Application of Particle Swarm Optimization Algorithm to Optimize Stope Layout for Underground Mines

T. M. Mmola, A. S. Nhleko and J. M. Atherfold

1 Introduction

Mining projects have long turnaround times and require large start-up capital to build and operate. The objective of mine production is to maximize return on investment, which is derived from the extraction and sale of the mineral. The return on investment will depend on the physical location of the ore, the mining layout and extraction sequence, technical factors associated with the orebody, grade of the orebody, and the available mining methods [8]. It is in the early planning stages where a mine has the greatest level of flexibility to make decisions on these economic and technical criteria for operating a mine.

Thorough planning done in advance of constructing the mine lowers the risk of failure. Once the construction of the mine begins, the ability to alter the mine design diminishes exponentially as the mine matures [6]. Therefore, the mine engineer is required early on in a mining project to make long-term decisions that must optimize the cost efficiency and profitability of the mine operation.

The limited number of tested operations research (OR) techniques and the lack of tools and appropriate computer programs to address underground mine planning problems is an issue of concern to mining professionals [9]. This lack of software limits a company's capacity to develop underground mine plans that maximizes the net present value (NPV) of the project [1, 8]. There is a recognized need by

T. M. Mmola
School of Mining, Metallurgy and Chemical Engineering, University of Johannesburg, Johannesburg, South Africa

A. S. Nhleko (✉)
School of Mining Engineering, University of the Witwatersrand, Johannesburg, South Africa
e-mail: Sihesenkosi.Nhleko@wits.ac.za

J. M. Atherfold
School of Computer Science and Applied Mathematics, University of the Witwatersrand, Johannesburg, South Africa

E. Widzyk-Capehart et al. (eds.), *Proceedings of the 27th International Symposium on Mine Planning and Equipment Selection - MPES 2018*,
https://doi.org/10.1007/978-3-319-99220-4_17

the mining industry for improved software tools to support the planning, design and operation of underground mines [2]. The strategic planning tools could help to minimize the potential for suboptimal decisions being made at the outset of an operation by reviewing many different alternatives.

2 Background

In the mine design process of underground mines, the mining engineer must first select a mining method that is amenable to extracting the orebody and then decide on a cut-off grade for extracting the orebody. The next step is to create a stope design that maximizes the value of the mine. A stope is an underground production area from which ore is extracted from the surrounding rock mass [11]. The mine engineer will then design the access to the identified stopes. In addition, the mining engineer must sequence the extraction order of the stopes with the purpose to maximize economic ore recovery. Throughout this process, the mining engineer must consider the technical factors associated with the orebody and economic factors associated with the selected mining method [6]. Therefore, the design of the mine stopes, mainly their dimensions and location, is a critical aspect of the mine design process.

Historically, the mining engineer would design the stopes manually, which is a time-consuming process. Furthermore, the use of rules-of-thumb in determining the dimensions and locations of the stopes would be common practice. However, rules-of-thumb calculations do not always produce optimized designs. Since the subsequent introduction and proliferation of computers, the use of software applications with built-in algorithms that can automatically design and optimize the stope layout has increased. While this has reduced the time required for the stope design process, the literature indicates that none of the current algorithms are able to guarantee the optimum stope design [10]. Evolutionary algorithms and more specifically the particle swarm optimization (PSO) algorithm have been used successfully in a variety of industrial optimization problems. This begs the question: can the PSO algorithm generate an optimum 3D underground stope layout?

3 The Particle Swarm Optimization Algorithm

The PSO algorithm optimizes a problem by generating a population of particles, representing candidate solutions, and having each particle iteratively try to improve on its solution with regard to a given measure of quality. Each particle will evaluate its current solution quality against the personal best solution it has achieved so far and also the global best solution found by any particle in the population. Each particle moves in search of better solutions throughout the search space according to simple mathematical formulae that define the particle's position and velocity over time [7].

Table 1 Selected PSO parameters based on literature review

Parameter	Description	Value
ωmax	Maximum inertia coefficient	0.9
ωmin	Minimum inertia coefficient	0.4
K	Total number of iterations	100
$c_1,\ c_2$	Velocity coefficients	$c_1 = c_2 = 2$

To search for the optimal solution, the velocity and positions of each particle are updated by the following equations:

$$\begin{cases} v_i^d(k+1) = \omega v_i^d(k) + c_1 r_1\left(P_{\text{best}}^d(k) - P_i^d(k)\right) + c_2 r_2\left(g_{\text{best}}^d(k) - P_i^d(k)\right) \\ P_i^d(k+1) = P_i^d(k) + v_i^d(k+1) \end{cases}, \tag{1}$$

where c_1 and c_2 are acceleration constants regulating the relative velocities with respect to the personal best and global best positions, respectively; r_1 and r_2 are $N \times 1$ vectors of random numbers drawn from a uniform distribution in the interval (0,1); and ω is an inertia parameter given by

$$\omega = \omega_{\max} - \frac{\omega\text{max} - \omega\text{min}}{K} k, \tag{2}$$

where ωmax and ωmin are the initial and final weights, respectively, k is the iteration number and K is the total number of iterations.

The advantages of the PSO are that it is simple to code and it only requires the problem and a few parameters to solve [7]. We will look at the problem definition then encoding strategy and the parameters and then apply the PSO to an orebody.

3.1 Parameter Selection

One of the advantages of the PSO compared to other algorithms is the relatively few number of parameters that have to be tuned in the algorithm [7]. The parameter values used for this research were based on a literature survey of the existing research on parameter selection for the PSO algorithm [3, 5]. These parameters are indicated in Table 1.

The PSO is known to be very sensitive to the choice of parameters and parameter selection is one of the most important aspects in the PSO algorithm. It is accepted that generally the choice of the parameters will be problem dependent and that parameter hyper-tuning will be often required.

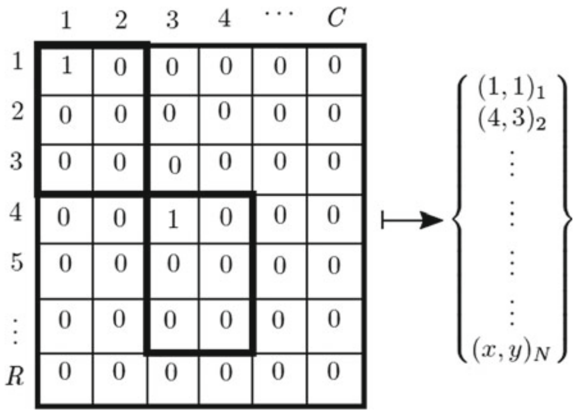
4 Model Details

As there was no readily available application to run the PSO on the stope boundary optimization problem (SBOP), it was necessary to code the SBOP and PSO algorithm from the ground up. The Python programming language was used for this purpose and the modeling strategy is discussed.

4.1 Problem Formulation and Modelling

The encoding of the problem is specified in two dimensions with the intention of clarifying the strategy for modelling the problem. It was assumed that the sublevel open stoping method would be used to mine the deposit and three constraints associated with this mining method, namely overlap constraint, level constraint, and uniqueness constraint, had to be considered in encoding the PSO algorithm. Figure 1 illustrates an example of a section of an orebody, and how the selected mine configuration is encoded such that it can be passed to the PSO algorithm.

In this trivial example, there are R rows and C columns in the orebody, representing the orebody extent. Each block represents a block in the orebody. The stope size is fixed at 3 × 2 blocks, and there are N possible stopes. The stopes selected in this particular configuration are marked in bold. The starting corner of each selected stope is marked with the number one, and the rest of the ore body is padded with zeros. The coordinates of all the ones in the ore body are then found, and stored in a set, as illustrated in Fig. 1. This set of coordinates forms one member of the population. Each member of the population is therefore an entire mine configuration. Figure 1 is an illustration of the encoding of the 2D SBOP. In 3D, the dimensionality of the problem is $N \times 3$.



	1	2	3	4	⋯	C
1	1	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	1	0	0	0
5	0	0	0	0	0	0
⋮	0	0	0	0	0	0
R	0	0	0	0	0	0

$$\mapsto \left\{ \begin{array}{c} (1,1)_1 \\ (4,3)_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ (x,y)_N \end{array} \right\}$$

Fig. 1 Encoding of a mine configuration

4.2 Fitness Evaluation

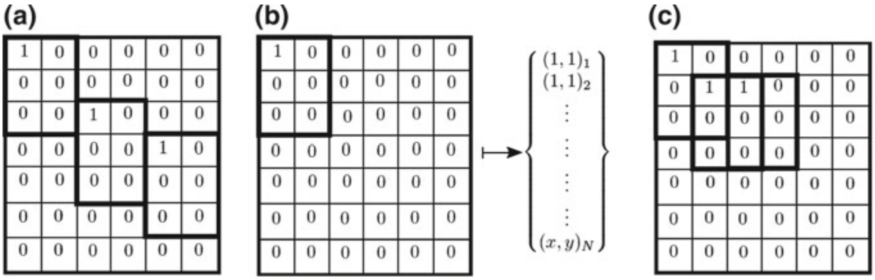
The PSO is initialized by generating random solutions, i.e. particles, which represent a specific mine layout. The number of stopes that may be used in the mine layout are predefined. Therefore, each particle will consist of the predefined number of stopes randomly selected from the set of all possible stopes.

The fitness of a particle is a direct function of the final value of the mine, i.e., the sum of the values of the selected stopes in the mine layout. The net smelter return (NSR) was used as the measure of value. Incorporated into the fitness evaluation are three important constraints; the level constraint, uniqueness constraint, and overlap constraint. The nature of these constraints is illustrated in Fig. 2.

The level violation indicates multiple stopes which have blocks on different mining levels. This is not allowed according to the design of the mine as each stope must lie within the defined level spacing. Uniqueness violation occurs when two stopes occur at the same location. Since a stope cannot be mined twice, the number or stopes needs to be explicitly specified in the model. Overlap violation occurs when stopes are overlapping. These stopes may or may not be on the same level. This incurs a penalty because stopes may not overlap, again because once a stope or a portion of it has been mined, it cannot physically be mined again. The fitness of the configuration is therefore taken to be the linear combination of the mine value, and the three penalties, i.e.,

$$\text{Fitness} = V_{\text{Mine}} - k_1 \times P_{\text{Level}} - k_2 \times P_{\text{Unique}} - k_3 \times P_{\text{Overlap}}, \tag{3}$$

where V_{Mine} is the calculated value of the mine configuration, P's are the penalties incurred by each respective constraint violation, and k's are constants, chosen large enough such that even if one penalty occurs, the fitness will indicate that the mine configuration will not be economically viable. This ensures that all economically viable configurations follow all the constraints. The goal of the algorithm is to maximize the fitness.



(a)

1	0	0	0	0	0
0	0	0	0	0	0
0	0	1	0	0	0
0	0	0	0	1	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

(b)

1	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

(c)

1	0	0	0	0	0
0	1	1	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

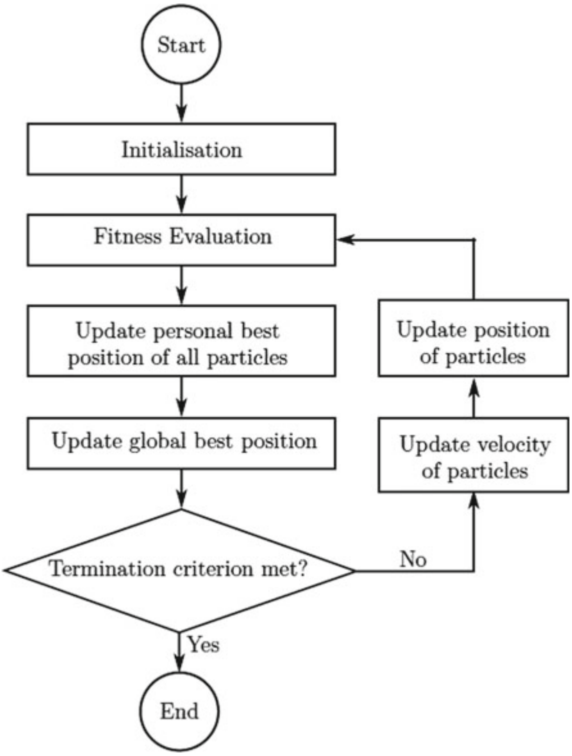
Fig. 2 Visual representation of **a** level constraint violation; **b** uniqueness constraint violation; and **c** overlap constraint violation

The experimental method utilizing PSO for the purpose of mine configuration optimization can be described as follows:

1. Initialize swarm size, maximum number of generations, and initial velocities and positions of each of the particles, where each particle represents a particular mine configuration.
2. Calculate the fitness of each member according to Eq. 3.
3. Update model parameters if necessary; the personal best position of each particle and the global best position at the current time step.
4. Update the velocity and positions of each particle according to Eq. 1.
5. The algorithm terminates when the maximum number of generations occurs. The output is an optimal mine configuration.

This method was conducted for a varying number of stopes of a fixed size. 20 experiments were conducted for each stope number. Figure 3 shows the flowchart for the mine configuration optimization.

Fig. 3 Flowchart of PSO algorithm


Start
Initialisation
Fitness Evaluation
Update personal best position of all particles
Update global best position
Termination criterion met?
No
Yes
End
Update velocity of particles
Update position of particles

5 Experimental Results

The optimization of a mine configuration was attempted, using PSO as an optimization tool. Training data from a conceptual orebody was used to test the optimization algorithm, and the results are discussed.

The algorithm requires a regularized economic block model. The orebody model used in this study represents a theoretical gold deposit. The block model consists of 15,572 blocks of a uniform block size. The geological attributes assigned to each block are the gold grade and the rock density. The metal content per block was determined from these two attributes, taking into consideration the block size. An economical value, the Net Smelter Return (NSR), was calculated for each block based on assumptions of the mining costs, processing costs, logistical costs, and metal price. The economic block model data is summarized in Table 2.

The block model data was then imported into the Python script that was developed for this research. Then, the PSO algorithm optimization was run using a fixed stope size of 10 m × 10 m × 20 m along the x, y, and z axis, respectively.

Figure 4 shows the final results of all experiments. The maximum mine values are plotted as a function of the number of stopes. The maximum mine value found by the PSO algorithm is about 22,000 with 12 stopes.

Figure 5 shows how the mean and maximum fitness in the population change with iteration number from one of the experimental runs using 13 stopes. Figure 5a illustrates the convergence process of the algorithm. Convergence does not mean that the population has reached an optimum (local or global). Rather it means that the population has reached an equilibrium state, i.e., the particles converged to a point, which may not be an optimum point [4].

From Fig. 5a, the algorithm clearly converges to a maximum value within the 100 iterations. In this particular case, the maximum value is relatively small, at approximately 4400 and therefore this value is a local maximum, not the global maximum. The algorithm getting trapped in local maxima is the reason multiple experiments are run.

From Fig. 5b, the effect of the heavy penalties on the constraints violations can be observed. The mean mine value starts off at approximately −3,800,000, indicating a large number of constraints violations. The mean mine value then increases with the number of iterations to about 4000 as the violations are resolved.

Table 2 Summary of economic block model data

Attribute	Value
Number of blocks	15,572
Block sizes (x, y, z)	5 m × 5 m × 5 m
Rock density variation	$2.8 - 3.6$ t/m^3
Net smelter return variation	$0.6 - 301$ \$/t

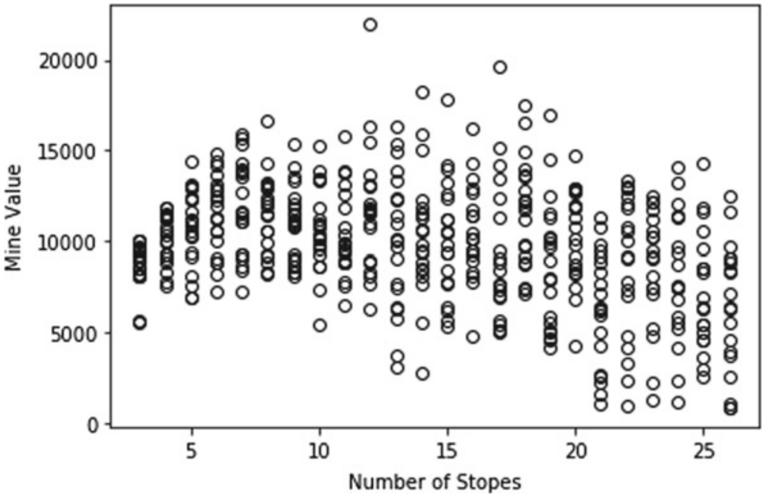
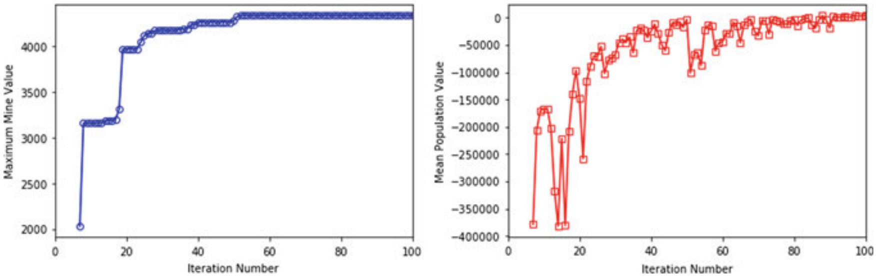


Fig. 4 Maximum mine values for all experiments



(a) Maximum mine value

(b) Mean mine value

Fig. 5 Results from an experimental run with 13 stopes

6 Conclusions

This paper proposes an approach to the SBOP in which the mine layout is optimized using the PSO algorithm. Such an approach has not previously been demonstrated as feasible for the SBOP. The reported experimental results show the convergence progress of the algorithm as well as the maximum mine value obtained by the optimization. Furthermore, the results indicate that the algorithm is able to handle the specified mining constraints associated with the SBOP. Moreover, the results indicate that a PSO approach is feasible, and warrants further investigation.

Further work is required to refine the algorithm, in particular, with respect to parameter selection. Since the PSO is not problem dependent, any other mining constraints, such as minimum pillar sizes, or mining parameters, such as variable stope sizes, can be simply defined in the problem's objective function. The PSO algorithm will then run an optimization on the defined objective function.

References

1. Ataee-Pour, M.: A heuristic Algorithm to Optimise Stope Boundaries. University of Wollongong (2000)
2. Australian Minerals Industry Research Association International: AMIRA International Annual Report 2015–2016 (2016)
3. Clerc, M., Kennedy, J.: The particle swarm—explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2002)
4. Engelbrecht, A.P.: Computational Intelligence. Wiley, Chichester, UK (2007)
5. Engelbrecht, A.P.: Particle swarm optimization: global best or local best? In: Proceeding of 1st BRICS Congress on Computational Intelligence. BRICS-CCI 2013 vol. 1, pp. 124–135 (2013)
6. Kazakidis, V.: Operating Risk: planning for Flexible Mining Systems. University of British Columbia (2001)
7. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN '95—International Conference on Neural Networks IEEE, pp. 1942–1948 (1995)
8. Little, J., Knights, P., Topal, E.: Integrated optimization of underground mine design and scheduling. J. South. African Inst. Min. Metall **113**(10), 775–785 (2013)
9. Musingwini, C.: Presidential address: optimization in underground mine planning-developments and opportunities. J. South. African Inst. Min. Metall **116**(9), 809–820 (2016)
10. Nhleko, A.S., Tholana, T., Neingo, P.N.: A review of underground stope boundary optimization algorithms. Resour. Policy. **56**, 59–69 (2018)
11. Sandanayake, D.S.S., Topal, E., Asad, M.W.A.: A heuristic approach to optimal design of an underground mine stope layout. Appl. Soft Comput. **30**, 595–603 (2015)