Development of a new smart evacuation modelling technique for underground mines using mathematical programming

R.S.H. Meij[1], J. Sattarvand[2], and M.S. Shishvan[1]

[1]Delft University of Technology, The Netherlands
[2]University of Nevada, USA

Navigating miners during an evacuation using a smart evacuation technology can significantly decrease the evacuation time of an underground mine in case of an emergency. This paper presents a mathematical programming model to calculate the most efficient escape path for each miner as a critical component of the smart evacuation technology. In this model, the total evacuation distance of the crew is minimised and scenarios with blocked pathways, and stamina categories for the miners are simulated. It was found that all the tested scenarios are 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. The paper also discusses the social and technical issues that need to be overcome before the algorithm can be implemented as an actual escape solution.

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Current strategies for mine evacuation are blind, and outdated technologies that only require people to run to predefined locations such as escape ways or refugee chambers during emergencies (Brenkley, Bennett, & Jones, 1999). These technologies can be divided up into passive methods (such as hanging signs on intersections of mining pathways (Chasko, Conti, Lazzara, & Wiehagen, 2005) or by using lifelines (Conti, 2001), and active methods (which give both visual and audible cues about the exit route). In many cases, the predefined routes may pass through danger zones and are definitely not the best choice for dispatching people.

In a virtual reality environment, it has been proven that smart evacuation is faster than conventional methods (Gaab, 2019). 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". In order to employ the smart evacuation, an algorithm is needed to determine the safest and fastest exit routes.

In order to utilise smart evacuation, real time localisation of the individuals underground is needed. Localisation can be done using radio frequency identification (where miners wear tags which communicate by readers using electromagnetic waves (Radinovic & Kim, 2008; Rusu, Hayes, & Marshall, 2011)), WiFi (which used the signal strength of smart devices (Mohapatra *et al.*, 2020)), Bluetooth (which is similar to WiFi, but works better in an environment with a lot of background noise) (Baek, Choi, Lee, Suh, & Lee, 2017)), Wireless Sensor Networks (which uses fingerprinting of radio signals (Chehri, Fortier, & Tardif, 2009; Yiu, Dashti, Claussen, & Perez-Cruz, 2017)), and Image-Assisted Person Location (which identifies miners by the lamp on their helmet (Niu, Yang, & Yin, 2018)).

There are several algorithms that can be used to determine the shortest path. Examples are Dijkstra's algorithm (Dijkstra, 1959; Hong, Li, Wu, & Xu, 2018), the Floyd-Warshall algorithm (Bari, 2018; Hougardy, 2010) and Ant Colony Optimisation (ACO) (Guangwei & Dandan, 2013; Mirjalili, 2018). All these methods have their own advantages and disadvantages. Dijkstra's algorithm and the Floyd-Warshall algorithm will always find the shortest route possible, but can become computationally inefficient if the network it is used upon is extensive (with a worst case running time of $O(n^2)$ for Dijkstra's algorithm (Biswas, Mishra, & Mahanti, 2005)). ACO is faster, but may deliver a suboptimal solution.

In this paper, the development of a smart evacuation algorithm, based on mathematical programming, will be presented as part of a larger project to develop a practical smart evacuation technology, supported by the US National Institute of Occupational Health and Safety at the University of Nevada, Reno. A case study is executed for a drift and fill gold mine located in north-central Nevada, USA. For the case study, the CAD model of the mine containing information of the network of the underground tunnels is used. The locations of the miners, fires, and their destinations in case of an emergency (refugee chambers and shafts) are randomised. This paper investigates if mathematical programming can be used to determine the most efficient escape solution in case of an evacuation.

METHOD

Mathematical Model

The task of finding the optimal evacuation routes for each specific miner during an emergency in an underground mine can be modelled as a Minimum-Cost Network Flow Problem (MCNFP), which is a well-known optimisation method. The mathematical programming model can be solved with different approaches (Winston & Goldberg, 2004). According to Winston and Goldberg (2004), a mathematical programming problem can be defined as follows:

- "It is attempted to maximise (or minimise) a linear function of decision variables. The function that is to be maximised or minimised is called the objective function."
- "The values of the decision variables must satisfy a set of constraints. Each constraint must be a linear equation or linear equality."
- "A sign restriction is associated with each variable. For any variable x_i, the sign restriction specifies that x_i must be either nonnegative ($x_i \geq 0$) or unrestricted in sign (urs)."

Modelling starts with construction of a network model which is identical to the physical model of the access roads in the mine. The mathematical programming representation of an MCNFP is stated as:

$$min \sum_{all\ arcs} c_{ij} * x_{ij} \tag{1}$$

Subject to

$$\sum_{j} x_{ij} - \sum_{k} x_{ki} = b_i \tag{2}$$

Where,

- x_{ij} is the number of units of flow sent from node i to node j trough 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 (equation 1) the total distance that miners need to travel altogether is minimised as the length of the arc between two nodes, c_{ij}, is multiplied by the number of miners, x_{ij}, that take this route when they are heading for the exit. It should be noted that c_{ij} is not only based on the distance between nodes (Adjiski, Mirakovski, Despodov, & Mijalkovski, 2015). Influences like the slope angle of the path, the temperature and quality of the air can be incorporated in this parameter, so as to not only calculate the shortest route, but also one that is the safest and the most efficient. In this paper, only slope angle, closed pathways, and the stamina of the miners are considered. How the stamina for individual miners is introduced will be discussed later in this paper.

Constraints (as given in equation 2) describe the difference between the flows that lead towards a node (x_{ij}), and the flows that lead 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 can be simulated. For instance, if a worker is present at node 1, b_1 can be set to one. This way, the workers are introduced in the network of nodes and arcs. If a refugee chamber at node i can house ten people, b_i should be set to more or equal than -10. This way, when a miner reaches a safe haven, he 'disappears' from the system. A relatively large coefficient is assigned to the mine shaft due to its high capacity. Keeping b_i zero at nodes that serve no particular purpose, makes sure that all miners arriving at this node will have to leave as well. This way, miners have to keep passing 'empty' nodes until they find a safe haven.

Equation 1 and 2 are not the full mathematical description of the MNCFP. Normally, an extra equation considering the capacities of the specific arcs is included. For instance, if a certain drift in the mine can only harbour twenty miners, a constraint setting this limit could be included. It is assumed, however, that miners will arrive at specific drifts at different points in time, eliminating the need for specific capacities for the arcs. The main reason for this assumption is that the model is static rather than dynamic: the escape solution is calculated at a certain point in time, and is not modified afterwards. Setting a constraint for a specific arc could lead to miners finding detours or even being trapped, while this is not necessary. If the model would be dynamic (i.e. is constantly or regularly updated during an emergency), one could consider extending the model with this feature.

Implementation
The Python programming language is used to implement the escape algorithm proposed in this paper. Special note needs to be made of the GUROBI library that solves the optimisation problems, using the network simplex algorithm.

Scenarios and Situations
A total of four scenarios are 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 with ten blocked pathways.
- Scenario 3: Pathlengths are corrected for the stamina of the miners and no blocked pathways.
- Scenario 4: Pathlengths are corrected for the stamina of the miners with ten blocked pathways.

As can be seen, stamina categories are introduced in the last two scenarios. In these scenarios, miners are given a value from zero to four (where zero indicates a low stamina, two means an ordinary stamina, and four is the highest category). Miners in the lowest category are given a stamina of 80% while the miners in the highest category are given a stamina of 120%. By adjusting the objective function and constraints so that each category is represented, miners in a low category will be favoured for a spot in a refugee chamber (if this is relatively close by, and the capacity is limited). A miner in a higher category will be sent to a safe haven further away in their place. This is done by adjusting the pathlengths relative to the percentages for each category. For instance, for an arc of 100 metres, the pathlength for a miner in the lowest category will become $\frac{100}{0.8}$, which is 125 metres. For miners in the highest category this will be $\frac{100}{1.2}$, which is 83.3 metres. Because the paths for miners in a high category are relatively shorter, it is cheaper to send them to safe havens further away.

Each of the scenarios were run for five different situations. In each situation, the miners and blocked pathways are located differently and chosen randomly by the computer. The situations (i.e., the location of miners and fires) are the same in each scenario and are run with 1, 5, 10, 50, 100, 200, 500, and 1000 workers underground. The locations of the safe havens are kept the same in all simulations. The algorithm is not sensitive to location of fires. The random locations were only chosen for the sake of this research. In the future, when the algorithm is applied for the real-world examples, there will be no concerns and no bias will be introduced in the generated results. The number of ten blocked pathways may seem a bit high. The reason this number has been chosen was to make sure that in some simulations miners get trapped, so as to demonstrate the algorithm will still work in these cases.

Visualisation
The escape solution for each individual miner can be visualised. An example of this is given in Figure 1. The red dot indicates the initial location of the miner. The red lines give the route she/he needs to follow to reach a safe haven.

In scenario 2 and 4 the random locations of fires or roof collapses are implemented. In these scenarios, because some of the paths are blocked, it is possible that some miners may get trapped without any feasible escape way. In this case, the mathematical model identifies their locations and provides a plot of their positions for the rescue teams. These types of plots consist only of red dots to indicate the locations of the trapped miners.

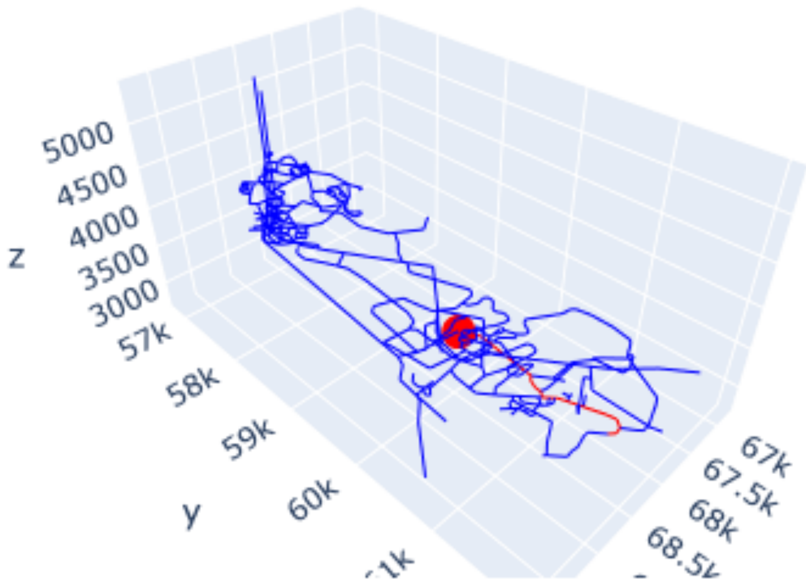

Figure 1. Example of route plot. The red dot indicates the original position of a miner. The red lines are his/her route to safety.

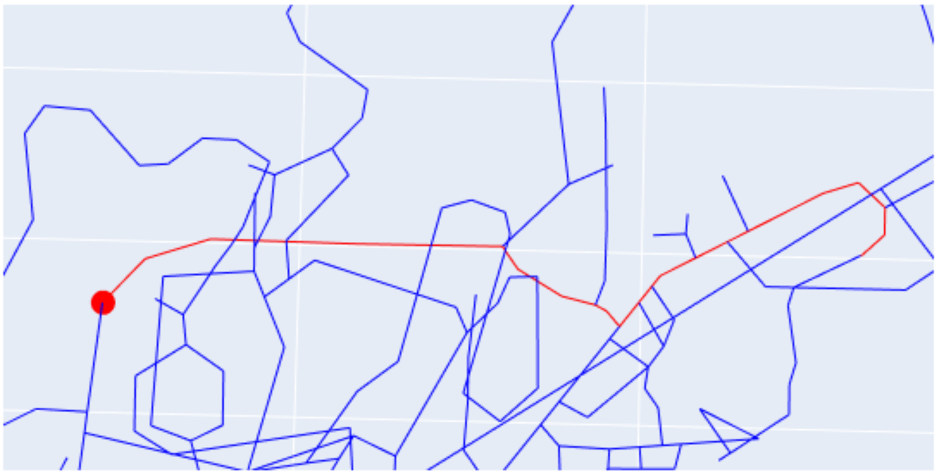

Figure 2. Close up of escape path.

RESULTS

Examples of the distribution of miners among safe havens, the paths of an individual miner (called *Miner X)* in the different scenarios, and the computation times of the algorithms are presented as results.

Distribution of miners among safe havens

Five safe havens are used in each of the simulations: two shafts (located at node 200 and node 250) and three refugee chambers (located at nodes 120, 150, and 300). As mentioned, these nodes were chosen randomly by the author. The numbers of the specific nodes were used for programming purposes, and given in this paper only for means of identification. The refugee chambers are assumed to have a capacity of 30 miners, while the shafts have infinite capacity. Table 1 to Table 4 show the distribution of the miners among the safe havens for each different scenario in situation 3. For brevity, only situation 3 is given as an example. For full results please see (Meij, 2020).

A first point that one should notice is that with up to 200 miners underground, the capacities of the refugee chambers are sufficient. This does not mean that all miners have to be directed to a refugee chamber, if they are close to the exit shaft, the algorithm will direct them to this point. Furthermore, in some conditions some miners will be sent to a safe haven that is not the closest (i.e., if refugee chamber A, which is the closest, is full, a miner may be guided to the other refugee chamber). This is one of the strongest points of this approach; everything can be planned in real-time, and if someone has to go to another chamber, it is known from the beginning. As there were a lot of randomisations in the localisation of miners and safe havens, no hard conclusions can be drawn from these results. However, it is important to see that if the refugee chambers were to fill up, more miners would be sent to the shafts. From 500 miners and up, the refugee chambers will reach their maximum capacity, which leads to more miners being sent to the exit shaft. Besides this, in scenario 2 and 4 not every miner will make it to a safe haven. This is, as mentioned, due to the blocked paths in these scenarios, which lead to miners being trapped.

In order to compare the scenarios, in Table to

Table , the absolute difference in the number of miners in a specific safe haven for the different scenarios can be found.

Table 1. Defined destinations for Scenario 1, Situation 3

Refugee / # 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
Trapped Miners	0	0	0	0	0	0	0	0

Table 2. Defined destinations for Scenario 2, Situation 3

Refugee / # 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
Trapped Miners	0	0	0	1	1	3	5	5

Table 3. Defined destinations for Scenario 3, Situation 3

Refugee / # 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
Trapped Miners	0	0	0	0	0	0	0	0

Table 4. Defined destinations for Scenario 4, Situation 3

Refugee / # 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
Trapped Miners	0	0	0	1	1	3	5	5

Table 5. Difference in number of miners in safe haven between Scenario 1 and 2, Situation 3

Refugee / # 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. Difference in number of miners in safe haven between Scenario 3 and 4, Situation 3

Refugee / # 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

Table 7. Difference in number of miners in safe haven between Scenario 1 and 3, Situation 3

Refugee / # 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. Difference in number of miners in safe haven between Scenario 2 and 4, Situation 3

Refugee / # 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

In Table 5 and Table 6, scenarios with and without blocked paths are compared. As can be seen, the blocked paths cause a significant shift in how miners are divided among the safe havens. There are two reasons for this. The first, and most important, is that due to blocked pathways miners may have to take detours. On their new route, the safe haven that was closest by before may now be relatively further away. Moreover, the route to a safe haven may be blocked entirely for a specific miner. This can lead to miners heading to a different destination than they would have gone to when all paths were available. The second, and less important, reason for the differences are the trapped miners. These miners will not make it to a safe haven at all, which can be found back in the numbers (note that this does not mean that trapped miners are considered less important individuals, but simply that their impact on the difference in division of workers among the safe havens is low).

Table 7 and Table 8 depict the differences with, and without the use of stamina categories in the algorithms. In these cases, no differences can be seen for the simulations with up to 200 miners. Minimal differences can be seen around the shafts for the simulations with 500 miners and more. This is because in simulations with up to 200 miners the refugee chambers have sufficient capacity. This means that miners are sent to the safe haven closest to them, no matter what stamina category they are in. In the simulations where the refugee chambers are full, miners that have a lower stamina will be favoured for a spot in the refugee chambers (if this is relatively closer by). This means that a miner with better stamina will be sent to a shaft in his/her place. This shaft, however, may not be the same as the weaker miner would have gone to. This leads to some differences in the division of miners around the shafts.

Miner X

The purpose of this section is to investigate how blocked pathways and stamina categories may influence the path of an individual miner. Miner X is part of the simulations executed in the second situation. He/she is in the highest stamina category and his/her paths are determined in the case where 500 miners are present. His/her paths are depicted in Table 9.

Table 9. Destinations Miner X

Scenario	Distance in Metres	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)

As can be seen, there is no difference between scenario 1 and 2. This means that, despite the fires, Miner X encounters no blocked paths on his/her route to safety in scenario 2. When looking at scenario 3, one can see that the distance that Miner X has to travel increases significantly compared to scenarios 1 and 2. This is because a weaker miner is favoured for a spot in refugee chamber 150, and Miner X has to head to a shaft in his place. Finally, in scenario 4, the path for Miner X is longer than in scenario 3. This is because one or more pathways on his original route are blocked because of fire, which means he has 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 may add significantly to the path of an individual miner. In the case of blocked pathways, this isn't necessarily undesirable. Although the route for an individual miner may be longer, it does avoid hazardous situations such as fires. Therefore, pathlength is exchanged for safety. That said, the added pathlength because of stamina categories does raise social and ethical questions. Can you expect a miner to travel further in favour 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 randomisation that was used in the simulations for the locations of miners and safe havens. Therefore, one should see these results as a simulation, and not as an actual escape solution.

Computation times

The average times to run the algorithms for the different scenarios are presented in Figure 3. These times only refer to solve the optimisation problem. Times to read, set up the model and reporting are negligible.

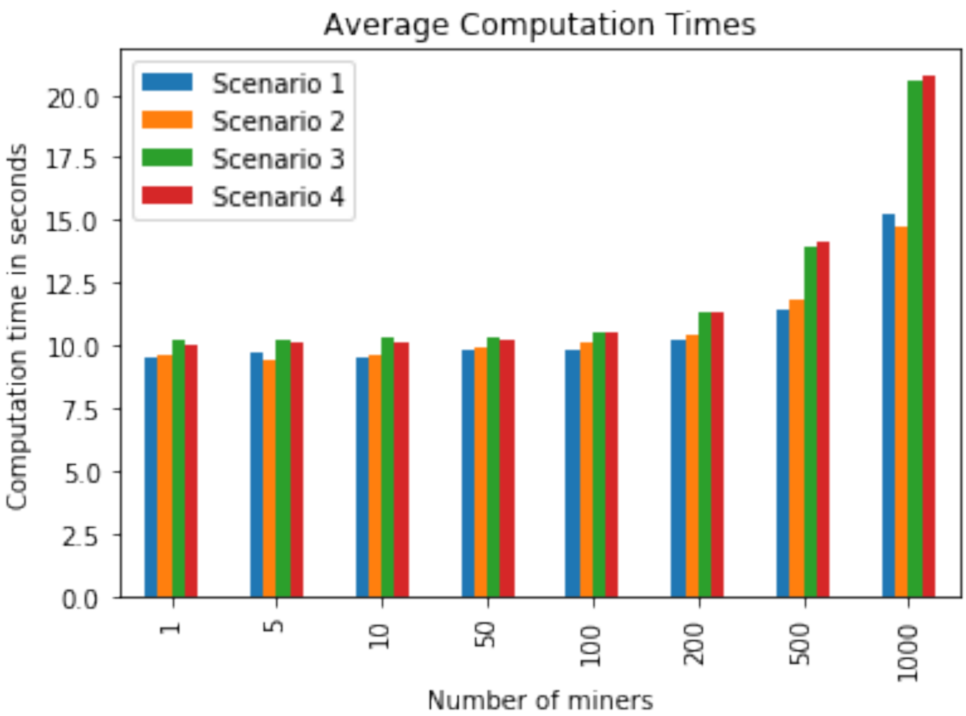

Figure 3. Average computation times for different scenarios.

As can be seen, blocked pathways have a negligible effect on the computation time of the algorithms. This is remarkable, as these scenarios require more execution of code. That said, it does mean there is virtually no downside to including the feature that filters out blocked pathways. Trapped miners are localised 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 the escape solutions.

Using stamina categories, on the other hand, does add to the computation time (up to five seconds in simulations with 1000 miners underground). Although this time difference will 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 the algorithm generates will be the same. In this case it is definitely 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 simulations were run on an everyday use laptop (MacBook Pro, 3.1 GHz Dual-Core Intel Core i5, 16 GB RAM 2133 MHz LPDDR3). If this system were to be implemented in an actual mining operation, most likely a specialised computer would be used. This could lead to a significant decrease in the time it takes for the algorithms to run.

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 gender and fitness discriminations into decision making. The categories can also 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 favour of a weaker colleague? That said, using these category types may be useful if the operation, for instance, uses vehicles such as trucks underground. These categories should perhaps then not be used to divide up miners, but could be used for a different purpose.

The use of the algorithm without the use of stamina categories can also raise ethical issues. For instance, if five miners are positioned at the same workstation, and the closest refugee chamber can only harbour four more workers, a decision will have to be made as to which miner misses out on a spot in this safe haven. This can be a tough dilemma. Who do you let to decide as to who cannot go to the closest safe haven; the algorithm, the miners, or the emergency controller? There is, probably, not one best answer to this question. That said, this issue does need to be resolved before a smart evacuation system can be brought into practice.

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 gasses 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 this does require that the conditions for every location underground are extensively monitored. Besides this, a system needs 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.

Practically, the current algorithm is quite crude: given more time and technical experience it could be made more compact, efficient, and elegant. Therefore, the main purpose of this algorithm is to prove that linear programming can be used to generate an escape solution for an underground mine. There are alternative methods to solving mathematical programming problems, e.g., other programming languages or Python packages. It would be worthwhile investigating which method would be the best for an actual mining environment.

CONCLUSION

In this paper, a mathematical programming component for a new smart evacuation algorithm for underground mines is proposed. The algorithm sets the evacuation model as a Minimum-Cost Network Flow Problem, which can be solved using any mathematical programming solver. This is done by setting the evacuation as an objective function, which is subject to a number of constraints. The objective function minimises the total distance travelled by all miners underground. The constraints are used to indicate the locations of workers and safe havens. The algorithms for this paper are written in the Python programming language, and solved by use of the GUROBI library (Version 8.0).

A total of four scenarios are tested: with- and without blocked pathways and with- and without dividing the miners up in stamina categories. Each of the scenarios were run for five different situations, with simulations 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 each different situation. The safe havens were chosen randomly by the author, and kept the same for all simulations.

It was found that in simulations with up to 200 miners, the capacity of the refugee chambers is sufficient. For simulations with 500 miners or more, the refugee chambers reach capacity, which leads to more miners heading to the shafts. The introduction of blocked pathways has a significant influence on how the miners are divided up among the safe havens. This has two reasons: the first being that blocked paths may cause a safe haven that was initially closest by, but is now relatively further away, or entirely unreachable. This will lead to miners being sent to a different safe haven. The second reason is that blocked pathways may cause miners to be trapped. They will therefore not reach a safe haven, which can be found back in the numbers. Stamina categories only have some effect on the division of miners around the shaft in simulations with 500 miners and more. This is because when a stronger miner is sent to a shaft in favour of a weaker miner, this does not need to be the same shaft as the latter would have originally gone to.

To give an idea how the different scenarios influence the path of an individual worker, Miner X was introduced (who comes from a simulation with 500 miners, and is 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 could be raised.

Introducing a feature that filters out blocked pathways and localises 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 do 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.

ACKNOWLEDGEMENTS

This research is based on the MSc thesis of the first author and is a 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 Keersemaker, for his feedback on the thesis.

REFERENCES

Adjiski, V., Mirakovski, D., Despodov, Z., & Mijalkovski, S. (2015). Simulation and optimization of evacuation routes in case of fire in underground mines. *Journal of Sustainable Mining, 14*(3), 133-143.

Baek, J., Choi, Y., Lee, C., Suh, J., & Lee, S. (2017). BBUNS: Bluetooth beacon-based underground navigation system to support mine haulage operations. *Minerals, 7*(11), 228.

Bari, A. (2018). 4.2 All Pairs Shortest Path (Floyd-Warshall) - Dynamic Programming.

Biswas, P., Mishra, P., & Mahanti, N. (2005). Computational Efficiency of Optimized Shortest Path Algorithms. *IJCSA, 2*(2), 22-37.

Brenkley, D., Bennett, S., & Jones, B. (1999). *Enhancing mine emergency response.* Paper presented at the Proceedings of 28th International Conference on Safety in Mines Research Institutes, Sinaia, Romania.

Chasko, L. L., Conti, R. S., Lazzara, C. P., & Wiehagen, W. J. (2005). Fire response preparedness for underground mines.

Chehri, A., Fortier, P., & Tardif, P. M. (2009). UWB-based sensor networks for localization in mining environments. *Ad Hoc Networks, 7*(5), 987-1000.

Conti, R. S. (2001). *Emerging technologies: aiding responders in mine emergencies and during the escape from smoke-filled passageways.* Paper presented at the Proceedings of the Northwest Mining Association's 107th Annual Meeting. Spokane, WA.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik, 1*(1), 269-271.

Gaab, S. J. (2019). *Evaluation of Smart Underground Mine Evacuation Efficiency through Virtual Reality.*

Guangwei, Y., & Dandan, F. (2013). Escape-route planning of underground coal mine based on Improved Ant Algorithm. In: Hindawi Publishing Corporation. Mathematical Problems in Engineering.

Hong, Y., Li, D., Wu, Q., & Xu, H. (2018). Dynamic route network planning problem for emergency evacuation in restricted-space scenarios. *Journal of Advanced Transportation, 2018*.

Hougardy, S. (2010). The Floyd–Warshall algorithm on graphs with negative cycles. *Information Processing Letters, 110*(8-9), 279-281.

Meij, R. (2020). " Development of a new smart evacuation modelling technique for underground mines using Mathematical Programming", MSc Thesis, http://resolver.tudelft.nl/uuid:e0a9091e-fd24-4d61-96e7-70086e2586cf

Mirjalili, A. (2018). How the Ant Colony Optimization algorithm works.

Mohapatra, A. G., Keswani, B., Nanda, S., Ray, A., Khanna, A., Gupta, D., & Keswani, P. (2020). Precision local positioning mechanism in underground mining using IoT-enabled WiFi platform. *International Journal of Computers and Applications, 42*(3), 266-277.

Niu, Q., Yang, X., & Yin, Y. (2018). IPL: Image-Assisted Person Localization for Underground Coal Mines. *Sensors, 18*(11), 3679.

Radinovic, G., & Kim, K. (2008). Feasibilty study of RFID/Wi-Fi/BlueTooth wireless tracking system for underground mine mapping–Oklahoma. *procedings of "Incorporating Geospatial Technologies into SMCRA Business Processes", Atlanta, GA*.

Rusu, S. R., Hayes, M. J. D., & Marshall, J. A. (2011). *Localization in large-scale underground environments with RFID.* Paper presented at the 2011 24th Canadian Conference on Electrical and Computer Engineering (CCECE).

Winston, W. L., & Goldberg, J. B. (2004). *Operations research: applications and algorithms* (Vol. 3): Thomson Brooks/Cole Belmont.

Yiu, S., Dashti, M., Claussen, H., & Perez-Cruz, F. (2017). Wireless RSSI fingerprinting localization. *Signal Processing, 131*, 235-244.

Dr. Masoud Soleymani Shishvan

Assistant Professor
Delft University of Technology

Dr. Shishvan is assistant professor of Resource Engineering at Delft University of Technology since May 2019. He is a mining engineering with deep affection for applied mathematics. His research interests are mining process optimization, application of AI in mining and IoT for smart connected mining.