

# 基于 Dijkstra 算法的矿井突水水流漫延模型与逃生方案研究

## 摘要

针对矿井水灾问题,本文研究矿井突水漫延和矿工逃生的数学模型,为矿井安全管理提供科学依据。

针对问题一,首先对附件 1 和附件 2 进行数据预处理和可视化分析,接着通过巷道连通图分析建立巷道中水流速度、巷道间每点的长度及巷道水流充满模型的数学模型,分别为

$V = \frac{E}{A} = \frac{30}{0.4} = 75\text{m/min}$ ,  $L = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$ ,  $T = t_0 + \frac{4(H - h_0)L}{E_1}$ 。最后,结合最短路 Dijkstra 算法,运用 python 软件进行求解,得出附件 1 和附件 2 巷道中突水点到每一点的时刻和水流充满的时间。如附件一中端点水流到达最早编号是 P0356,时间是 0.36min,端点水流到达最晚编号是 P0404,时间是 99.05min,巷道充满水最早编号是 H0994,时间是 10.50min,巷道充满水最晚编号是 H0755,时间是 509.87min,

针对问题二,在问题一的水流漫延模型,对问题一的 Dijkstra 算法进行优化,计算出巷道的实时水位  $H_t = \frac{E_1 \times (T - t_{初})}{w \times L_{巷}}$ 。根据计算结果判断是否能满足通行的约束条件,再通过

比较所有矿井出入口节点的到达时间,计算候选到达时间  $t_{cand} = t_{cur} + T$ ,以出入口作为起点,依次找出巷道的前驱节点与对应巷道,直到回溯到逃生起点,将得到节点和巷道反向序列,将其反转,可计算出各矿工的最优逃生路径。

针对问题三,针对矿井双突水点水流漫延问题,基于附件 1、2 的矿井端点(X/Y/Z 坐标)与巷道拓扑数据,结合水量均衡理论与改进 BFS 算法,允许坡度小于 10%的缓坡上行来增强数据结果的现实性,流速按“基础流速(2m/s) × (1-坡度占比)”折减,以此来构建双突水点漫延模拟模型  $Q_t = 30 \times (t - t_0 - t_f) + 30 \times (t - (t_0 + \Delta t) - t_f)$ 。

针对问题四,矿井双突水点动态水流漫延场景,需在第二突水点突水 1 分钟后,为 6 名矿工(附件 1、附件 2 各 3 名)规划逃生路径。通过构建双突水水流漫延模型与改进 Dijkstra 动态路径规划模型,结合水流速度、水位对人员通行的时变约束,最终输出各矿工的最优逃生路径。

本文具有以下特色与亮点是每一个问题所建立的模型普遍简单,具有很强的推广下。

**关键词:** 矿井突水 水流漫延模型 Dijkstra 算法 路径规划 python

## 一、问题重述

矿井突水灾害作为矿山安全开采的首要威胁,具有突发性强,危害程度大,影响范围广,极易引发重大人员伤亡与财产损失。本文通过构建智能化的矿井突水逃生路线决策保障矿工的安全。

(1) 根据附件 1 和附件 2 端点坐标计算出端点水流到达时刻和巷道充满水时刻,并保存到文件 result1-1 和 result1-2 中。

(2) 根据问题一所得出的水流漫延模型,规划出每个矿工的逃生方案,并保存到文件 result2-1 和 result2-2 中。

(3) 根据附件 1 和附件 2 给出的巷道网络,计算双突水点水流漫延模型,并计算出端点水流到达时刻和巷道充满水时刻,并保存到文件 result3-1 和 result3-2 中。

(4) 基于问题 3 的双点突水漫延模型,在第二突水点突水 1 分钟后发布调整逃生方案通知,为处于巷道中的矿工重新规划最佳逃生路径,结果保存至 result4-1.xlsx 与 result4-2.xlsx。

## 二、问题分析

### 2.1 问题一分析

针对问题一,通过查看相关文献知道水流在巷道中的传播本质是一个一维流动问题,可以通过经典水力学理论求解,水流速度与流量和断面面积有直接相关性,可知, $E_{\text{突水量}}$ 与过流断面面积  $A$  和水流速度  $V$  之间满足  $E_{\text{突水量}} = A \times V$ ,计算过流断面面积需要考虑实际水深,当水深为  $h_0$  时,过断面的面积  $A$  的计算公式为  $A = w \times h_0 = 4h_0$ ,通过计算可得水流的平均速度。为了计算端点水流到达时刻和巷道充满时刻可以利用巷道两端点三维坐标计算实际长度,设端点坐标为  $(X1, Y1, Z1)$  与  $(X2, Y2, Z2)$ ,则长度。结合前文计算的水流速度  $V$ ,可得到无分流情况下水流从巷道一端到达另一端的初始时间(公式为  $t=L/V$ );若存在分流节点(设节点处有  $k$  个分支),则需按“平均分流”规则调整流量( $E_{\text{分支}} = E_{\text{总}}/k$ ),对于计算巷道充满时刻可以从水深变化,可知当水位从  $h_0$  上升到  $H$  时,有  $V = 4 \times (3 - 0.1) \times L_{\text{巷}}$ ,则巷道充满时刻可表示为  $\tau = t_0 + \frac{4 \times (3 - 0.1) \times L_{\text{巷}}}{V}$ ,最后将其数学模型转换为 python 代码,并计算出端点水流到达时刻和巷道充满时刻。

### 2.2 问题二分析

问题 2 相较于问题一新增了动态水流约束的核心特征,由于水位上升导致速度变化,因

此巷道通行时间是随进入时刻动态变化的, 然后对附件 1 和附件 2 中所构建的无向图确定各矿工在不同水位状态的行进速度, 再通过使用改进的最短路 Dijkstra 算法, 可解出巷道中各矿工的最佳逃生方案。

### 2.3 问题三分析

问题 3 研究双点突水的水流漫延, 分两阶段模拟: 先仅首个突水点工作, 按问题 1 模型计算; 待第二个突水点启动后, 叠加两突水点的分流流量与充水体积, 同时遵循水流优先级判定规则, 即同一巷道取较早到达时刻, 充水体积叠加后若满足充满条件则重新计算充满时刻, 以此得到双点突水时各巷道的水流到达与充满时刻。

### 2.4 问题四分析

问题 4 基于问题 3 双点突水的漫延结果, 在第二突水点突水 1 分钟后, 重新评估各巷道因水流叠加后的水位状态, 更新对应的边权重, 排除新增不可通行的高水位巷道, 再次运用改进的 Dijkstra 算法, 为矿工重新规划最佳逃生路径, 应对双点突水带来的更复杂逃生环境。

## 三、符号说明

$E_{\text{突水量}}$	突水点的突水量
$A$	断面面积
$V_{\text{水}}$	水流的平均速度
$w$	断面的宽度
$h_0$	突水水流的初始水位
$L_{\text{巷}}$	巷道的长度
$t_0$	端点水流到达时刻
$E_i$	每个分支的流量
$E_{\text{总}}$	节点处的水流总量
$k$	分流节点
$H$	断面的高度
$T$	充满水的时刻
$V_{\text{总}}$	巷道总容积

## 四、模型假设

1. 假设突水点在各巷道的水流流向为平均分流且为匀速直线运动。
2. 假设工人具有基本逃生能力，且工人的前进速度为匀速直线运动。
3. 假设忽略水压对巷道变形的影响。
4. 假设在巷道中信号接收正常，使工人能实时接收到逃生路线。

## 五、模型的建立与求解

### 5.1 问题一的模型建立与求解

#### 5.1.1 数据预处理和可视化分析

对附件 1 和附件 2 中的数据进行预处理，以矿井端点为节点，巷道为边，使用 MATLAB 软件绘制出附件 1 和附件 2 的巷道网络三维节点连通图（见图 1 和图 2）。根据绘制出来的连通图可视化分析，可以观察得出附件 1 和附件 2 中数据没有缺失和出现异常值现象。因此，我们认为附件 1 和附件 2 的数据是可行的。

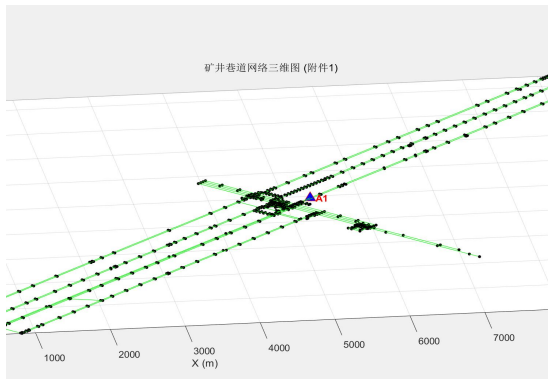


图 1 附件 1 的三维无向图

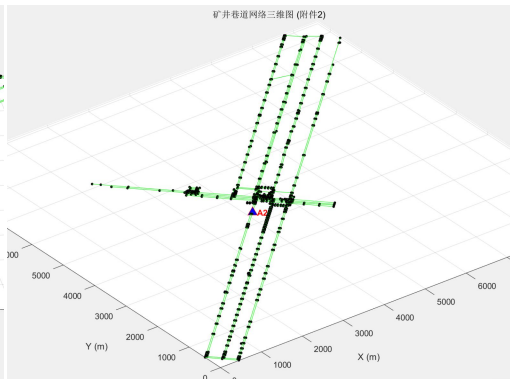


图 2 附件 2 的三维无向图

#### 5.1.2 巷道水流漫延和充满模型

##### 1. 巷道水流蔓延模型

###### (1) 水流速度计算

为了求解巷道水流蔓延，我们首先计算出水流的速度，由于突水量为  $E_{\text{突水量}} = 30 \text{ m}^3/\text{min}$ ，断面 A 是矩形面积。我们可以建立水流速度的数学模型，即  $E_{\text{突水量}}$  与过流断面 A 和水流速度  $V$  之间满足： $E_{\text{突水量}} = A \times V_{\text{水}}$ 。

对于矩形断面巷道，其矩形面积为巷道的断面宽度乘以突水水流的高度，即

$$A = w \times h_0 = 4 \times 0.1 = 0.4$$

根据上述结论可计算出水流在巷道的初始漫延速度  $V_{\text{水}} = \frac{E_{\text{突水量}}}{A} = \frac{30}{0.4} = 75(m/\text{min})$

## (2) 巷道间两点长度的计算模型

假设巷道内水流的速度是恒定的，即为 75m/min。为了求出突水点到达和灌满巷道的所有点时间，需要先求出巷道内任意两点间的距离，即可设巷道两端点坐标为

$p_1(x_1, y_1, z_1)$  和  $p_2(x_2, y_2, z_2)$ ，则根据欧氏距离公式，可得巷道内任意两个点间的长度为：

$L_{\text{巷}} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$ 。从而可得水流到达巷道每一个端点的初始时间：

$$t_{\text{初}} = \frac{L_{\text{巷}}}{V_{\text{水}}}$$

根据以上公式，可以计算出的水流在无分流情况下，从巷道一端到达另一端的时间。如果巷道网络中，出现分流情况，我们假设水流到达有 k 个分流节点时，根据题目假设可知，水流将平均分配到各个分支。设节点处的水流总量为  $E_{\text{总}}$ ，则每个分支的流量为：

$$E_i = \frac{E_{\text{总}}}{k}, i = 1, 2, 3, \dots, k$$

综合以上分析，接下来，我们结合最短路 Dijkstra 算法，运用 python 代码对附件 1 和附件 2 运算分析，可以得出附件 1 和附件 2 巷道中突水点到达每个点时刻，具体见表 1 和表 2。

## 2. 巷道水流充满模型

巷道充满水过程中，需要考虑水位上升时间，当水流流到巷道末端时，该巷道开始蓄水，其中  $R = w \times L \times h(t)$  是巷道的蓄水体积，设  $E_1$  和  $E_2$  分别是流入和流出该巷道流量，当  $E_2 = 0$  时，表示该巷道开始蓄水，则水位上升速率为：

$$\frac{h}{t} = \frac{E_1}{w \times L}$$

当水流充满巷道时，则充满水时刻为

$$T = t_0 + \frac{4(H - h_0)L}{E_1}$$

综合以上分析，我们继续结合最短路 Dijkstra 算法，运用 python 软件对附件 1 和附件 2 运算分析，可以得出附件 1 和附件 2 巷道中水流充满每一个巷道点的时刻，具体见表 1 和表 2。

综合以上分析，我们继续结合最短路 Dijkstra 算法<sup>[8]</sup>，运用 python 软件对附件 1 和附件 2 运算分析，可以得出附件 1 和附件 2 巷道中水流充满每一个巷道点的时刻，具体见表 1 和表 2。

**表 1 result1-1 水流漫延到各个点的时刻表**

端点编号	端点水流到达时刻 (min)	巷道编号	巷道水流充满时刻(min)
P0000	25.17	H0000	45.09
P0001	25.82	H0001	85.48
P0002	27.13	H0002	85.48
P0003	28.98	H0003	15.14
P0004	18.08	H0004	21.89
P0005	9.55	H0005	21.24
P0006	9.35	H0006	23.26
P0007	10.73	H0007	251.38
P0008	9.08	H0008	251.38
.....	.....	.....	.....
P0660	5.32	H0974	41.32
P0661	6.42	H0975	29.69
P0662	6.28	H0976	25.27

(具体结果详见支撑材料 result1-1 )

**表 2 result1-2 水流漫延到各个点的时刻表**

端点编号	端点水流到达时刻 (min)	巷道编号	巷道水流充满时刻(min)
P0000	25.17	H0000	3959.89
P0001	25.82	H0001	40352.56
P0002	27.14	H0002	40362.15
P0003	28.99	H0003	26977.86
P0004	18.09	H0004	26797.53
P0005	9.56	H0005	9936.55
P0006	9.36	H0006	9824.04
P0007	10.73	H0007	1721.21
P0008	9.09	H0008	1721.21
.....	.....	.....	.....
P0660	5.32	H0974	3959.89
P0661	6.42	H0975	25626.68
P0662	6.28	H0976	11696.59

(具体结果详见支撑材料 result1-2 )

## 5.2 问题二的模型建立与求解

基于问题 1 的水流漫延模型，我们运用改进的最短路 Dijkstra 算法（时间依赖型 Dijkstra 算法）对问题二中矿工最佳逃生路径的进行计算，具体执行步骤如下：

### 1. 计算巷道的实时水位

为了求解出在突水 1 分钟时，附件 1、附件 2 矿井中的各矿工设计最佳逃生路径，本文首先通过（图 1 和图 2）确定连接二者的目标巷道，并根据当前时刻计算出该巷道的实时水位：

$$H_t = \frac{E_i \times (T - t_{初})}{w \times L_{巷}}$$

### 2. 工人在巷道水位通行的约束条件

然后依据实时水位判断出巷道是否可以通行，若为高水位（水面大于 0.3m），则需跳过该巷道，若为无水流，则按工人速度为 4m/s 计算，若为低水位（水面小于等于 0.3m），则按照工人顺水速度为 2m/s 和逆水速度为 1m/s，从而计算出满足水位通行要求的通过目标巷道时间。

### 3. 计算候选到达时间

再根据满足水位通行要求的通过目标巷道时间计算出各巷道邻居节点的候选到达时间，若候选时间小于邻居节点的最早到达时间，则更新邻居节点为候选时间，即：

$t_{cand} = t_{cur} + T$ ，其中  $t_{cur}$  为当前节点到达时间， $T$  为巷道通行时间并在代码的前驱节点字典中记录该邻居节点的前驱巷道，同时将此候选到达时间和邻居节点加入后续的优先队列，直至迭代到优先队列为空。

### 4. 迭代更新节点最优路径

等待迭代过程结束后，提取所有矿井的出入口节点的最早到达时间，选择时间最短的出入口最为矿工最优逃生终点，再从终点出发，反向查看前驱节点字典，依次找出终点的前驱节点与对应巷道，直到回溯到逃生起点，此时将得到节点和巷道反向序列，最后再将此反向序列再反转，即可得到各矿工最佳逃生路径。

## 5. 代码运行结果

综合以上分析，我们继续结合时间依赖型 Dijkstra 算法，运用 python 软件对附件 1 和附件 2 运算分析，可以得出附件 **result2-1** 和附件 **result2-2** 巷道中水流充满每一个巷道点的时刻，具体见表 3 和表 4。

将上述逻辑转化成 python 代码（详见支撑材料 22.py）并在在 PyCharm 软件进行编译,最后再根据代码模型将附件 1 和附件 2 数据代入 python 代码中运行可得到如下结果:

表 3 result2-1 工人逃生路径表（部分）

工人 1 逃生路径						
序号	端点编号	x (m)	y (m)	z (m)	到达时刻	前序巷道
0	——	5808.18	5367.75	10	1.00	——
1	P0659	5801.85	5321.84	10	1.23	H0966
2	P0357	5511.93	5048.24	10	2.89	H0963
3	P0352	5225.86	4773.33	10	4.55	H0398
4	P0350	5204.47	4752.97	10	4.67	H0391
.....	.....	.....	.....	.....	.....	.....
39	P0395	3470.58	3532.61	10	14.6	H0646
40	P0392	3325.07	3395.97	10	15.43	H0454
41	P0393	3252.16	3326.62	10	15.85	H0452
工人 2 逃生路径						
0	——	5194.00	4785.31	10	1.00	——
1	P0349	5182.93	4774.75	10	1.13	H0389
2	P0348	5154.54	4747.39	10	1.29	H0385
3	P0332	5116.99	4711.19	10	1.51	H0387
4	P0330	5106.55	4701.13	10	1.57	H0369
.....	.....	.....	.....	.....	.....	.....
27	P0395	3470.58	3532.61	10	10.98	H0646
28	P0392	3325.07	3395.97	10	11.81	H0454
29	P0393	3252.16	3326.62	10	12.23	H0452
工人 3 逃生路径						

0	——	6190.81	3434.29	10	1.00	——
1	P0024	5965.98	3598	10	2.1	H0733
2	P0022	5950.58	3615.75	10	2.2	H0019
3	P0023	5931.56	3635.98	10	2.31	H0017
4	P0121	5916.81	3651.45	10	2.4	H0106
.....	.....	.....	.....	.....	.....	.....
37	P0395	3470.58	3532.61	10	16.69	H0646
38	P0392	3325.07	3395.97	10	17.52	H0454
39	P0393	3252.16	3326.62	10	17.94	H0452

(具体结果详见支撑材料 result2-1 )

表 4 result2-2 工人逃生路径表 (部分)

工人 1 逃生路径						
序号	端点编号	x (m)	y (m)	z (m)	到达时刻	前序巷道
0	——	4395.15	4614.53	6.59	1.00	——
1	P0349	4406.22	5321.84	6.59	1.13	H0389
2	P0348	4434.6	5048.24	6.62	1.29	H0385
3	P0332	4472.16	4773.33	6.8	1.51	H0387
4	P0330	4482.59	4752.97	6.81	1.57	H0369
.....	.....	.....	.....	.....	.....	.....
27	P0395	6118.56	3532.61	71.41	11	H0646
28	P0392	6264.08	3395.97	50.62	11.84	H0454
29	P0393	6336.99	3326.62	36.15	12.26	H0452
工人 2 逃生路径						
0	——	3398.34	5965.56	1.31	1.00	——
1	P0024	3623.17	5801.84	0.1	2.1	H0733
2	P0022	3638.57	5784.1	1.3	2.2	H0019
3	P0023	3657.59	5763.87	0.47	2.31	H0017

4	P0121	3672.34	5748.4	0.47	2.4	H0106
.....	.....	.....	.....	.....	.....	.....
37	P0395	6118.56	5867.24	71.41	16.71	H0646
38	P0392	6264.08	6003.87	50.62	17.54	H0454
39	P0393	6336.99	6073.22	36.15	17.97	H0452
<b>工人 3 逃生路径</b>						
0	——	3879.44	4125.47	6.22	1.00	——
1	P0358	4096.18	4331.93	6.28	2.66	H0968
2	P0351	4384.08	4603.97	6.59	4.31	H0399
3	P0349	4406.22	4625.09	6.59	4.44	H0389
4	P0348	4434.6	4652.45	6.62	4.61	H0385
.....	.....	.....	.....	.....	.....	.....
29	P0395	6118.56	5867.24	71.41	14.32	H0646
30	P0392	6264.08	6003.87	50.62	15.15	H0454
31	P0393	6336.99	6073.22	36.15	15.58	H0452

（具体结果详见支撑材料 result2-2 ）

### 5.3 问题三的模型建立与求解

为简化工程问题、聚焦关键矛盾，结合矿井突水实际场景与附件数据特征，提出以下假设：

1. 允许缓坡向上漫延，最大允许坡度  $\max\_slope=0.1$ （10%），陡坡（坡度  $> 10\%$ ）因水流爬升能力有限，视为不可漫延；
2. 水流在巷道内为明渠流，不考虑管壁摩擦损耗外的其他能量损失，基础流速取水平巷道实测均值 2 m/s；
3. 附件 1 中所有端点高程  $Z=10$  m，默认巷道水平；附件 2 按实际  $Z$  坐标计算坡度，不额外简化。

基于此，利用基于附件 1 和附件 2 的“端点 - 巷道”数据，构建矿井空间拓扑关系，核心计算以下基础参数：

**巷道长度计算模型：**通过端点三维坐标计算空间距离，公式为：

$$L_{\text{巷}} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}。$$

由于附件 2 中 z 轴并不相同所以我们要根据高度差来进行巷道流向判断根据端点高程差： $\Delta z = z_2 - z_1$ 。

当  $\Delta z < -0.01$  时为下巷道；当  $\Delta z \leq 0.01$  时为水平巷道；当  $\Delta z > 0.01$  时为向上巷道（按照  $2 \times (1 - \Delta z)$  衰减）。

考虑到模型的复杂程度，我们将问题简化忽略掉影响水流因素的复杂计算，用上述公式判断巷道类型依此计算出每条巷道的水流速度。当一个端点连接 k 条水平/下巷道时则每条巷道的分流量是： $Q_d = \frac{Q_t}{k} = \frac{30}{k}$ 。

随后我们需要知道突水点 1 的起始时间  $t_0$  以及突水点 2 的起始时间  $t_0 - \Delta t$ 。由于两个突水点都会流经某一端点所以我们去最短时间为到达某点的时刻

$t_{a-t} + \min(t_0 + t_r, t_0 + \Delta t + t_{r1})$  ( $t_r, t_{r1}$  分别代表突水点 1 和突水点 2 到达该点的相对时间) 当两个突水点水流都到达某一巷道时则水量会叠加：

$$Q_t = 30 \times (t - t_0 - t_f) + 30 \times (t - (t_0 + \Delta t) - t_f)。$$

综合以上分析，我们继续结合最短路 Dijkstra 算法，运用 python 代码对附件 1 和附件 2 运算分析，可以得出附件 1 和附件 2 巷道中水流充满每一个巷道点的时刻，具体见表 5 和表 6。

**表 5 result3-1 水流漫延到各个点的时刻表**

端点编号	端点水流到达时刻 (min)	巷道编号	巷道水流充满时刻 (min)
P0000	15.73	H0000	43.36
P0001	16.16	H0001	81.42
P0002	16.96	H0002	449.63
P0003	18.21	H0003	21.18
.....	.....	.....	.....
P0661	4.01	H0975	34.02
P0662	3.92	H0976	42.79

(具体结果详见支撑材料 result3-1 )

表 6 result3-2 水流漫延到各个点的时刻表

端点编号	端点水流到达时刻 (min)	巷道编号	巷道水流充满时刻(min)
P0000	15.8	H0000	45.48
P0001	16.23	H0001	84.51
P0002	18.91	H0002	452.39
P0003	20.19	H0003	23.58
.....	.....	.....	.....
P0661	4.01	H0975	36.74
P0662	3.92	H0976	45.5

(具体结果详见支撑材料 result3-2 )

## 5.4 问题四的模型建立与求解

### 5.4.1 双突水水流漫延模型与求解

以“双源广度优先搜索(BFS)”模拟水流在巷道网络的蔓延过程,为路径规划提供实时巷道通行状态。我们先做如下假设:

#### 1. 核心假设与参数

巷道为矩形断面(宽 4m、高 3m,横截面积  $12m^2$ ),单突水点突水量  $30m^3/min$ ,双源水流独立叠加;

水流仅向水平 / 下行巷道(端点高程 $\leq$ 当前节点)分流,初始水位恒为 0.1m。

**巷道长度:**  $L_{巷} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$  (为巷道端点坐标)。

**水力坡度:**  $J = \frac{|\Delta z|}{L}$  ( $\Delta z$  为巷道两端高程差)。

**水位随时间变化:** 巷道累计水量  $V_{水}(t) = Q_{入}(t) \Delta t - Q_{分}(t)$ , 水位  $h(t) = \frac{V_{水}(t)}{4L} (h \leq 3m)$ 。

具体双源 BFS 传播算法如下:

对 A1/A2、B1/B2 两个突水点,分别以其最近巷道端点为起点,BFS 遍历巷道网络;计算每条巷道的水流到达时刻  $t$  到达和充满时刻  $t$  充满,生成水位函数  $h(t)$ ,刻画巷道通行状态的时变规律。

#### 2. 水流漫延模型求解

以附件 1 (A1 突水 5 分钟后、B1 突水 1 分钟)和附件 2 (A2 突水 6 分钟后、B2 突

水 1 分钟) 为输入, 运行双源 BFS 算法:

(1) 对 A1 (坐标(5349.03, 4931.90, 10.00)) 和 B1 (坐标(3760.40, 3808.33, 10.00)), 计算各巷道的  $t$  到达、 $t$  充满及  $h(t)$ ;

(2) 对 A2 (坐标(4143.12, 4376.28, 6.33)) 和 B2 (坐标(5883.14, 5643.35, 40.37)), 重复上述计算, 得到双突水叠加后的巷道状态。

#### 5.4.2 动态逃生路径规划模型

基于以上算法, 我们对此进行动态分析, 将巷道网络抽象为时变加权图, 节点为巷道端点, 边为可行巷道, 边权为人员通行时间, 采用改进 Dijkstra 算法求解最短逃生时间路径。

##### 1. 时变网络定义

节点集  $V$ : 巷道端点 (含矿工初始位置映射点、出入口);

边集  $E(t)$ :  $t$  时刻满足 “未充满 ( $t < t_{\text{充满}}$ ) 且水位  $h(t) \leq 0.3 \text{ m}$ ” 的巷道;

边权 (通行时间):  $t_{\text{通行}} = \frac{L}{v}$  人, 人员速度规则如下:

巷道状态	通行速度 (m/s)
无水流 ( $t < t_{\text{到达}}$ )	4
水位 $\leq 0.3 \text{ m}$ (顺水)	2
水位 $\leq 0.3 \text{ m}$ (逆水)	1
水位 $> 0.3 \text{ m}$ 或充满	不可通行

##### 2. 改进 Dijkstra 算法

传统 Dijkstra 算法无法处理时变权重, 因此进行以下改进: 优先队列存储 “时间偏移量 + 当前节点 + 到达绝对时间”; 遍历巷道时, 实时验证每一条路线是否都不会超过 0.3m。

##### 3. 动态路径规划求解

针对附件 1 的 3 名矿工（初始位置如(5808.18, 5367.75, 10.00)）和附件 2 的 3 名矿工（初始位置如(4395.15, 4614.53, 6.59)），以“第二突水点突水后 1 分钟”为逃生起始时刻，运行改进 Dijkstra 算法：

- (1) 对每个矿工，找到其初始位置最近的巷道端点作为起始节点；
- (2) 在时变网络中搜索至出入口的最短时间路径，逐段验证巷道通行可行性；
- (3) 输出路径的节点序列、总逃生时间（秒转换为分钟）及所经巷道编号。

根据以上分析，运用两个模型进行计算，运用 python 软件进行求解<sup>[8]</sup>，得出如下附件 1 和附件 2 矿工逃生的最优路线，具体结果如下：

表 6 result4-1 矿工逃生最优路线

矿工 ID	初始位置 (X, Y, Z)	目标出入口 (X, Y, Z)	最佳路径（端点序列）	总逃生时间 (秒)	路径可行性
附件 1-矿工 1	5808.18, 5367.75, 10.00	3252.16, 3326.63, 10.00	P0661→P0659→P0357→P0352→P0350→P0347→P0345→P0658→P0653→P0651→P0652→P0053→P0338→P0641→P0625→P0317→P0316→P0430→P0313→P0315→P0314→P0637→P0635→P0193→P0195→P0645→P0644→P0419→P0420→P0170→P0169→P0649→P0160→P0151→P0147→P0150→P0153→P0520→P0518→P0395→P0392→P0393	890.78	可行
附件 1-矿工 2	5194.00, 4785.31, 10.00	3252.16, 3326.63, 10.00	P0351→P0349→P0348→P0332→P0330→P0331→P0052→P0327→P0627→P0319→P0321→P0416→P0648→P0185→P0181→P0182→P0631→P0170→P0169→P0649→P0160→P0151→P0147→P0150→P0153→P0520→P0518→P0395→P0392→P0393	674.04	可行

附件1-矿工3	6190.81, 3434.29, 10.00	3252.16, 3326.63, 10.00	P0010→P0024→P0022→P0023→P0121→P0126→P0369→P0606→P0607→P0611→P0612→P0370→P0139→P0137→P0134→P0133→P0136→P0201→P0561→P0205→P0365→P0367→P0342→P0566→P0565→P0161→P0155→P0163→P0164→P0158→P0149→P0146→P0148→P0150→P0153→P0520→P0518→P0395→P0392→P0393	1016.32	可行
---------	-------------------------	-------------------------	---	---------	----

表7 result4-2 矿工逃生最优路线

矿工ID	初始位置(X, Y, Z)	目标出入口(X, Y, Z)	最佳路径(端点序列)	总逃生时间(秒)	路径可行性
附件2-矿工1	4395.15, 4614.53, 6.59	6416.05, 6579.88, 8.69	P0351→P0349→P0348→P0332→P0330→P0331→P0052→P0652→P0053→P0338→P0564→P0359→P0341→P0340→P0342→P0566→P0154→P0156→P0526	716.22	可行
附件2-矿工2	3398.34, 5965.56, 1.31	6416.05, 6579.88, 8.69	P0010→P0024→P0022→P0023→P0121→P0126→P0369→P0606→P0607→P0611→P0612→P0370→P0139→P0137→P0134→P0133→P0136→P0201→P0561→P0205→P0365→P0367→P0342→P0566→P0154→P0156→P0526	1040.33	可行
附件2-矿工3	3879.44, 4125.47, 6.22	6416.05, 6579.88, 8.69	P0660→P0358→P0351→P0349→P0348→P0332→P0330→P0331→P0052→P0652→P0053→P0338→P0564→P0359→P0341→P0340→P0342→P0566→P0154→P0156→P0526	915.03	可行

## 六、模型的评价与推广

### 6.1 模型的优点

1. 利用无向图理论精确描述了水流传播过程，具有推广性。

2. 考虑了动态速度变化，逃生路径更符合实际情况。
3. 所使用的的双点突水模型有效处理了水流叠加的效应。

## 6.2 模型的缺点

人员速度模型较为简单，未考虑体力因素。

## 6.3 模型的推广

本模型不仅可以用于本赛题，也同样适用于矿井通风系统、类网络流等问题。

### 参考文献

- [1] 苑亚南, 朱希安, 王占刚. 矿井水灾多逃生路径模型应用研究[J]. 中国矿业, 2019, 28(01):135-140.
- [2] 符辉. 矿井突水及避灾仿真算法的研究与实现[D]. 中国矿业大学(北京), 2013.
- [3] 刘欣倩. 煤矿突水漫延过程及最佳避灾路线算法的研究[D]. 西安建筑科技大学, 2016. DOI:10.27393/d.cnki.gxazu.2016.000244.
- [4] 周越, 朱希安, 王占刚. Dijkstra 算法在矿井水灾动态避灾路径中的改进与应用[J]. 煤炭工程, 2019, 51(03):18-22.
- [5] 史东超. 基于 EMD-BFS-ELM 的径流量预测方法研究[J]. 陕西水利, 2025(2):22-25.
- [6] 王庆众. 基于三维巷道网络的矿井突水灾害救援路径动态规划方法[D]. 山东科技大学, 2019. DOI:10.27275/d.cnki.gsdku.2019.000302.
- [7] 谢彪, 朱登奎, 李柏辰, 等. 基于动态 D-K 算法的矿井突水应急疏散最优路径研究[J]. 煤矿安全, 2024, 55(06):192-199. DOI:10.13347/j.cnki.mkaq.20231697.
- [8] DeepSeek, DeepSeek-V3, 深度求索, 2025-9-7.
- [9] 豆包, 豆包-V1.47.4, 字节跳动, 2025-9-6.

## 附录

问题一计算欧几里得距离：

```
import pandas as pd
import numpy as np
import networkx as nx
import os
from math import sqrt
import warnings
from collections import defaultdict, deque
warnings.filterwarnings('ignore')
def calculate_distance(coord1, coord2):
    return sqrt((coord1[0] - coord2[0]) ** 2 + (coord1[1] - coord2[1]) ** 2 +
                (coord1[2] - coord2[2]) ** 2)
```

1. 其他 AI 工具生成程序具体见支撑材料。
2. 具体附件 3 详细结果见支撑材料“附件 3 结果”。