

Article

The Development of a New Smart Evacuation Modeling Technique for Underground Mines Using Mathematical Programming

Richard Meij¹, Masoud Soleymani Shishvan^{1,*}  and Javad Sattarvand² ¹ Faculty of Civil Engineering and Geosciences, Delft University of Technology, 628 Delft, The Netherlands² Department of Mining and Metallurgical Engineering, University of Nevada, Reno, NV 89557, USA; jsattarvand@unr.edu

* Correspondence: m.soleymanishishvan@tudelft.nl

Abstract: Navigating miners during an evacuation using smart evacuation technology can significantly decrease the evacuation time of an underground mine in case of emergency hazards. This paper presents a mathematical programming model to calculate the most efficient escape path for miners as a critical component of smart evacuation technology. In this model, the total evacuation distance of the crew is minimized and scenarios with blocked pathways and stamina categories for the miners are simulated. The findings revealed that all the tested scenarios were technically feasible. Using the feature that filters out blocked pathways has no downsides as safer routes are calculated and there is no penalty in the computation time. This paper also discusses the social and technical issues that must be resolved before the algorithm can be implemented as an actual escape solution.

Keywords: smart evacuation; mathematical programming; hazards; underground mines



Citation: Meij, R.; Shishvan, M.S.; Sattarvand, J. The Development of a New Smart Evacuation Modeling Technique for Underground Mines Using Mathematical Programming. *Mining* **2024**, *4*, 106–119. <https://doi.org/10.3390/mining4010008>

Academic Editors: Krzysztof Skrzypkowski, René Gómez, Fhatuwani Sengani, Derek B. Apel, Faham Tahmasebinia and Jianhang Chen

Received: 27 November 2023

Revised: 16 February 2024

Accepted: 19 February 2024

Published: 23 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Current strategies for mine evacuation are blind and outdated technologies that only require people to run to predefined locations such as escape ways or refugee chambers during emergencies [1]. An evacuation is usually initiated by an explosion [2,3] and the release of a stench gas [4]. Upon smelling the stinky gas, miners leave their workplaces and start moving to predefined locations. This is quite a blind methodology because (i) it takes a relatively long time for the air current to reach the people underground [5]; (ii) not all people may be familiar with the layout of the mine, which may lead to confusion when trying to find the route to a safe haven; (iii) some of the predefined escape paths may cross danger zones (such as fires), and miners may approach the danger instead of running away from it; and (iv) people may get confused while navigating when the visibility decreases, which could lead to them making decisions while panicking.

Currently, several methods are used to guide miners to a safe haven during an emergency. These conventional technologies can be divided into passive and active guidance systems [1]. One passive method is to hang signs with directions to the nearest exit at intersections in the mine [4]. These, however, may be hard to read when dirty or when the visibility in the mine is limited (for instance, because of smoke). Another passive method is the use of lifelines [6]. These lines lead the miner from the workstations to the safe havens and use cones to indicate the direction of egress. Active guidance methods give the miners visual and audible cues about the route to safety. For visible cues, LEDs, lasers, or strobe lights may be used. These visual aids have a green color when one is heading towards a safe haven and a red color when one is moving away from it. Audible cues are given to augment the visual cues and use the pitch of sound to guide miners in the right direction.

In a virtual reality environment, it has been proven that smart evacuation is faster than conventional methods of evacuation [7]. According to Gaab (2019), smart evacuation systems are real-time evacuation guidance systems that are adaptable to changing conditions

such as location and spreading of fire and resulting safest and fastest exit routes. To employ smart evacuation, an algorithm is needed to determine the safest and fastest exit routes.

In order to utilize smart evacuation, the real-time localization of individuals underground is needed. There are different methods for localization:

- Radio Frequency Identification (RFID): RFID systems make a connection between what are called tags and readers [8–10]. Tags can be worn by miners or attached to vehicles and contain information about the wearer. Tags communicate with readers using electromagnetic waves. The location of the wearer of the tag can be calculated using the difference in time of arrival, the received signal strength indicator (RSSI), or the time-of-flight method.
- Wi-Fi: Uses the RSSI principle [11]. In this case, access points of the wireless local area network measure the signal strength of smart devices (e.g., smartwatches and smartphones) [9].
- Bluetooth: Uses the RSSI principle in a similar way to Wi-Fi, but can use multiple channels (which is an advantage when there is a lot of background noise) [9]. Bluetooth offers good signal detection, is relatively cheap, and enables the easy introduction of new applications [12]. Also, Bluetooth devices can communicate with each other.
- Wireless Sensor Networks (WSNs): Uses fingerprinting, which is a localization method based on the RSSI principle [13–15]. In this case, a map of sensor patterns (that is, signal strengths of tags at known locations in the mine) is made before the operation is started. These maps compare the actual signals in the mine, giving an idea of where miners or vehicles may be located.
- Image-Assisted Person Location: Identifies miners by the lamp on their helmet (each of which has a unique shape) [16].

There are a number of the shortest path algorithms that could be used for smart evacuation. Three notable examples will be given here:

- Dijkstra's Algorithm: Dijkstra's algorithm is a well-established and popular shortest path algorithm [17–20]. The algorithm assumes a network of nodes, which are connected to one another by arcs. It consists of two methods: one for tree networks (where only one path between two nodes is possible) and one for more complicated networks (such as in underground mines) [21]. The algorithm will always return the shortest path possible but is computationally inefficient (with a running time of $O(n^3)$) [22]. Many shortest path algorithms are derived from Dijkstra's algorithm, each with its own advantages and disadvantages.
- Floyd–Warshall Algorithm: Another method that can be used to compute the shortest path is the Floyd–Warshall algorithm [23]. According to Hougardy (2010), it is widely used and relatively simple. Just like Dijkstra's algorithm, Floyd–Warshall's method works on a network of nodes that are connected by arcs (again, it will be assumed that one is not dealing with tree networks). The worst-case runtime of the algorithm is $O(n^3)$. If used correctly, the method will always return the shortest route between all nodes in a network. However, it does not work in networks where some of the edges have negative values. This, however, is not the case in underground mines.
- Ant Colony Optimization (ACO): ACO can also be used to find a route of egress in case of an underground mine emergency [24]. ACO is inspired by foraging ants [25]. Ants are blind and, therefore, initially, seek food at random. However, they do leave a trail of pheromones while they are on their way. Once they have found food they follow the pheromone trail back to their anthill, leaving new pheromones, making the trail stronger. The crux of the matter is that ants are more likely to follow the stronger pheromone trails. As, over time, the shorter routes to food will have more deposited pheromones on them, more ants will use the more efficient routes. The main advantage of this method is that it is relatively fast; the main disadvantage is that the final solution may be suboptimal [26].

It is clear, then, that each shortest path algorithm has its own advantages and disadvantages. In this paper, a new method based on mathematical programming will be introduced. More specifically, the method will be used to calculate the escape strategy for an underground mine. The development of this method will be presented as a part of a larger project to develop practical smart evacuation technology, supported by the US National Institute of Occupational Health and Safety (NIOSH) at the University of Nevada, Reno. A case study was executed for a drift and fill gold mine located in north-central Nevada, USA. For the case study, a CAD model of the mine containing information on the network of the underground tunnels was used. The locations of the miners, fires, and their destinations in case of an emergency (refugee chambers and shafts) were randomized. This paper investigates if mathematical programming can be used to determine the most efficient escape solution in case of an evacuation.

2. Method

This section outlines a step-by-step procedure for the proposed technique in this study. Figure 1 illustrates this process, which includes gathering necessary data, acquiring the digital mine model, investigating triggered hazards, identifying personnel and hazard location data, constructing the mathematical model, developing and evaluating diverse scenarios, utilizing optimization techniques to solve the built model for each scenario, disseminating results to individuals, guiding individuals to their destinations, and, ultimately, receiving updated hazard locations and furnishing new results if necessary.

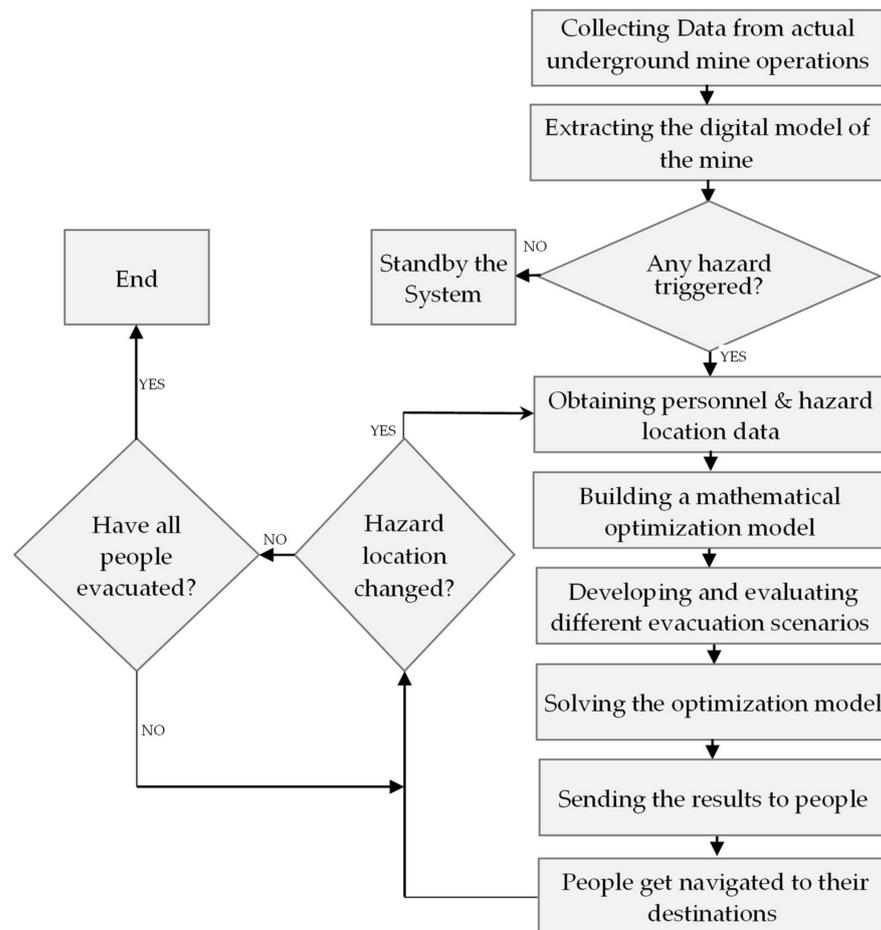


Figure 1. A step-by-step flowchart for the proposed procedure in this study.

2.1. Digital Model of the Mine

The digital modeling process started by constructing a network model that reflected the physical model of the access roads within the mine. The data concerning the dimensions of the mine used for the case study was delivered in an AutoCAD file (in a '.dxf' format). Then, the AutoCAD file was loaded onto the Python environment. Subsequently, the data were converted to an Excel file ('.xlsx' format), serving as the source of data for the main programming file. Converting the data to the Excel format made it easier to comprehend and manipulate the data.

A visual inspection of the file revealed that it contained three categories of data: 'AIR-DATA', '2DARROW', and 'AIRLINES'. The first category contained points in space, while the latter two were line strings, with a beginning and an end node. It was found that 'AIR-DATA' and '2DARROW' were used merely for ventilation purposes, while the 'AIRLINES' dataset contained the drifts making up the mine used for the case study. Therefore, all further use and manipulation of data were performed solely with the 'AIRLINES' category. A plot of the 'AIRLINES' data is given in Figure 2.

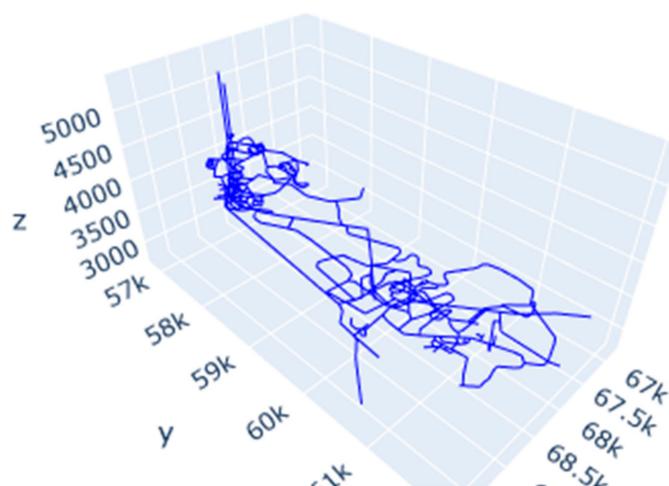


Figure 2. 'AIRLINES' data, with the unit being 'Meter' (k stands for 1000).

The next step was to build a model in which optimization models could be run. It must be noted that although line strings had been drawn, they had not been quantified yet. The only information that was known at this point was the start and end coordinates of each arc. The first step that needed to be taken, then, was to identify each individual node in the network. This was done using a function in Python specifically designed for this purpose. This function first runs through all coordinates of the starting nodes of the line strings. If a combination of coordinates does not occur, it is added to a list. After implementing this process, a sweep is made through the tail nodes to see if any original nodes can be found there. These are also added to the list. In total, 453 nodes were found using this function.

Afterwards, it was essential to find which nodes were combined to form arcs. This procedure was conducted by using another Python function. Firstly, the nodes that were found before were stored in a Pandas data frame. Secondly, the coordinates of each begin and end node of the arcs in the 'AIRLINES' data were compared to their position in the data frame. In this way, a list could be made with the relative positions of the node pairs. For instance, node zero was connected to node six. This appeared as [0, 6] in the list of arcs. In total, there were 522 arcs. Also, the pairs were reversed, as it was assumed that miners could travel in both directions on each arc. In this case, both directions of a pathway could be used during the evacuation optimization process.

Finally, the distance between the connected nodes, the slope of the particular path, and a correction to the distance for this slope needed to be calculated. The calculations for the distances between the nodes were performed using the Euclidian method. The slope

angle of each path was calculated using the tangent. Adjiski et al. (2015) [27] devised a formula that makes a correction to the Euclidian distance to give paths that go more steeply upwards more weight. The correction factor was calculated using the following equation:

$$k_{gi} = \frac{m \times g \times v_0 \times \sin\theta_i}{P_0} + \cos\theta_i \quad (1)$$

where

- k_{gi} is the correction factor
- m is the mass of the miner (taken to be eighty kilograms on average)
- g is the gravitational constant (9.81 m/s²)
- v_0 is the average walking speed (taken to be 1.35 m per second [28])
- θ_i is the slope angle of the path in degrees
- P_0 is the human's walking power (taken to be 200 Watts)

The correction factor was only used for slopes that rose upwards. Slopes going downwards were kept at their original length. The correction for the slope was linked to a maximum angle (taken to be 80 degrees). This was done to prevent the weights for the shafts, which went steeply upwards, becoming erratic. The Euclidian distances corrected by Adjiski's formula were used as the definitive distances between the nodes to be used in the optimization model.

2.2. Mathematical Model

The task of finding the optimal evacuation routes for each specific miner during an emergency in an underground mine can be modeled as a 'Minimum-Cost Network Flow Problem' (MCNFP). This mathematical programming model can be solved with different approaches, such as linear programming and integer programming [29].

The divisibility assumption is an important characteristic of mathematical programming. This assumption states that each variable x_i is allowed to take fractional values. This, of course, is not the case in mine evacuation scenarios (one cannot split miners into fractions). Therefore, a special type of mathematical programming, called integer programming, will be used for all optimization problems in this paper.

The mathematical programming representation of an MCNFP is stated as follows:

$$\min \sum_{\text{all arcs}} c_{ij} * x_{ij} \quad (2)$$

subject to

$$\sum_j x_{ij} - \sum_k x_{ki} = b_i \quad (3)$$

where

- x_{ij} is the number of units of flow sent from node i to node j through arc (i, j)
- c_{ij} is the cost of transporting one unit of flow from node i to node j via arc (i, j)
- b_i is the net supply (outflow minus inflow) at node i

In the objective function (Expression (2)), the total distance that miners needed to travel altogether was minimized as the length of the arc between two nodes, c_{ij} , was multiplied by the number of miners, x_{ij} , that took this route when they were heading for the exit. It should be noted that c_{ij} , as mentioned, was not only based on the distance between nodes [27]. Influences like the slope angle of the path, the temperature, and the quality of the air could be incorporated into this parameter. This allowed for the calculation of not only the shortest route but also one that prioritized safety and efficiency. In this paper, only slope angle, closed pathways, and the stamina of the miners are considered.

Constraint (3) describes the difference between the flows that led towards a node (x_{ij}) and the flows that led away from it (x_{ki}). By setting parameter b_i to a certain value, places where miners are located at the time of an emergency and the nodes where they can find a

safe haven could be simulated. For instance, if a worker was present at node 1, b_1 could be set to one. In this way, the workers were introduced into the network of nodes and arcs. If a refuge chamber at node i could house 10 people, b_i needed to be set to greater or equal to -10 . Also, if a miner reached a safe haven, this individual ‘disappeared’ from the system. A relatively large coefficient was assigned to the mine shaft due to its high capacity. Keeping b_i zero at nodes that served no particular purpose made sure that all miners arriving at this node would have to leave as well. This way, miners had to keep passing ‘empty’ nodes until they found a safe haven.

2.3. Implementing the Built Model

Two pieces of code were written to perform the optimizations: one where all miners were assumed to have the same stamina and one where the miners were divided into stamina categories. In this sub-chapter, a brief overview will be presented, outlining the design process of both codes.

2.3.1. Common Stamina

Although different methods of localizing miners underground were briefly given previously, the exact locations of workers or workstations were not available for this study. Moreover, the locations of safe havens and possible safety hazards were also not known. Therefore, a lot of randomizations were used in the optimization processes. The locations of safe havens were chosen by the author at random nodes in the network and kept equal for all optimizations. The positioning of miners was a bit more complicated as the workers needed to be located at the nodes of the network and always exactly halfway down an arc. This problem was solved by letting the computer pick two things at random: the arc where a miner was located and a number between 0 and 1, which indicated how far a miner was along the specific arc. At this location, a temporary node was created together with two temporary arcs, which connected the temporary node with the two adjacent ordinary nodes. To simulate a safety hazard (for instance, a fire), a random arc was selected and subsequently removed (in both directions) from the network. This arc could not be used in escape routes.

Now that the miners and safety hazards had been introduced to the underground tunnel network, the objective function and constraints (as given in Expression (2) and Constraint (3)) could be set. The linear functions were loaded onto GUROBI, which is a Python application that can be used to solve linear optimization problems. Subsequently, a first optimization attempt could be undertaken. This could lead to two situations: an escape solution being determined or the model being found to be infeasible, which would mean that one or more miners could not reach a safe haven without crossing a safety hazard. If the model came back as infeasible, two things needed to happen: the trapped miners needed to be localized and their colleagues needed to be directed to a safe haven. This could be done by linear relaxation, where GUROBI filters out the constraints that cannot be met and optimizes the model without these. As the violated constraints were linked to the temporary nodes where miners were trapped, it was quite easy to locate these workers.

At this point, the routes for the miners that had a means of escape were calculated and the trapped miners were localized. The route for each individual miner could be plotted, an example of which is given in Figure 3. The red dot indicates the initial location of the miner, while the red lines indicate the route that the miner needed to take to reach safety. In the case of a trapped miner, only a red dot is shown.

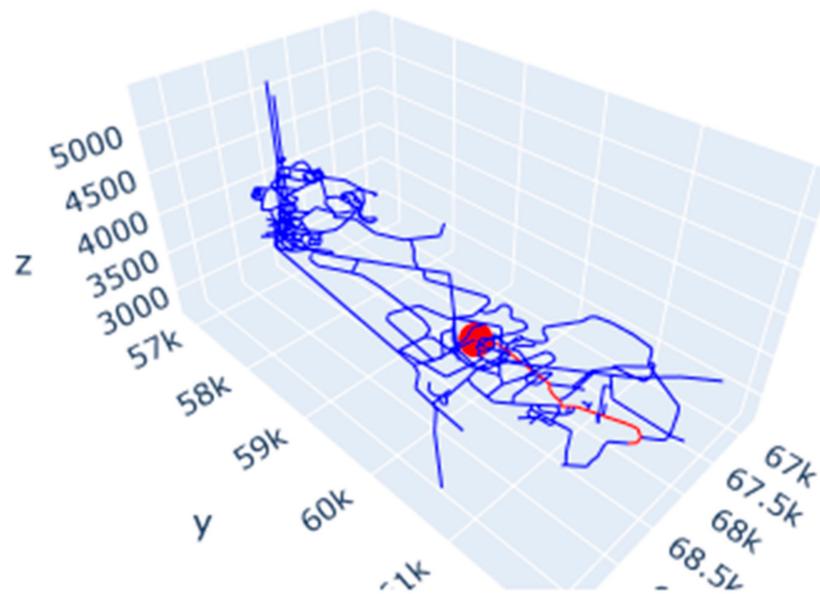


Figure 3. Example of route plot, with the unit being ‘Meter’ (k stands for 1000).

2.3.2. Stamina Categories

In the second piece of code, stamina categories were introduced. Miners could have stamina with a factor of 0.8, 0.9, 1.0, 1.1, or 1.2. A stamina of 0.8 (or 80 percent) meant that the stamina of the miner was low, while a stamina of 1.2 (or 120 percent) meant that the miner was in very good shape. The stamina values were used to correct the distances that the miners needed to travel, which will be explained later in this sub-chapter.

The localization of miners and safety hazards was performed in the same manner as when no stamina categories were used. However, to each individual miner, an extra piece of data was added: an integer between zero and four (which was also chosen by the computer at random). A zero indicated that a miner was in the lowest stamina category, while a four indicated that he or she was in the highest category. Afterwards, five spin-offs needed to be created for each drift in the mine (one for each stamina category). For instance, if a drift was 100 m long, the spin-off drift for miners in the lowest stamina category would be 125 m (which is $\frac{100}{0.8}$). For miners in the highest stamina category, this would be 83.3 m (which is $\frac{100}{1.2}$). In this way, the routes for miners with low stamina were made artificially longer, while, for miners with high stamina, they were made shorter. The purpose of this was to assign relatively lower weight to miners with higher stamina levels in the objective function (as given in Constraint (2)). This approach ensured that such miners were directed towards a distant safe haven if the closest option was almost their capacity. For instance, if two miners (one in a high and one in a low stamina category) were at location A, while the safe haven at location B could only harbor one more individual, the miner in the lower stamina category would be favored for this spot. Indeed, it was less expensive to send a miner in the higher category to a safe haven farther away.

At this point, the objective function and the constraints could be fed to the GUROBI application. This, however, was a bit more complicated than previously. This was because miners needed to take only the spin-off path that belonged to their stamina category. This meant a large increase in the constraints of the optimization problem (which were multiplied by a factor of five). This situation led to longer computation times (which will be further elaborated on in the Results section). The final result, however, was the same as in the version of the code without stamina categories: trapped miners were located, and all others received directions for a safe haven.

2.4. Scenarios and Situations

A total of four scenarios were tested:

- Scenario 1: No correction to the pathlengths for the stamina of the miners and no blocked pathways.
- Scenario 2: No correction to the pathlengths for the stamina of the miners and 10 blocked pathways.
- Scenario 3: Pathlengths corrected for the stamina of the miners and no blocked pathways.
- Scenario 4: Pathlengths corrected for the stamina of the miners and 10 blocked pathways.

Each of the scenarios was run for five different situations. In each situation, the miners and fires were located differently. The situations remained consistent across all scenarios, and optimizations were run with 1, 5, 10, 50, 100, 200, 500, and 1000 underground workers. The locations of the safe havens were kept the same in all optimizations.

3. Results

In this section, three different types of results relevant to utilizing a smart evacuation algorithm will be analyzed. Firstly, the division of miners among the safe havens in the different scenarios for a specific situation will be looked at. This is done for two reasons: firstly, to see how various scenarios influence the escape solution and, secondly, to see what the impact of an increasing number of miners is on this solution. After completing this step, the paths of an individual miner (called Miner X) in the different scenarios are analyzed to characterize how blocked paths and the use of stamina categories influence the route to safety for a specific miner. Finally, the computation times of the different situations will be compared to see how added features influence the efficiency of the algorithms.

3.1. Distribution of Miners among Safe Havens

Five safe havens were used in each of the optimizations: two shafts (located at nodes 200 and 250) and three refugee chambers (located at nodes 120, 150, and 300). The refugee chambers were assumed to have a capacity of 30 miners, while the shafts had infinite capacity. Tables 1–4 provide the distribution of miners among the safe havens for different scenarios in Situation 3.

Table 1. Defined destinations for Scenario 1 in Situation 3.

Refuge/# of Miners	1	5	10	50	100	200	500	1000
Chamber 120	0	2	3	10	22	30	30	30
Chamber 150	0	0	0	4	11	30	30	30
Chamber 300	0	0	0	2	5	16	30	30
Shaft 200	0	0	4	22	42	85	223	482
Shaft 250	1	3	3	12	20	39	187	428
Total	1	5	10	50	100	200	500	1000

Table 2. Defined destinations for Scenario 2 in Situation 3.

Refuge/# of Miners	1	5	10	50	100	200	500	1000
Chamber 120	0	2	3	9	18	30	30	30
Chamber 150	0	0	0	4	11	21	30	30
Chamber 300	0	0	0	3	8	24	30	30
Shaft 200	0	0	4	22	44	87	242	503
Shaft 250	1	3	3	11	18	35	163	402
Total	1	5	10	49	99	197	495	995

Table 3. Defined destinations for Scenario 3 in Situation 3.

Refuge/# of Miners	1	5	10	50	100	200	500	1000
Chamber 120	0	2	3	10	22	30	30	30
Chamber 150	0	0	0	4	11	30	30	30
Chamber 300	0	0	0	2	5	16	30	30
Shaft 200	0	0	4	22	42	85	224	475
Shaft 250	1	3	3	12	20	39	186	435
Total	1	5	10	50	100	200	500	1000

Table 4. Defined destinations for Scenario 4 in Situation 3.

Refuge/# of Miners	1	5	10	50	100	200	500	1000
Chamber 120	0	2	3	9	18	30	30	30
Chamber 150	0	0	0	4	11	21	30	30
Chamber 300	0	0	0	3	8	24	30	30
Shaft 200	0	0	4	22	44	87	243	502
Shaft 250	1	3	3	11	18	35	162	403
Total	1	5	10	49	99	197	495	995

The first point that one should notice is that for up to 200 underground workers, the capacities of the refugee chambers were sufficient. This meant that each miner could be sent to the safe haven closest to them. From 500 miners and up, the refugee chambers reached capacity, which led to more miners being sent to a shaft. In Scenarios 2 and 4, not all miners could make it to a safe haven. This was, as mentioned, due to the blocked paths in these scenarios, which led to miners getting trapped.

In order to compare the scenarios, Tables 5–8 present the absolute differences in the number of miners located in a specific safe haven across all scenarios.

Table 5. The difference in the number of miners located in a safe haven between Scenarios 1 and 2 in Situation 3.

Refuge/# of Miners	1	5	10	50	100	200	500	1000
Chamber 120	0	0	0	1	4	0	0	0
Chamber 150	0	0	0	0	0	9	0	0
Chamber 300	0	0	0	1	3	8	0	0
Shaft 200	0	0	0	0	2	2	19	21
Shaft 250	0	0	0	1	2	4	24	26

Table 6. The difference in the number of miners located in a safe haven between Scenarios 3 and 4 in Situation 3.

Refuge/# of Miners	1	5	10	50	100	200	500	1000
Chamber 120	0	0	0	1	4	0	0	0
Chamber 150	0	0	0	0	0	9	0	0
Chamber 300	0	0	0	1	3	8	0	0
Shaft 200	0	0	0	0	2	2	19	27
Shaft 250	0	0	0	1	2	4	24	32

Tables 5 and 6 compare the scenarios with and without blocked paths. According to these tables, the blocked paths caused a significant shift in how miners were divided among the safe havens. This circumstance occurred due to two reasons. The first, and most important, was that miners may have needed to take detours due to blocked pathways. On their new route, the safe haven that was previously closest may then have been relatively farther away. Moreover, the route to a safe haven may have been blocked entirely for a specific miner. This could lead to miners heading to a different destination than the one

they would have gone to when all paths were available. The second, and less important, reason for the differences was the trapped miners. These miners did not make it to a safe haven at all, which can be found back in the numbers.

Table 7. The difference in the number of miners located in a safe haven between Scenarios 1 and 3 in Situation 3.

Refuge/# of Miners	1	5	10	50	100	200	500	1000
Chamber 120	0	0	0	0	0	0	0	0
Chamber 150	0	0	0	0	0	0	0	0
Chamber 300	0	0	0	0	0	0	0	0
Shaft 200	0	0	0	0	0	0	1	7
Shaft 250	0	0	0	0	0	0	1	7

Table 8. The difference in the number of miners located in a safe haven between Scenarios 2 and 4 in Situation 3.

Refuge/# of Miners	1	5	10	50	100	200	500	1000
Chamber 120	0	0	0	0	0	0	0	0
Chamber 150	0	0	0	0	0	0	0	0
Chamber 300	0	0	0	0	0	0	0	0
Shaft 200	0	0	0	0	0	0	1	1
Shaft 250	0	0	0	0	0	0	1	1

Tables 7 and 8 give the differences with and without the use of stamina categories in the algorithms. In these cases, no differences could be seen for the optimizations with up to 200 miners. Minimal differences could be seen for the shafts for optimizations with 500 miners or more. This was attributed to the fact that the refugee chambers could accommodate enough miners, up to a capacity of 200. This meant that miners were sent to the safe haven closest to them, regardless of their stamina category. In the optimizations where the refugee chambers were full, miners that had lower stamina were favored for a spot in the refugee chambers (if this was relatively closer by). This meant that a miner with better stamina would be sent to a shaft in his/her place. However, it is essential to note that this shaft may not have been the same one the weaker miner would have gone to. This led to some differences in the division of miners around the shafts.

3.2. Miner X

The purpose of this section is to investigate how blocked pathways and stamina categories influenced the path of an individual miner. Miner X was part of the optimizations executed in the second situation. This individual was in the highest stamina category, and his/her path was determined in the case where 500 miners were present. The different paths are presented in Table 9.

Table 9. Destinations of Miner X.

Scenario	Distance in Meters	Final Destination Node
1	2502.7	150 (Refugee Chamber)
2	2502.7	150 (Refugee Chamber)
3	4285.1	250 (Shaft)
4	6821.1	250 (Shaft)

According to Table 9, there was no difference between Scenarios 1 and 2. This means that, despite the fires, Miner X encountered no blocked paths on his/her route to safety in Scenario 2. When evaluating Scenario 3, it becomes evident that the distance Miner X needed travel increased significantly compared to Scenarios 1 and 2. This was because a

weaker miner was favored for a spot in refugee chamber 150, and Miner X had to head to a shaft in his/her place. Finally, in Scenario 4, the path for Miner X was longer than in Scenario 3. This was because one or more pathways on his/her original route were blocked because of fire, which meant that he/she needed to take a detour to get to the shaft at node 250.

It can be concluded from Table 9 that both blocked pathways and stamina categories added significantly to the path of an individual miner. In the case of blocked pathways, this was not necessarily undesirable. Although the route for the individual miner may have been longer, it did avoid hazardous situations such as fires. Therefore, pathlength was exchanged for safety. That said, the added pathlength due to the stamina categories does raise social and ethical questions. Can you expect a miner to travel farther in favor of a weaker colleague?

It should be noted that the pathlengths for Miner X seem unrealistically long. The reason for these unrealistic pathlengths is the randomization that was used in the optimizations for the locations of miners and safe havens. In this randomization, safety standards were disregarded. Naturally, when implementing this system in an actual underground mine, the locations of workstations and safe havens are known, and all pathlengths in case of an evacuation would be within the limits stated in safety regulations.

3.3. Computation Times

The average times to run the algorithms for different scenarios are indicated in Figure 4.

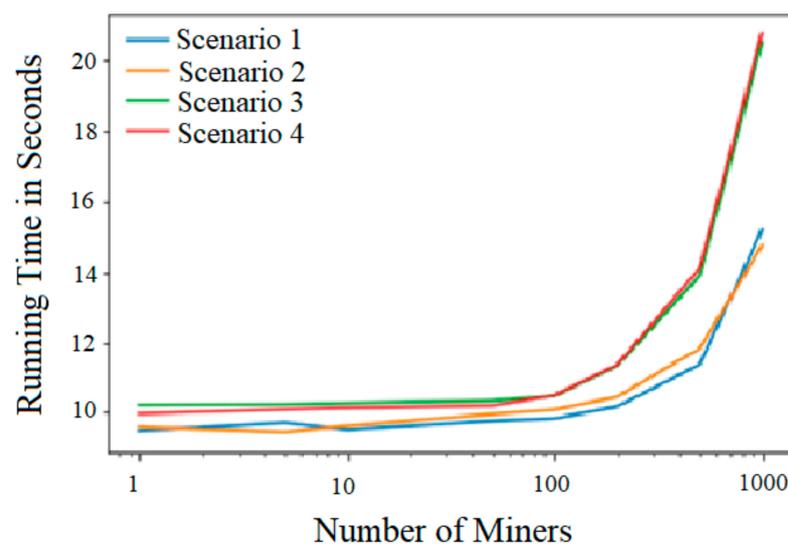


Figure 4. Average computation times for different scenarios.

As shown in Figure 4, blocked pathways had a negligible effect on the computation time of the algorithms. This is remarkable as these scenarios required more execution of code. This means that there is virtually no downside to including the feature that filters out blocked pathways. Trapped miners are localized, and their colleagues are sent on routes that avoid hazardous situations, without overly complicating the algorithm. It is therefore highly recommended to use this feature in escape solutions.

Using stamina categories, on the other hand, did add to the computation time (up to 5 s in optimizations with 1000 miners underground). Although this time difference would most likely not result in a matter of life and death, it does, again, raise some questions about desirability. Firstly, if the refugee capacity is sufficient, the final solution that the algorithm generates will be the same. In this case, it is undesirable to have the added computation time. Secondly, if the capacity of the refugee chambers is insufficient, added computation time is just one of the objections one can have against the use of stamina categories.

A final note is that the optimizations were run on an ordinary, everyday-use laptop. If this system were to be implemented in an actual mining operation, most likely, a specialized computer would be used. This could lead to a significant decrease in the time it takes for the algorithms to run.

4. Discussion

This paper raises some ethical and social questions about the obtained solutions. Firstly, there is the issue of using stamina categories. It has been proven that using this type of category is technically feasible, but one could question whether it is desirable to employ sexual and/or fitness discrimination in decision-making. Furthermore, they can have an adverse effect on the path of an individual miner (who has good stamina). Is it ethical or socially acceptable to expect a miner to accept a longer path of escape in favor of a weaker colleague? However, it is worth noting that these categories may be useful in cases where the operation employs vehicles, such as underground trucks. These categories, then, might not be used for dividing miners but can be used for a different purpose.

Using the algorithm without stamina categories also raises ethical issues. For instance, if five miners are positioned at the same workstation, the closest refugee chamber can only harbor four more workers. In this case, a decision would have to be made about which miner misses out on a spot in this safe haven. This can present a tough ethical dilemma. Who is entrusted with making the decision about which miner cannot access the closest safe haven—the algorithm, the miners themselves, or the emergency controller? There is, probably, not a single definitive answer to this question. Nevertheless, this issue must be addressed and resolved before implementing a smart evacuation system.

Another issue is how to classify which paths are suitable to use for escape and which are not. Naturally, in the case of a fire or toxic gases, the exclusion of a path is obvious. However, what if a path is partly blocked by machinery or other obstacles? What if a drift is partly filled with water? This issue can partly be resolved by giving these paths extra weight in the objective function, but it requires that the conditions for every underground location are extensively monitored. A system would need to be devised that assigns certain penalties to certain situations. It can be argued, then, that the current system of assigning weights to the pathways is not yet complete.

5. Conclusions

In this paper, a mathematical programming component for a new smart evacuation algorithm for underground mines was proposed. The algorithm set the evacuation model as a 'Minimum-Cost Network Flow Problem' that can be solved using any mathematical programming solver. This was conducted by setting the evacuation as an objective function that was subject to several constraints. The objective function minimized the total distance traveled by all underground miners. The constraints were used to indicate the locations of workers and safe havens. The algorithms for this paper were written in the Python programming language and solved by use of the GUROBI library.

A total of four scenarios were tested: with and without blocked pathways and with and without dividing the miners up into stamina categories. Each of the scenarios was run for five different situations, with optimizations for 1, 5, 10, 50, 100, 200, 500, and 1000 miners. The locations of miners and blocked pathways were chosen randomly by the computer in different situations. The safe havens were chosen randomly by the author and kept the same for all optimizations.

It was found that in optimizations with up to 200 miners, the capacity of the refugee chambers was sufficient. For optimizations with 500 miners or more, the refugee chambers reached the designated capacity, which led to more miners heading to the shafts. The introduction of blocked pathways had a significant influence on how the miners were divided up among the safe havens. There were two reasons for this. Firstly, blocked paths may cause a safe haven that was once closest to become relatively farther away or entirely unreachable. This can lead to miners being sent to a different safe haven. The second reason

is that blocked pathways may cause miners to get trapped. Therefore, they will not reach a safe haven, which can be found back in the numbers. Stamina categories only had some effect on the division of miners around the shafts in optimizations with 500 miners and more. This is because when a stronger miner was sent to a shaft in favor of a weaker miner, this did not need to be the same shaft as that the latter would have originally gone to.

To give an idea of how the different scenarios influence the path of an individual worker, Miner X was introduced (who came from an optimization with 500 miners and was in the highest stamina category). It was found that both blocked pathways and stamina categories may add significantly to the path of an individual miner. This is not that big of an issue in the case of blocked paths as pathlength is exchanged for a safer route. However, in the case of stamina categories, there are some social and ethical objections that can be raised.

Introducing a feature that filters out blocked pathways and localizes trapped miners does not give a penalty to the computation time of the algorithms. This means that there is virtually no downside to using this feature. Stamina categories add to the computation time. If there is sufficient refugee capacity, this is undesirable, as there will be no impact on the final solution. If the capacity is insufficient, the time penalty will most likely not result in a matter of life and death. However, social and ethical objections play a role here as well.

Author Contributions: All authors contributed to the study conception and design. R.M. participated in all phases of the study, including modeling, analysis, and interpreting the results. M.S.S. participated in data collection and analysis of the results. J.S. developed the research methodology and validated the models and outcomes. All authors have read and agreed to the published version of the manuscript.

Funding: The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data Availability Statement: The data presented in this study are available on request from the corresponding author (accurately indicate status).

Acknowledgments: This research is based on the MSc thesis of the first author and is part of a capacity-building project supported by the US National Occupational Safety and Health (NIOSH) at the University of Nevada, Reno. The authors acknowledge NIOSH for making this research happen. We would also like to thank Marco Keersemaaker for his feedback on the thesis.

Conflicts of Interest: The authors have no relevant financial or non-financial interests to disclose.

References

1. Brenkley, D.; Bennett, S.; Jones, B. Enhancing mine emergency response. In Proceedings of the 28th International Conference on Safety in Mines Research Institutes, Sinaia, Romania, 7–11 June 1999.
2. Hao, J.; Du, Z.; Zhang, T.; Li, H. Influence of $\text{NH}_4\text{H}_2\text{PO}_4$ powder on the laminar burning velocity of premixed CH_4 /Air flames. *Int. J. Hydrogen Energy* **2022**, *47*, 38477–38493. [[CrossRef](#)]
3. Jiang, H.; Bi, M.; Li, B.; Ma, D.; Gao, W. Flame inhibition of aluminum dust explosion by NaHCO_3 and $\text{NH}_4\text{H}_2\text{PO}_4$. *Combust. Flame* **2019**, *200*, 97–114. [[CrossRef](#)]
4. Conti, R.S.; Chasko, L.L.; Wiehagen, W.J.; Lazzara, C.P. *Fire Response Preparedness for Underground Mines*; US Department of Health and Human Services: Washington, DC, USA, 2005.
5. Conti, R.S.; Yewen, R.G. *Evaluation of a Signaling and Warning System for Underground Mines*; US Department of Health and Human Services: Washington, DC, USA, 1997.
6. Conti, R.S. Emerging technologies: Aiding responders in mine emergencies and during the escape from smoke-filled passageways. In Proceedings of the Northwest Mining Association's 107th Annual Meeting, Spokane, WA, USA, 3–7 December 2001.
7. Gaab, S.J. Evaluation of Smart Underground Mine Evacuation Efficiency through Virtual Reality. Ph.D. Thesis, University of Nevada, Reno, NV, USA, 2019.
8. Radinovic, G.; Kim, K. Feasibility study of RFID/Wi-Fi/Bluetooth wireless tracking system for underground mine mapping—Oklahoma. In Proceedings of the 2008 Incorporating Geospatial Technologies into SMCRA Business Processes, Atlanta, GA, USA, 25–27 March 2008.
9. Rusu, S.R.; Hayes, M.J.D.; Marshall, J.A. Localization in large-scale underground environments with RFID. In Proceedings of the 2011 24th Canadian Conference on Electrical and Computer Engineering (CCECE), Niagara Falls, Canada, 8–11 May 2011.

10. Yan, N.; Chen, H.; Lin, K.; Ni, Z.; Li, Z.; Xue, H. BFSearch: Bloom filter based tag searching for large-scale RFID systems. *Ad Hoc Netw.* **2023**, *139*, 103022. [CrossRef]
11. Mohapatra, A.G.; Keswani, B.; Nanda, S.; Ray, A.; Khanna, A.; Gupta, D.; Keswani, P. Precision local positioning mechanism in underground mining using IoT-enabled WiFi platform. *Int. J. Comput. Appl.* **2020**, *42*, 266–277. [CrossRef]
12. Baek, J.; Choi, Y.; Lee, C.; Suh, J.; Lee, S. BBUNS: Bluetooth beacon-based underground navigation system to support mine haulage operations. *Minerals* **2017**, *7*, 228. [CrossRef]
13. Chehri, A.; Fortier, P.; Tardif, P.M. UWB-based sensor networks for localization in mining environments. *Ad Hoc Netw.* **2009**, *7*, 987–1000. [CrossRef]
14. Yiu, S.; Dashti, M.; Claussen, H.; Perez-Cruz, F. Wireless RSSI fingerprinting localization. *Signal Process.* **2017**, *131*, 235–244. [CrossRef]
15. Aroba, O.J.; Nalindren, N.; Timothy, A.; Avintha, G.; Khadija, K. A review: The bibliometric analysis of emerging node localization in wireless sensor network. *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.* **2023**, *15*, 141–153.
16. Niu, Q.; Yang, X.; Yin, Y. IPL: Image-Assisted Person Localization for Underground Coal Mines. *Sensors* **2018**, *18*, 3679. [CrossRef] [PubMed]
17. Dramski, M. A comparison between Dijkstra algorithm and simplified ant colony optimization in navigation. *Zesz. Nauk. Akad. Morska Szczecinie* **2012**, *29*, 25–29.
18. Fan, D.; Shi, P. Improvement of Dijkstra’s algorithm and its application in route planning. In Proceedings of the 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery, Yantai, China, 10–12 August 2010.
19. Hong, Y.; Li, D.; Wu, Q.; Xu, H. Dynamic route network planning problem for emergency evacuation in restricted-space scenarios. *J. Adv. Transp.* **2018**, *2018*, 4295419. [CrossRef]
20. Hughes, T.M. Graph processing by which to evacuate a mine. In Proceedings of the 1990 Symposium on Applied Computing, Fayetteville, AR, USA, 5–6 April 1990.
21. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
22. Bari, A. 4.2 All Pairs Shortest Path (Floyd-Warshall)—Dynamic Programming. Youtube. 2018. Available online: <https://www.youtube.com/watch?v=oNI0rf2P9gE> (accessed on 20 February 2022).
23. Hougardy, S. The Floyd–Warshall algorithm on graphs with negative cycles. *Inf. Process. Lett.* **2010**, *110*, 279–281. [CrossRef]
24. Guangwei, Y.; Dandan, F. Escape-route planning of underground coal mine based on Improved Ant Algorithm. *Swarm Intell. Eng.* **2013**, *2013*, 687969.
25. Mirjalili, A. Inspiration of Ant Colony Optimization. 2018. Available online: <https://www.youtube.com/watch?v=1qpvpOHGRqA> (accessed on 10 September 2023).
26. Abdmouleh, Z.; Gastli, A.; Ben-Brahim, L.; Haouari, M.; Al-Emadi, N.A. Review of optimization techniques applied for the integration of distributed generation from renewable energy sources. *Renew. Energy* **2017**, *113*, 266–280. [CrossRef]
27. Adjiski, V.; Mirakovski, D.; Despodov, Z.; Mijalkovski, S. Simulation and optimization of evacuation routes in case of fire in underground mines. *J. Sustain. Min.* **2015**, *14*, 133–143. [CrossRef]
28. Cronkleton, E. What Is the Average Walking Speed of an Adult? Available online: <https://www.healthline.com/health/exercise-fitness/average-walking-speed> (accessed on 14 January 2023).
29. Winston, W.L.; Goldberg, J.B. *Operations Research: Applications and Algorithms*; Thomson Brooks/Cole Belmont: Belmont, CA, USA, 2004; Volume 3.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.