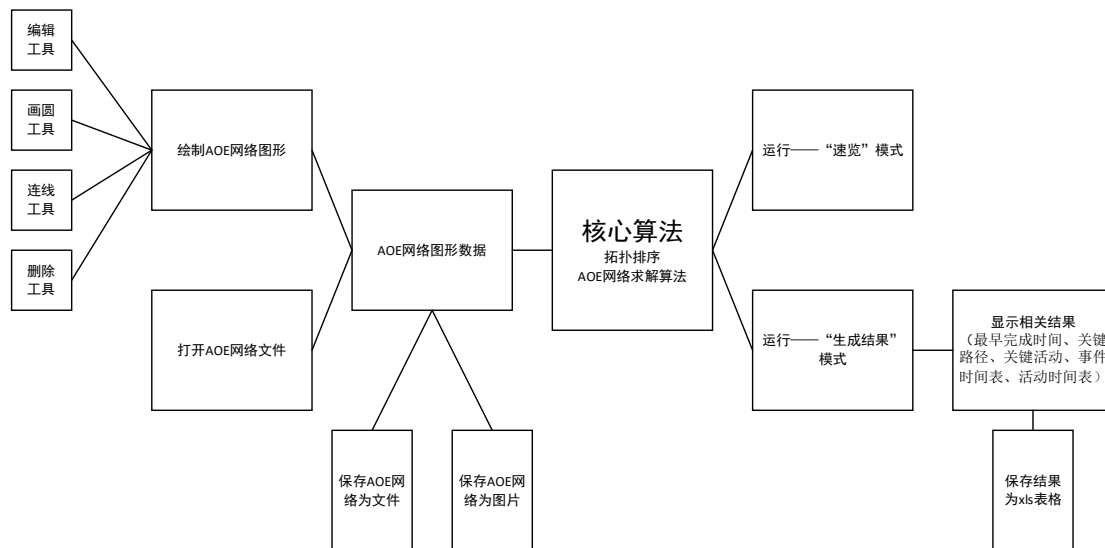
6. ★★已知假想的工程活动图 AOE 网，试设计一个算法，要求：

- (1) 判断工程是否可行；
- (2) 求出工程中每个活动的最早开始时间 $e(i)$ ，最迟开始时间 $l(i)$ 和全工程可以完成的最早时间；
- (3) 确定工程中关键路径和可使整个工程的工期缩短的关键活动。

说明：可以给出一个假想的工程活动图 AOE 网，如下图所示（仅为示例），也可以给出工程的活动情况，画出 AOE 网。



### 2.2 软件功能



（功能结构图）

#### 2.2.1 程序具有画图功能，能够让使用者使用鼠标在绘图区画出 AOE 网络。

实现方式：通过 PyQt5 提供的工具栏实现绘图工具的切换，根据鼠标行为和当前选中的工具，在绘图区内实现图形的新建、连接、更改、删除操作，并将图形数据保存。

#### 2.2.2 程序能够将使用者画出的 AOE 网络保存为文件，并能够随时在程序中打开这

些文件。打开文件后使用者可以继续编辑 AOE 网络并求解，修改后的 AOE 网络支持保存和另存为操作。

实现方式：通过二进制方式对文件进行读写，保存时将图形数据用 `pickle.dump` 方法写入文件，打开时用 `pickle.load` 方法从文件中读取图形数据并将图形绘制到界面绘图区

**2.2.3 程序可以把绘制出的 AOE 网络保存为图片，图片支持四种格式：jpg、png、tiff、bmp。**

实现方式：首先通过文件保存对话框选中保存文件的路径和文件名称，然后获取绘图区的相关数据，再把数据保存为指定图片格式。

**2.2.4 程序有两种运行方式：“速览”和“生成结果”。“速览”可以检查 AOE 网络是否可行，如果可行将在绘图区内把相应的关键路径标红，如果不可行将输出提示信息。“生成结果”除了能完成“速览”的功能外，还可以输出工程 AOE 网络的最早完成时间、关键路径、关键活动、事件时间表、活动时间表。**

实现方式：先将绘图区的图形数据转化为邻接表数据，然后对图进行拓扑排序，在此同时检查图是否是有向无环图，即 AOE 网络是否可行，如果是则开始运行 AOE 算法，计算最早完成时间、关键路径、关键活动、事件时间表、活动时间表。如果是“速览”模式则根据关键路径将绘图区内的对应路径标红，如果是“生成结果模式”还需要将其他信息展示到结果停靠栏中。

**2.2.5 程序可以将运行结果保存为 xls 表格文件。**

实现方式：首先通过文件保存对话框选中保存文件的路径和文件名称，然后获取结果停靠栏中的相关数据，再把数据保存为 xls 表格格式。

**2.2.6 程序可以提供程序的使用帮助。**

实现方式：通过消息提示框弹出帮助信息。

**2.2.7 程序的结果页面、工具栏都可以取消停靠，以新窗口的形式呈现。**

实现方式：工具栏通过 PyQt5 的 `toolbar` 控件实现，停靠窗口通过 PyQt5 的 `dockWidget` 控件实现。

**2.2.8 程序中绝大部分操作都支持快捷键操作。**

实现方式：通过 PyQt5 提供的 `setShortcut` 函数对相应控件进行设定。

**2.2.9 程序的标题栏和状态栏会随着程序状态的改变显示不同的信息。**

实现方式：在程序状态改变时通过 PyQt5 提供的 `setWindowTitle` 函数和 `QStatusBar.showMessage` 函数设定标题栏和状态栏显示内容。

## 2.3 设计思想

### 2.3.1 底层算法

#### 2.3.1.1 拓扑排序

##### 原理简介<sup>1</sup>

拓扑排序是由某个集合上的一个偏序得到该集合上的一个全序。直观地看偏序指集合中仅有部分成员可以比较，而全序指集合中全体之间均可以比较。对 AOE 网络进行拓扑排序是求解 AOE 网络的先决条件。

拓扑排序的方法如下：

1. 在有向图中选一个没有前驱的顶点输出
2. 从图中删除该顶点和所有以它为尾的弧
3. 重复以上两步，直到全部顶点均已输出，或者当前图中不存在无前驱顶点为止，后一种情况说明图中存在环。

##### 设计实现

数据结构设计：<sup>2</sup>

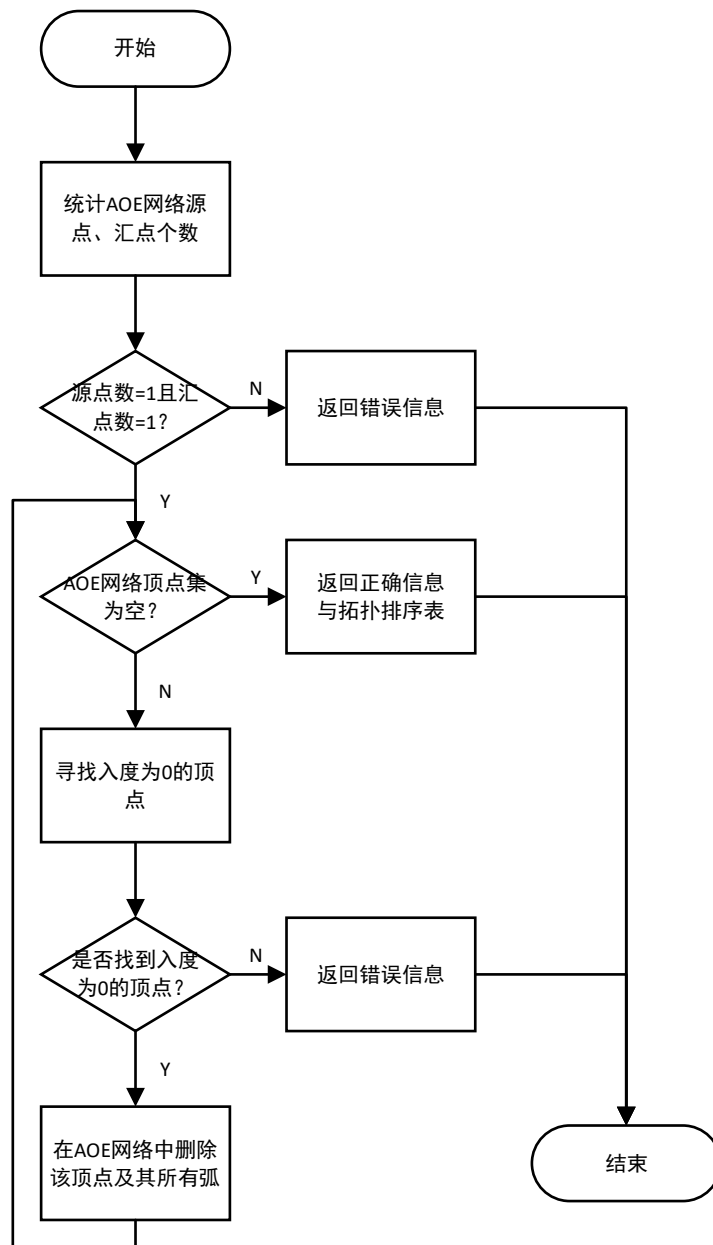
拓扑排序直接利用绘图区的图形数据进行运算。图形数据包含两个字典，一个字典以顶点为索引记录顶点的入弧和出弧，另一个字典以弧为索引，记录弧头和弧尾。其实这两个字典一起组成了一个十字表，只不过顶点结点和弧结点并不是通过记录指针相互连系，而是记录在各子字典中的索引实现。

算法设计：

---

<sup>1</sup> 参考自 严蔚敏 吴伟民，《数据结构（C 语言版）》，清华大学出版社，P180-P182

<sup>2</sup> 代码中的深拷贝操作参考自网站资料 《python-copy()与 deepcopy()的区别》  
<https://www.cnblogs.com/hokky/p/8476698.html>



(拓扑排序算法流程图)

因为要求网络为 AOE 网络，而拓扑排序仅能检查网络是否无环，因此在此给算法加入检测源点、汇点数目的步骤以适应程序要求，具体算法如下：

1. 遍历 AOE 网络的顶点，统计 AOE 网络中源点、汇点的个数，如果两种个数不为 1 则返回错误消息。
2. 当 AOE 网络顶点集不为空时
3. 从 AOE 网络中寻找入度为 0 的顶点
4. 如果 AOE 网络中没有入度为 0 的顶点，那么说明 AOE 网络中有环，输出错误信息
5. 否则将找到的顶点加入拓扑排序表，并在图中删除该顶点及其所有弧

## 6. 返回正确信息和拓扑排序表

### 2.3.1.2 AOE 网络求解算法

#### 原理简介<sup>3</sup>

AOE 网络是一个带权的有向无环图，顶点代表事件，弧表示活动，权表示活动持续时间，AOE 网络可以用来估算工程完成的最短时间。由于在 AOE 网络中有些活动可以并行地执行，所以完成工程的最短时间是从开始点到完成点的最长路径的长度，这一条路径就是关键路径，分析关键路径的目的是要辨别哪些活动是关键活动以便争取提高关键活动的功效缩短工期。

计算关键活动需要定义这些量：事件  $V_i$  的最早可能开始时间  $Ve[i]$ ，事件  $V_l[i]$  的最迟允许开始时间  $Vl[i]$ ，活动  $a_i$  的最早可能开始时间  $e[i]$ ，活动  $a_i$  的最迟允许开始时间  $l[i]$ 。时间余量  $l[i]-e[i]$  表示活动  $a_i$  的最早可能开始时间和最迟允许开始时间的的时间余量。 $l[i]=e[i]$  表示活动  $a_i$  是没有时间余量的关键活动。为了找出关键活动需要求出各个活动的  $e[i]$  与  $l[i]$  而两个量的求解又基于  $Ve[i]$  和  $Vl[i]$ 。具体递推公式如下：

$$\begin{cases} e(i) = Ve(i) \\ l(i) = Vl(k) - dut(< j, k >) \end{cases} \quad \text{其中 } T \text{ 是所有以第 } j \text{ 个顶点为头的弧的集合}$$
$$\begin{cases} Ve(j) = \underset{i}{Max}\{Ve(i) + dut(< i, j >)\}, < i, j > \in T, j = 1, 2, \dots, n-1 \\ Vl(i) = \underset{j}{Min}\{Vl(j) - dut(< i, j >)\}, < i, j > \in S, i = n-2, \dots, 0 \end{cases} \quad \text{其中 } S \text{ 是所有以第 } i \text{ 个顶点为尾的集合。}$$

寻找关键路径的方法如下：

1. 输入  $e$  条弧  $< j, k >$  建立 AOE 网的存储结构
2. 从源点  $V_0$  出发，令  $Ve[0]=0$ ，按拓扑有序求其余各顶点的最早发生时间  $Ve[i]$ 。如果得到的拓扑有序序列中顶点个数小于网中顶点数  $n$  则说明网中存在环，不能求关键路径，算法终止；否则执行下一步。
3. 从汇点  $V_n$  出发，令  $Vl[n-1]=Ve[n-1]$ ，按拓扑逆有序求其余各顶点的最迟发生时间  $Vl[i]$ 。
4. 根据各顶点的  $Ve$  值和  $Vl$  值，求每条弧  $s$  的最早开始时间  $e(s)$  和最迟开始时间  $l(s)$ 。若某条弧满足条件  $e(s)=l(s)$ ，则为关键活动。

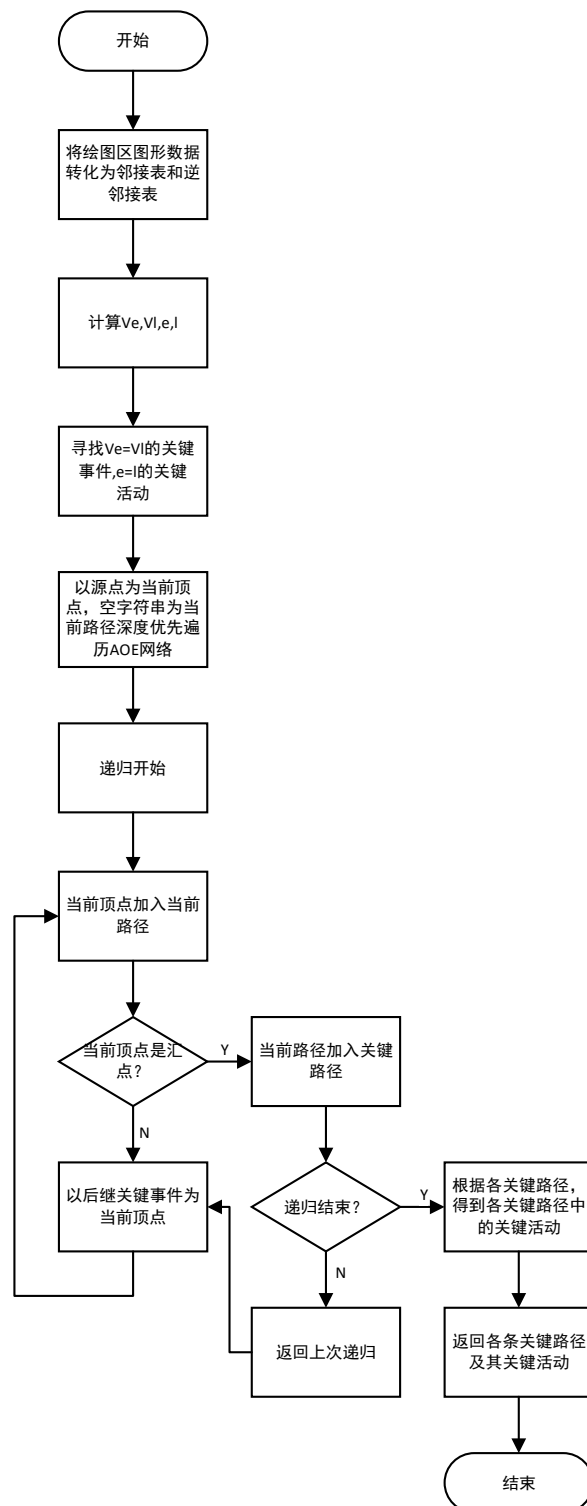
#### 设计实现

##### 数据结构设计

<sup>3</sup> 参考自 严蔚敏 吴伟民，《数据结构（C 语言版）》，清华大学出版社，P183-P184

因为求解是需要先拓扑有序计算  $V_e$  再拓扑逆有序求解  $V_l$ ，为了方便计算，数据结构使用两张邻接表——邻接表和逆邻接表。邻接表采用字典方式的数据类型，键为顶点名，值中记录了该顶点的后继和指向后继弧的权值。逆邻接表也采用字典方式的数据类型，键为顶点名，值中记录了该顶点的前驱继和由前驱指向它的权值。

#### 算法设计



### (AOE 网络求解算法流程图)

因为上一步拓扑排序中已经检验了 AOE 网络的合法性并给出了拓扑排序表, 因此求解关键路径时无需进行以上操作。

1. 将绘图区图形数据转化为邻接表和逆邻接表
2. 根据上文中给出的公式计算  $Ve$ 、 $Vl$ 、 $e$ 、 $l$ <sup>4</sup>
3. 寻找  $Ve[i]=Vl[i]$  的事件加入关键事件表
4. 寻找  $e[i]=l[i]$  的活动加入关键活动表

由于一个 AOE 网络中可能有多条关键路径, 而上述步骤中仅能给出关键事件、关键活动的集合, 并不能直观地给出各条关键路径, 因此还需要将各条关键路径分离出来。

5. 以源点为当前顶点, 空字符串为当前路径进入递归操作, 深度优先遍历 AOE 网络。
6. 将当前顶点加入当前路径中
7. 如果当前顶点是汇点, 将当前路径加入到关键路径中。否则检查当前顶点后继中的关键事件递归地调用该算法进入 6
8. 根据各条关键路径, 得到各条关键路径中的关键活动。
9. 返回各条关键路径和各条关键路径中的关键活动。

## 2.3.2 UI 界面

### 2.3.2.1 绘图功能实现

#### 数据结构

AOE 网络的图形主要由圆形和箭头这两种图形组成, 此程序采用字典类型记录这两种图形数据的记录, 对于圆形需要记录用户给事件赋予的名字、圆心的坐标、入弧集、出弧集; 对于箭头需要记录用户赋予的弧权值、弧头对应的圆的名字、弧尾对应的圆的名字。

#### 绘制图形

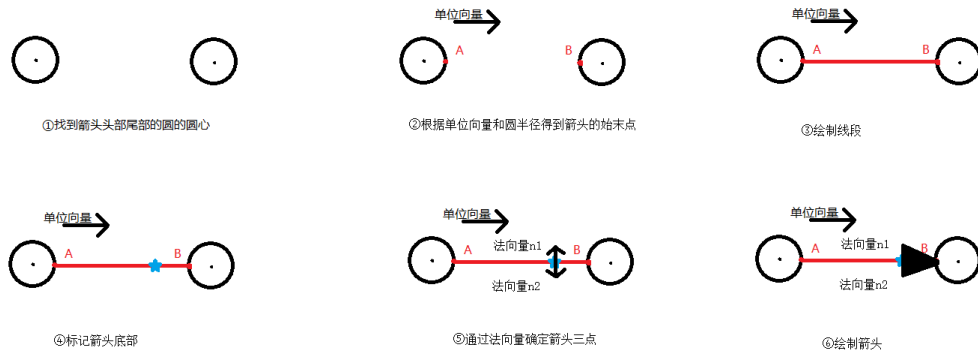
为了美观, 本程序先画所有箭头再画所有圆, 这样可以让圆将箭头遮盖。绘制箭头时先找到箭头头部尾部的圆的圆心坐标, 然后由这两点计算箭头的单位向量, 根据单位向量和圆半径得到箭头的始末点, 然后绘制线段。之后再从线段尾向线段头选取一个单位向量的长度标记箭头底部, 然后通过法向量确定箭头的三点, 最后用填充工具绘制箭

---

<sup>4</sup> 求最值的过程参考自网站资料 《python 使用 max 函数求字典的最大值 (lambda 表达式)》  
[https://blog.csdn.net/sinat\\_38068807/article/details/86021686](https://blog.csdn.net/sinat_38068807/article/details/86021686)



头，最后标注权值到箭头中点<sup>5</sup>。需要注意的是，因为绘制字体无法实现遮盖效果，因此需要先在字体下方绘制一个矩形区域盖住箭头的中点区域再绘制权值。绘制箭头的过程可以参考下图<sup>6</sup>：



绘制圆形的过程相对简单，找到圆心坐标位置后绘制圆形然后将事件名字标注到圆形内即可。需要注意的是，为了让鼠标点击重叠的圆形时选中更上层的圆（即没有被遮盖的圆）需要以逆序的顺序绘制各圆。

### 移动圆形<sup>7</sup>

鼠标左键按下后记录下鼠标指针坐标，然后遍历所有圆形，用点的距离公式  $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} < r$  找到鼠标所点击的圆形。然后记录下该圆形，当鼠标移动时让该圆心不断跟紧鼠标指针。当鼠标左键松开时解除圆形和鼠标的绑定。

### 新建圆形

在鼠标左键单击后记录下鼠标指针坐标，根据该坐标新建圆形加入到数据，在下一次刷新绘图区时便可以展示出该圆形。

### 新建箭头

鼠标左键按下后记录下鼠标指针坐标，然后遍历所有圆形，用点的距离公式  $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} < r$  找到鼠标所点击的圆形。然后记录下该圆形，以该圆的圆心作为弧头，鼠标指针作为弧尾新建箭头到绘图数据，由于此时箭头的终点未确定因此绘制该箭头时绘制的是一条虚线。当鼠标左键松开时，遍历所有圆形，继续用点的距离

<sup>5</sup> 参考自网站资料 《Qt 中获取字体的像素高度和宽度》

<https://blog.csdn.net/hanzengyi/article/details/6305791>

<sup>6</sup> 参考自网站资料 《Qt 学习笔记#9：画一条带箭头的线》

<https://blog.csdn.net/founderznd/article/details/51661523>

<sup>7</sup> 拖拽功能参考自网站资料 《Python 在线【第九节】PyQt 拖拽》 <http://code.py40.com/2035.html>

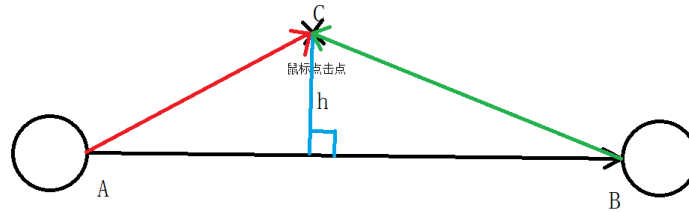
公式找到鼠标所点击的圆形，要求该圆形不能是箭头起点所在的圆，该圆与箭头起点所在圆之间也不能有任何弧。如果找到了点击的圆形就将该圆形作为箭头终点所在圆，更新相关数据后弹出整数输入框让用户输入弧权值，最后刷新绘图区；如果找不到这样的圆，那么箭头新建失败，从绘图数据中删除该箭头。

### 编辑圆的事件名、删除圆

鼠标左键双击后（删除操作是鼠标左键单击）记录下鼠标指针坐标，然后遍历所有图形，用点的距离公式  $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} < r$  找到鼠标所点击的圆形。选定圆形成功后如果使用绘图工具就弹出文本对话框让用户输入事件名称，输入完毕后修改相关数据并更新绘图区。如果使用删除工具就将该圆形从绘图数据中删除。

### 编辑箭头的权值、删除箭头

鼠标左键双击后（删除操作是鼠标左键单击）记录下鼠标指针坐标，然后遍历绘图区箭头判断鼠标选中了哪一条箭头，判断方法是利用向量工具，如下图所示：

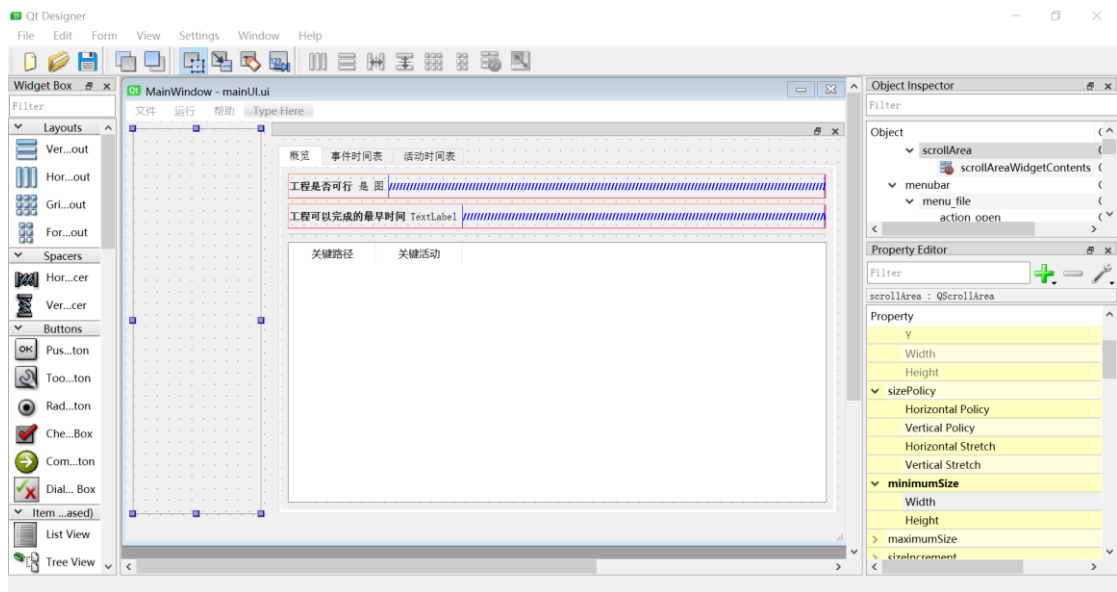


首先要保证鼠标点击点是否离直线足够近，利用公式  $d = |\vec{AC}| \sqrt{1 - \left( \frac{\vec{AB} \cdot \vec{AC}}{|\vec{AB}| |\vec{AC}|} \right)^2}$

计算出鼠标点击点到直线的距离并要求该距离小于指定的距离，除此以外还需要确定鼠标点击到了箭头的实体部分而不是延长线，这就需要鼠标点击点满足  $\vec{AB} \cdot \vec{AC} > 0$  和  $\vec{BA} \cdot \vec{BC} > 0$  这两个条件。选定箭头成功后如果使用编辑工具就弹出整数输入框让用户输入弧权值，更新图形数据刷新绘图区，如果使用删除工具就将该箭头从数据中删除。

### 2.3.2.2 前端设计

在可视化设计软件 QtDesigner 中设计整个界面的布局 and 主要控件的相关属性：



整个页面分为四大部分：菜单栏、绘图区、结果区、状态栏。菜单栏中包含文件、运行、帮助三大功能，文件菜单下包含打开、保存、另存为、另存为图像、保存运行结果、退出这些按钮，运行菜单下包含速览、生成结果两个按钮、帮助菜单下包含使用说明按钮。绘图区包含工具栏和滚动窗口，QtDesigner 不支持对工具栏设定，工具栏的实现需要用代码实现。使用 Scroll Area 控件作为绘图区以便在有限的空间内显示更多内容。结果区是一个停靠窗口，可以根据需求选择停靠位置或者单独为新窗口。该窗口内使用三个标签页显示程序运行结果，第一个标签页是概览，显示工程的可行性、工程完成的最早时间、全部关键路径和对应的关键活动，第二个标签页显示各事件的  $V_e$  和  $V_L$ ，第三个标签页显示各活动的  $e$  和  $l$ 。

### 2.3.2.3 界面功能实现

绘图功能通过 PyQt5 提供的 QPainter 实现，实现鼠标绘图通过对相关 event 函数重写实现。

绘图工具栏功能通过 setDrawStatus 函数实现，根据信号源设定当前的绘图行为。

预览功能通过 preview 函数实现，先检查绘图区内是否为空，如果为空输出错误提示否则调用拓扑排序算法，如果算法返回错误信息则调用错误信息弹窗，否则调用 AOE 网络求解算法将关键路径在绘图区内标红。

生成结果功能通过 run 函数实现，首先完成 preview 所做的工作，然后在状态栏内显示“运行结果已更新”并启用保存结果按钮，然后唤起结果停靠栏并将具体结果显示其中。

打开 AOE 文件功能通过 open\_aoe 函数实现，首先先检查当前绘图区的内容是否保存，如果未保存就弹出对话框让用户选择是否保存当前内容，如果用户选择“是”那么就先调用 save 函数。检查之后先用文件对话框获取文件路径及文件名，调用文件打开函数 read 获取图形数据，之后将图形数据存入绘图窗口中，并将程序标题栏更新为当前文件名，如果文件操作错误就弹出消息提示框显示错误信息。

文件保存功能通过 save 函数实现<sup>8</sup>，首先先检查绘图区数据是否有对应的文件，如果没有就调用另存为函数 save\_as 新建文件；否则调用 write 函数将绘图区数据写入到当前文件并取消标题栏中代表未保存的‘\*’号。如果文件操作错误就弹出消息提示框显示错误信息。

文件另存为通过 save\_as 函数实现，先用文件对话框获取文件路径及文件名，调用 write 函数将绘图区数据写入到指定文件并更新标题栏中的文件名称。如果文件操作错误就弹出消息提示框显示错误信息。

绘图区图像保存通过 save\_image\_as 函数实现，通过文件对话框选定保存路径及文件名，然后导出绘图窗口数据至 QPixmap 对象，最后将 QPixmap 对象保存。

保存运行结果通过 save\_result 函数实现，通过文件对话框选定保存路径及文件名，然后导出结果数据到 Workbook 对象，最后将 Workbook 对象保存。

显示使用功能通过 instruction 函数实现，通过消息对话框展示功能说明文本。

关闭功能通过重写 closeEvent 函数实现。<sup>9</sup>

#### 2.3.2.4 前后端对接

1. 绘图窗口的对接使用 setWidget 方法将绘图窗口放入主窗口的滚动窗口（scrollArea\_tree）中。
2. 按钮和文本框的触发信号通过 connect 函数挂载到对应的功能函数上。

### 2.4 逻辑结构与物理结构<sup>10</sup>

程序中操作的数据主要为 AOE 网络邻接表、逆邻接表、图形数据（圆图形数据和箭头图形数据组成的十字表）、拓扑排序表、关键路径表、关键活动表、Ve/Vl/e/l 表

数据名	数据类型	逻辑结构	物理结构
-----	------	------	------

<sup>8</sup> 具体还需要 pickle 这一部分参考自网站资料《python pickle 模块的使用》  
<https://www.cnblogs.com/lincappu/p/8296078.html>

<sup>9</sup> 参考自网站资料《PyQt5 点击按钮关闭窗口》  
[https://blog.csdn.net/qq\\_27061049/article/details/89394477](https://blog.csdn.net/qq_27061049/article/details/89394477)

<sup>10</sup> 参考自网站资料《Python 的列表(List)的底层实现原理》  
<https://blog.csdn.net/Yuyh131/article/details/83592608>

AOE 网络邻接表	字典	网状结构	表头顶点：散列存储结构 顶点数据域：顺序存储结构。
AOE 网络逆邻接表	字典	网状结构	表头顶点：散列存储结构 顶点数据域：顺序存储结构。
图形数据（十字表）	字典	网状结构	顶点结点：散列存储结构 弧结点：散列存储结构 顶点结点数据域：顺序存储结构。 弧结点数据域：顺序存储结构。
拓扑排序表	列表	线性结构	顺序存储结构
关键路径表	列表	集合结构	顺序存储结构
关键活动表	列表	集合结构	顺序存储结构
Ve/Vl/e/l 表	字典	集合结构	散列存储结构

## 2.5 开发平台

开发平台：Anaconda3 Spyder

开发语言：Python 3

具体版本：conda 4.7.12 python 3.7.1

核心使用库：PyQt5

运行环境：Windows 10（64 位）

## 2.6 开发结果

**正确性：**程序可以正确地画出使用者要求的 AOE 网络，并能求解出该 AOE 网络的正确

关键路径并输出其他相关结果。程序对于 AOE 文件的打开、保存、另存为操作正确。

**稳定性：**经过多组测试，程序能够快速给出正确的结果，程序在运行期间无闪退、崩溃现象发生。可见程序稳定性良好。

**容错性：**程序可以对 AOE 网络的可行性做出分析，可以抛出空网络、非法 AOE 网络的错误信息。如果在操作文件时发生了错误，程序也能弹出相应的错误信息。文件对于正在编辑的文件具有良好的保护功能，在执行关闭窗口、打开文件等操作时会先检查当前文件是否保存，大大减小了因操作不当而丢失当前文件的可能。当主动更改绘图区的 AOE 网络后，程序会自动关闭运行结果窗口，以防旧的运行结果与新的 AOE 网络产生歧义。可见程序的容错性良好。