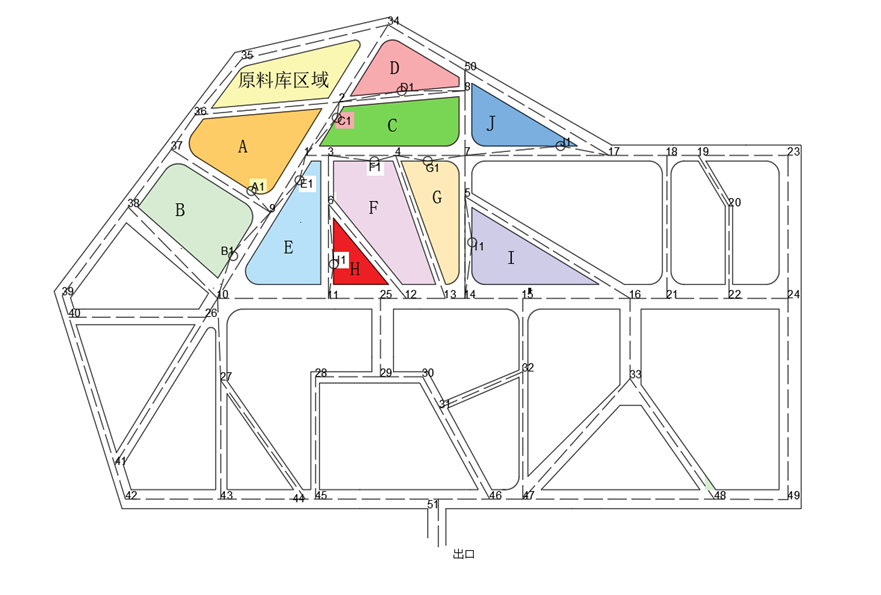
**大作业**

**子问题4**

**模型建立**



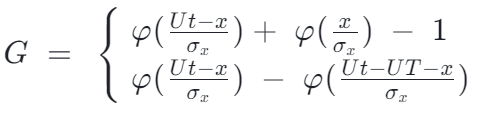
某时刻原料库区域发生原料泄漏，泄漏的盐酸经过挥发泄漏到空气中于是对周围厂房人员的安全造成威胁。

**污染物扩散模型**

工业园区中泄漏的有毒有害气体多数是轻气体以及中性气体，考虑到工业园区的泄漏扩散主要是非正常排放，结合物理现实给出了泄漏扩散预测模型：

其中C为气体浓度；xyz为三维距离；Q为排放源强度。

其中F为混合层反射项，G为非正常排放项，两者的计算式如下：



其中，u为平均风速；t为时间；h为混合层高度；H\_e为泄漏有效高度；K为反射次数；σx、σy、σz为扩散参数。

由于工业园区中盐酸沸点高于环境温度，对于热量蒸发可以忽略不计，因此仅仅计算盐酸的质量蒸发速度：

, n为大气稳定系数；p为液体表面蒸发气压；M为分子量；R为气体常数；T\_0为环境温度；u为风速；r为液池半径。

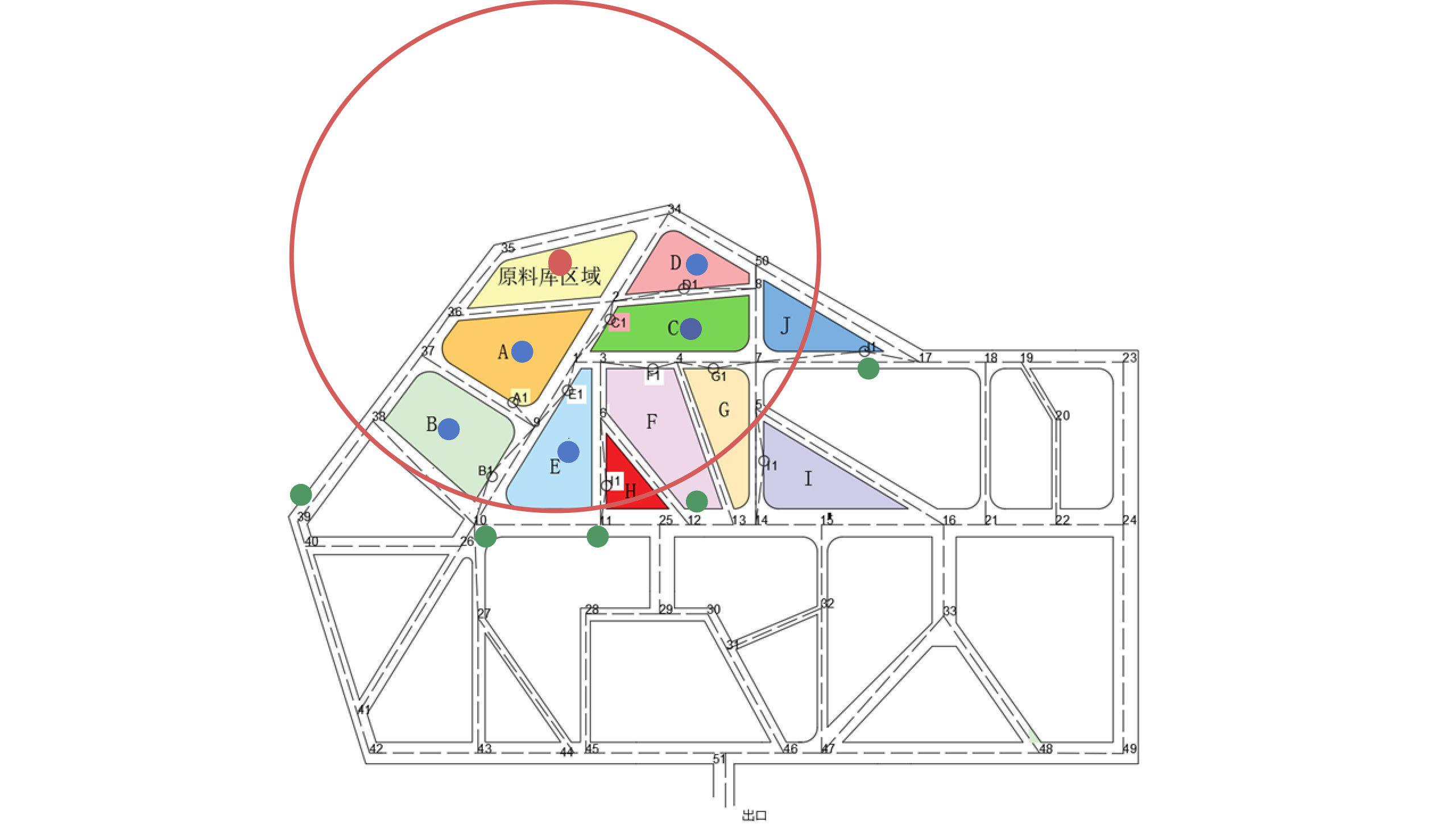
液池半径r可以通过泄露面积进行估算，有：

根据以上污染物扩散模型，可以得到污染物扩散距离、扩散时间和盐酸浓度的一个大致关系（数据来源参考文献）：

可见，在前10min内，仅200m的范围内盐酸浓度超过HCl嗅觉阈值，且考虑到实际的逃生时间，在200m的有害范围内采用前10min的盐酸浓度来换算危害值。

**疏散起点、终点和人群疏散路线的模型**

根据工业园区一般的大小，离原料库泄漏中心200m范围大致可以表示如图（红色圆圈框选处）：



图中，蓝色点代表工人一般聚集的厂房，疏散以这些厂房的大门为起点，分别为节点A1, B1, E1, C1, D1；绿色代表疏散的出口 ，分别为节点39, 10, 11, 12, J1。接下来创建的疏散模型以这几个节点之间的距离关系为基础。

根据数据计算得到的结果表示，在前10min，灾源25m内的盐酸浓度为4.9275mg/m^3；灾源25m~50m的盐酸浓度为2.1351mg/m^3；灾源50m~100m内的盐酸浓度为0.7789mg/m^3（受灾2min后）；灾源100m~200m内的盐酸浓度为0.2295mg/m^3（受灾2min后）。

考虑到受灾后还有反应时间等因素，50m~100m范围和100m~200m范围的盐酸浓度可以直接采用受灾2min后的数据。而其他更远处的盐酸浓度基本在有害阈值以下，因此将距离原料库储料位置200m开外的几个节点作为疏散规划的出口。

本子问题有以下几个设定：

1. 本规划根据盐酸浓度 c 和路径通行能力capacity来**换算危害值，并以此评估路径可通行程度，最终输出结果使得整条规划危害值达到**最小。即将原先规划常用的疏散时间指标加上盐酸浓度的惩罚项，换算成了危害值，规划的总目的是使得总的危害值达到最小。
2. 将每个厂房内的人数进行分批，不同节点的不同人数各自到达最优的一个出口
3. 由于出口具有人流量限制，对于出口的疏散时间和疏散能力也进行考虑。

根据武汉大学与香港大学发展的SGEM模型，人群的疏散速度与人群密度关系可以用以下式子表示：

据此给出了路径上通行波次与人群疏散速度的一个近似公式。

明显地，有通行波次加倍，人群密度加倍。设置一个道路承载能力，根据每条道路实际的路径宽度进行设定。每有一个波次的人通过这条道路，则道路承载能力减去一个设定值，当路径承载能力低于一个阈值时，人群疏散速度不再是1.4，而是根据

其中，m\_i表示此时通过的波次数量；m\_i0表示达到阈值时的波次数量。

作为上述公式中的rho代入式子算出新的疏散速度，再更新道路危害值进行规划。

对于出口的疏散时间，根据每个出口的大小设定各出口的通行能力H\_k，计算每个出口通行能力与所有出口通行能力的比值：

将P\_k乘上疏散总人数M，即得到每个出口对应的理论疏散人数

将理论疏散人数除以疏散能力作为危害值函数惩罚项

**危害值总模型**

除了路径上的污染物危害，人群到达疏散出口时若每个出口疏散人数分配不当，还可能会导致拥堵践踏等问题。

因此不仅要使得路径上受到的危害尽可能小，还必须使得出口人数尽量和每个出口根据通行能力计算出的理论疏散人数 尽量接近。

H和D分别是路径上和门口处的危害值计算函数。

H表示合计整条路径上的危害值大小，c\_i表示根据该条路径中点距离原料库的距离进行计算的盐酸浓度值；v\_i表示当前人群通过当前道路所需要的耗时，可以由\rho进行计算，\rho是路径通行能力和当前经过人数的函数，可以加入当前人群流动速度的惩罚项，当道路能力下降到一定阈值后，后面的波次再经过这条道路会存在风险加成。

其中，

为简化计算，当路段capacity小于5时对速度进行更新，用 来计算拥堵情况下的速度

设定路径上危害系数的计算式子

所以总危害值

其中，t\_i = s\_i / v\_i，即通过这一路段需要的时间，以上计算式即盐酸浓度在时间尺度上的平均值

而对于出口拥堵的处理，为了简化计算，采取将通过人数超过出口理论通行人数的出口节点移出网络。

**算法流程**

根据以上思路，建立了算法流程：

1. Dijkstra算法首先根据路径长度、路径上盐酸浓度，以及逃生者移动速度对每个起点到每个终点的最短路径进行规划（逃生者的移动速度由上一次路径规划最终结果和路段的初始容量得出），程序中具体式子如下（即本规划中的目标函数）：



该式子中线性加和的第一项为路径的长度项；第二项为上述的盐酸危害系数，即盐酸浓度在时间尺度上的平均值H

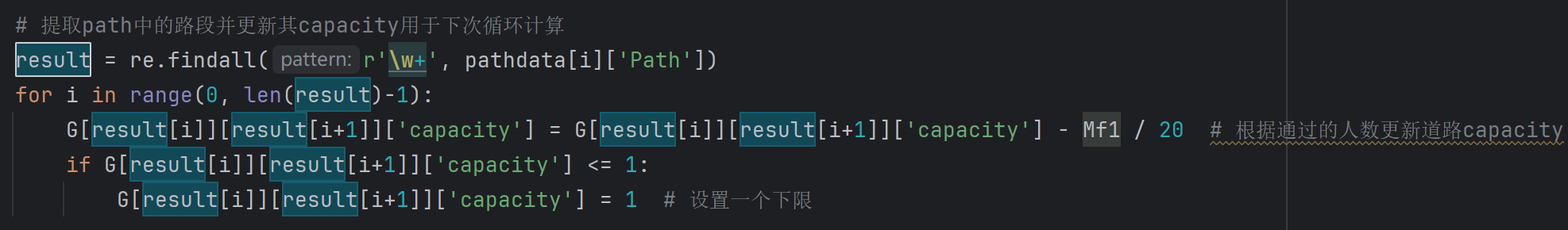
1. 求得的路径目标函数最小值被赋给c列表对应字典项的['Tfk\_v\_road']键，根据['Tfk\_v\_road']键值的大小，从小到大对my\_data\_cp列表中的总共“出口数 \* 入口数”条字典进行了排序。
2. 这一步是对于出口限制人数的处理。

**情况1.** 当对于一条特定字典，节点人数M\_i小于出口人数B\_i时，删除人群节点为s\_i的所有字典，更新出口节点d\_i的剩余人数容量为B\_i = B\_i - M\_i。完成以上步骤后，重复步骤1

**情况2.** 当对于一条特定字典，节点人数M\_i大于出口人数B\_i时，删除出口节点为d\_i的所有字典，更新人群节点s\_i的剩余人数容量为M\_i = M\_i - B\_i。完成以上步骤后，重复步骤1

**情况3.** 当对于一条特定字典，节点人数M\_i等于出口人数B\_i时，同时删除出口节点为d\_i和人群节点为s\_i的所有字典。完成以上步骤后，重复步骤1

步骤1~3为一个循环，步骤1被封装在evacuate函数中（evacuate函数中调用了dijkstra函数进行运算），而步骤2~3被封装在一个adjust函数中，在运算过程中，将循环执行evacuate + adjust函数，每次运行在adjust中均会根据edge通过的人数对graph edge的capacity进行更新：



从而实现在规划中对路段经过人数造成拥堵，进而影响人群移动速度的考虑

说明：

* 本方案创建的graph edge包含3个属性，分别为"weight"（路段长度）, "capacity"（路段通行能力）, "Concentration"（路段盐酸浓度）
* 第一步中，第二项计算时k值直接取为了1；而所需的时间项根据路径的通行能力进行了计算，将逃生者速度设置为了 ，5是通行能力的阈值。式子中的路径通行能力会根据上一次路径规划的结果进行更新，整个流程会重复多次以使得最终规划结果更贴合实际
* 为了更加清楚地说明列表my\_data\_cp列表的结构，这里给出一张截图，是对于上述5个逃生点和5个逃生出口的建模的列表

|  |  |
| --- | --- |
|  | 列表中总共有起点数 \* 终点数个字典。  遍历了每个起点到每个终点的情况 |

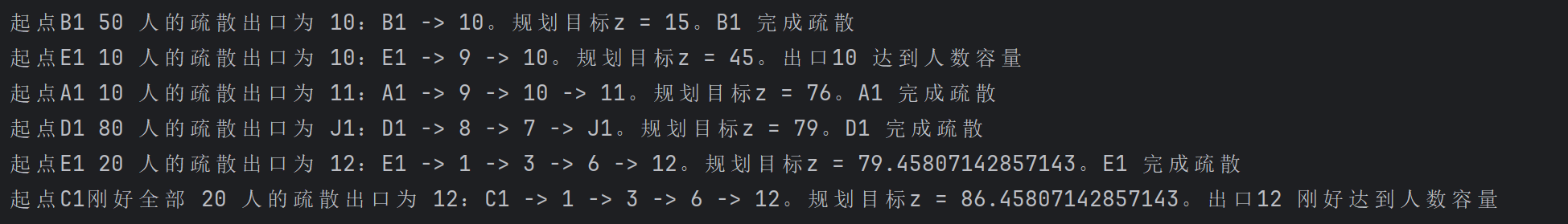
**运行效果**

本实验设置5个疏散起点：

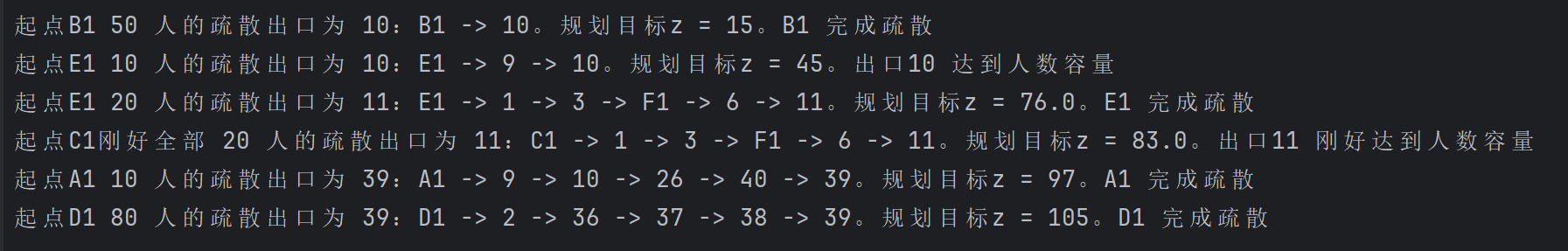
5个出口

设置循环3次运行得到的结果如下：

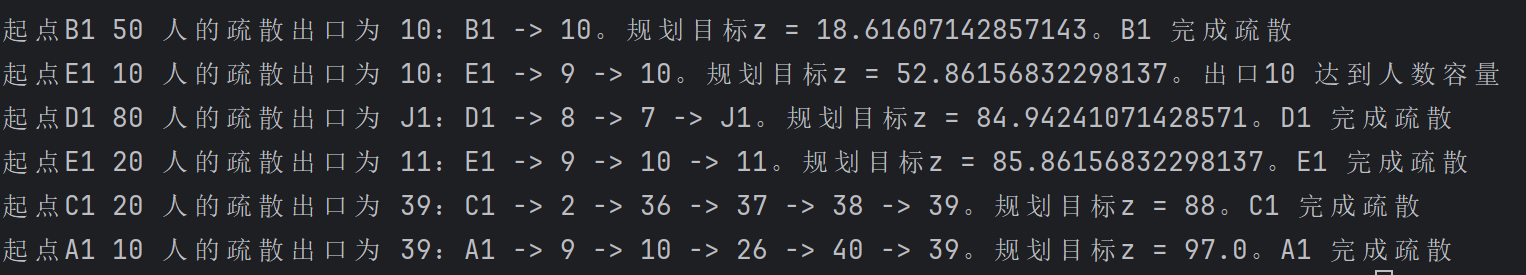
Time1



Time2



Time3



实现了在对逃生时间、泄漏物危害两个因素的考虑下进行路径规划，同时在考虑出口容量的前提下对每个起点的逃生人员进行了出口人数分配；此外，加入了路径上人员拥堵造成通过时间变长导致盐酸危害加大的因素并实现了整体的优化。