STOPE BOUNDARY OPTIMISATION FOR UNDERGROUND MINES

Supervisor: Prof. Montaz Ali

Alex Alochukwu, Babatunde Sawyerr, John Atherfold, Krupa Prag, Micheal Olusanya, Patience Adamu, Peter Popoola, Sakirudeen Abdulsalaam, Vincent Langat

January 19, 2018

GROUP 3: MATHEMATICS IN INDUSTRY STUDY GROUP (MISG) 2018

INTRODUCTION

The Stope Boundary Optimization Problem (SBOP) involves choosing a certain configuration of stopes which maximizes the Net Profit Value, (NPV), subject to the stope dimension constraints.

INTRODUCTION

Our approach involves the following:

- The development of a mathematical model for the SBOP
- The exploration of various optimization algorithms for the SBOP
- The development of a hybrid algorithm for the SOP which contains components of Dynamic Programming (DP) and Particle Swarm Optimization (PSO).

2D Schematic of Stope Configuration Instance

15	0	3	-8	7	32
8	74	1	19	27	-4
-22	10	90	2	-71	4
30	60	32	-40	3	-20
55	1	-13	-44	22	85
-22	33	64	2	-9	4

Mathematical Modelling

For the sake of simplicity, we developed the mathematical formulation in 2D, and extend it to 3D. The following assumptions were made in creating the 2D mathematical model:

- Let the mining area be represented by a grid with dimension, n x m.
- The grid is made up of distinct blocks with predefined values
- The stope dimension is fixed for 2D case, say α x β .
- The decision variable is binary,
- To ensure that the stopes are on the same level for easy mining, we use the following strategy: If $x_{ij} = 0$, then move to x_{ij+1}; If $x_{ij} = 1$, then move to $x_{ij+\beta}$; Once the level has been exhausted, move to $x_{i+\alpha j}$, and repeat the steps.

MATHEMATICAL FORMULATION: 2D

$$\text{Maximize} \quad \sum_{i=1}^{n-p} \sum_{j=1}^{m-q} V_{ij} x_{ij}, \tag{1}$$

Subject to:

$$\sum_{i}^{i+p} \sum_{j}^{j+q} x_{ij} \leq 1, \forall\, i \in \{1, \cdots n-p\}, \forall\, j \in \{1, \cdots m-q\} \tag{2}$$

$$x_{ij} - \sum_{j'=j+1}^{j+q} x_{ij'} = 1 \quad \forall\, i \in \{1, \cdots n-p\}, \forall\, j \in \{1, \cdots m-q\} \tag{3}$$

$$x_{ij} - \sum_{i'=i+1}^{i+p} x_{i'j} = 1 \quad \forall\, i \in \{1, \cdots n-p\}, \forall\, j \in \{1, \cdots m-q\} \tag{4}$$

MATHEMATICAL FORMULATION: 2D

$$x_{ij} - \sum_{i'=i+1}^{i+p} \sum_{j'=j+1}^{j+q} x_{i'j'} \quad = \quad 1 \quad \forall\, i \in \{1, \cdots n-p\}, \forall\, j \in \{1, \cdots m-q\} \tag{5}$$

$$\text{where} \quad V_{ij} = \sum_{i}^{i+p} \sum_{j}^{j+q} u_{ij}, \quad p = \alpha - 1 \text{ and } q = \beta - 1. \quad x_{ij} \in \{0, 1\}$$

MATHEMATICAL FORMULATION: 3D

Next, we extend this model to the 3D SBOP, allowing for variable stope dimensions:

$$\text{Maximize} \quad \sum_{i=1}^{n-p} \sum_{j=1}^{m-q} \sum_{k=1}^{s-r} V_{ijk} x_{ijk}, \tag{6}$$

Subject to:

$$\sum_{i}^{i+p} \sum_{j}^{j+q} \sum_{k}^{k+r} x_{ijk} \leq 1, \forall\, i \in \{1, \cdots n-p\}, \forall\, j \in \{1, \cdots m-q\}, \forall\, k \in \{1, \cdots s-r\}$$

$$x_{ijk} - \sum_{j'=j+1}^{j+q} x_{ij'k} = 1 \quad \forall\, i \in \{1, \cdots n-p\} \forall\, j \in \{1, \cdots m-q\}, \forall\, k \in \{1, \cdots s-r\}$$

MATHEMATICAL FORMULATION: 3D

$$x_{ijk} - \sum_{i'=i+1}^{i+p} x_{i'jk} = 1 \quad \forall\, i \in \{1, \cdots n-p\}, \forall\, j \in \{1, \cdots m-q\}, \forall\, k \in \{1, \cdots s-r\}$$

$$x_{ijk} - \sum_{k'=k+1}^{k+r} x_{ijk'} = 1 \quad \forall\, i \in \{1, \cdots n-p\}, \forall\, j \in \{1, \cdots m-q\}, \forall\, k \in \{1, \cdots s-r\}$$

$$x_{ijk} - \sum_{i'=i+1}^{i+p} \sum_{j'=j+1}^{j+q} \sum_{k'=i+1}^{k+s} x_{i'j'k'} = 1 \quad \forall\, i \in \{1, \cdots n-p\}, \forall\, j \in \{1, \cdots m-q\},$$

$$\forall\, k \in \{1, \cdots s-r\}$$

MATHEMATICAL FORMULATION: 3D

$$x_{ijk} - \sum_{i'=i+1}^{i+p} \sum_{j'=j+l}^{m-1} \sum_{k'=k+1}^{s-1} x_{i'j'k'} = 1 \quad \forall\, i \in \{1, \cdots n-p\}, \forall\, j \in \{1, \cdots m-q\},$$

$$\forall\, k \in \{1, \cdots s-r\}$$

$$\text{where} \quad V_{ijk} = \sum_{i}^{i+p} \sum_{j}^{j+q} \sum_{k}^{k+r} u_{ijk}, \tag{7}$$

$$p = \alpha - a\ , q = \beta - b \text{ and } r = \gamma - 1 \tag{8}$$

$$x_{ijk} \in \{0, 1\} \tag{9}$$

$$a = 1, ..., n,\ \ b = 1, ..., m. \tag{10}$$

Heuristic Solution Methods

We have been able to successfully implement and test the following heuristics for the SBOP:

- Maximum value algorithm
- Multi-Start Algorithm
- DP-inspired Heuristic
- Particle Swarm Optimisation

DP-Inspired Heuristic

This heuristic is inspired by the Dynamic Programming idea of sub-dividing a large problem into smaller ones, solving them, and then combining the solutions to get a solution to the large version of the problem. The algorithm is as follows:

- Input: Array containing the value of each block, stope size, mining site dimentions. Let K be the stope size, fd, the width of the site, and fl the length of the site.
- STEP 1: Create an array $(fd - k + 1 \times fl - k + 1)$ that stores all the possible stopes.
- STEP 2: Get the value of each possible stope.
- STEP 3: Get all the possible configurations with their values, skipping stopes with negative values
- STEP 4: Pick the stope layout with the highest revenue.

GREEDY ALGORITHM: 2D

STEP 1: Calculate the V_{ij} by summing the u_{ij} for each possible stope.

STEP 2: Demarcate the whole orebody into levels of size equal to the height of the stope.

STEP 3: In each level determine the the possible number of stopes that can be extracted based on their values(A stope with a maximum value chosen first,followed by a stope with the second maximum value,and so on, ensuring that there is no stope overlap.

STEP 4: Determine the the Net Present Value(NPV) by summing up the V_{ij} for all the possible stopes that can be extracted.

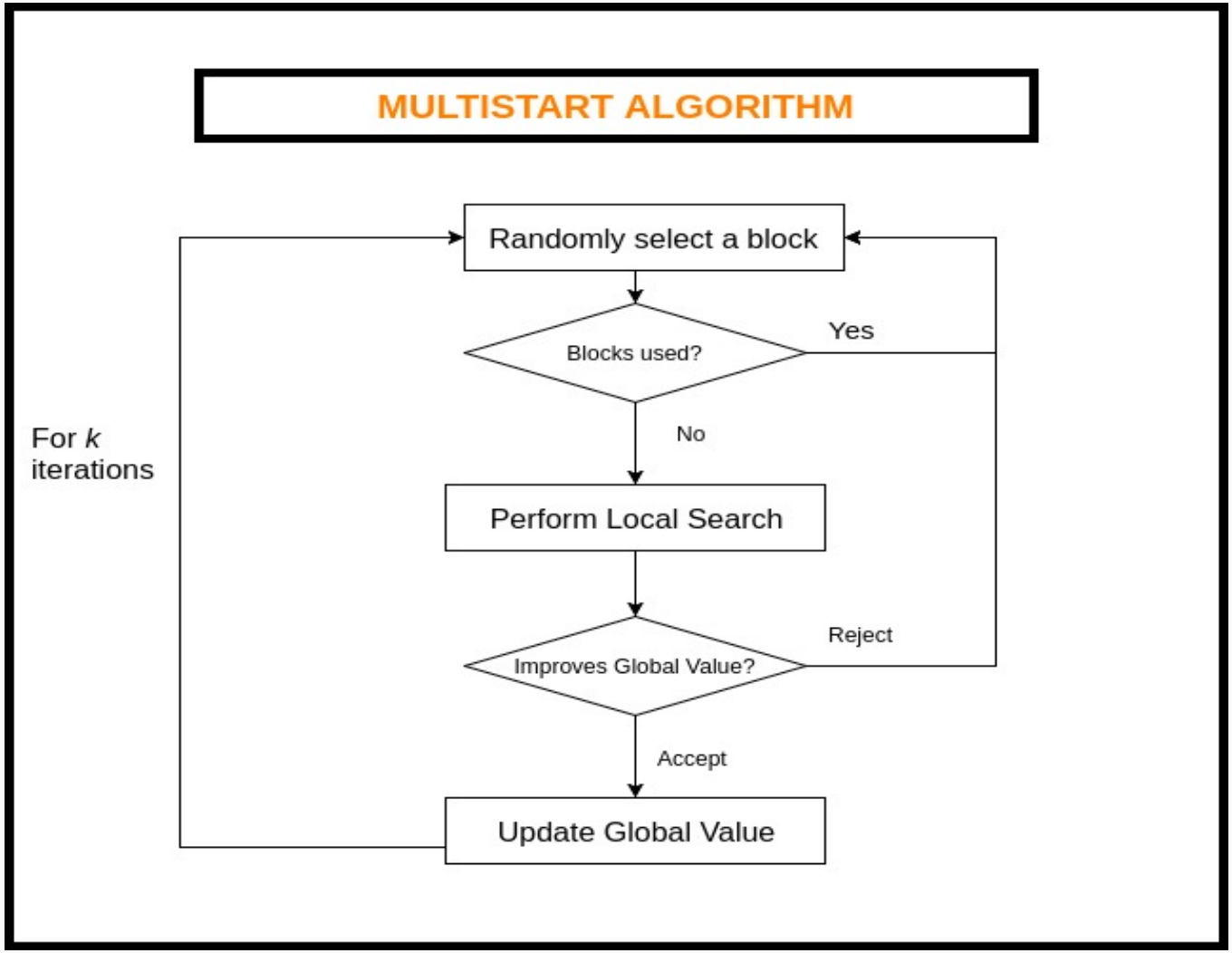
BLOCK VALUES AT EACH CELL

$$\begin{pmatrix}
-20 & -20 & 2 & 7 & 9 & -1 & 4 & -5 & 7 \\
9 & 6 & 3 & 2 & 3 & -13 & 23 & 56 & -21 \\
1 & -2 & 4 & 5 & 7 & -40 & 0 & -11 & 51 \\
3 & 0 & 5 & 4 & -60 & 30 & 14 & 1 & 31 \\
-4 & -7 & 6 & 3 & -10 & 4 & 12 & 14 & 34 \\
3 & 0 & 7 & 2 & -23 & 2 & 4 & 22 & -15 \\
-4 & -7 & 8 & 1 & 12 & 12 & 3 & 1 & 11 \\
5 & 2 & 6 & 2 & 3 & 0 & 1 & -50 & -20 \\
11 & 8 & 0 & -1 & 17 & -42 & 11 & 0 & 17 \\
17 & 0 & 3 & 4 & 23 & -2 & 2 & 65 & 23 \\
-3 & -6 & 2 & -60 & 18 & 11 & 1 & 54 & 12 \\
8 & 5 & 6 & 3 & 1 & 0 & 3 & 3 & 2 \\
10 & 7 & -7 & -2 & 18 & 4 & 14 & 9 & -57 \\
-8 & -11 & 0 & -50 & 12 & 18 & -45 & 21 & 7
\end{pmatrix}$$

STOPE VALUES AT EACH CELL

$$\begin{pmatrix}
-25 & -9 & 14 & 21 & -2 & 13 & 78 & 37 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 7 & 18 & -44 & -63 & 4 & 4 & 72 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-8 & 6 & 18 & -28 & -27 & 22 & 52 & 55 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-4 & 9 & 17 & 18 & 27 & 16 & -45 & -58 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
36 & 11 & 6 & 43 & -4 & -31 & 78 & 105 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & 7 & -49 & -38 & 30 & 15 & 61 & 71 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-2 & -11 & -59 & -22 & 52 & -9 & -1 & -20 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Multi-Start Algorithm



MS - Results Mine

Max Search Multi-Start

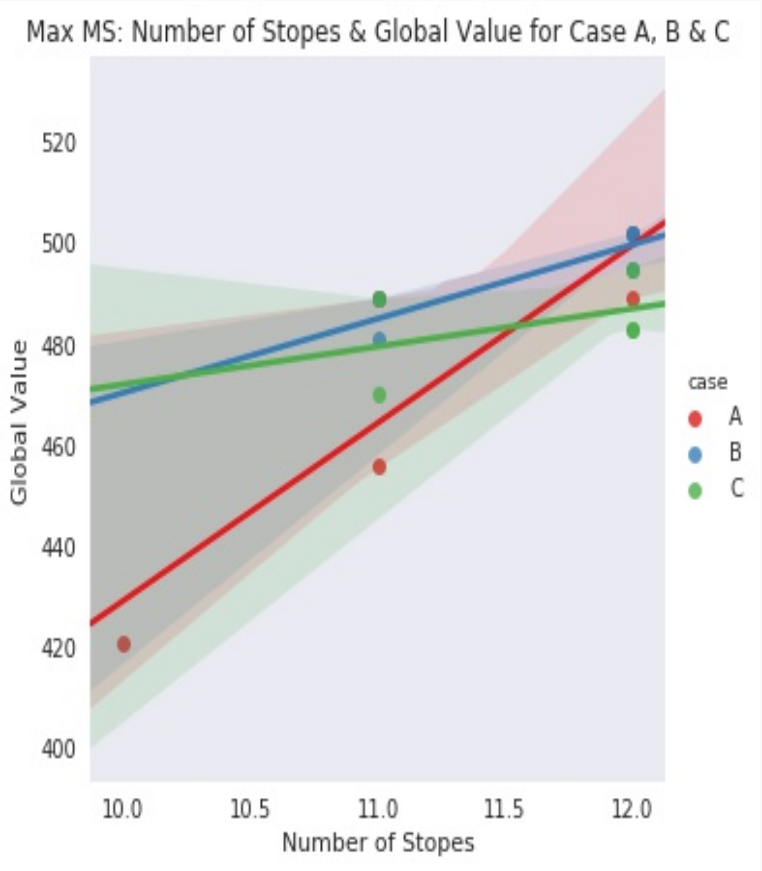
	Case A	Case B	Case C
Global Searches	14	$2 * 14$	14^2
Local Searches	9	2*9	9^2
Average Stopes	11.2	11.6	11.6
Average Global Value	471.4	493.8	484.0
Stope Fraction	0.36	0.37	0.37

Random Search Multi-Start

	Case A	Case B	Case C
Global Searches	14	$2 * 14$	14^2
Local Searches	9	2*9	9^2
Average Stopes	2.6	2.6	2.4
Average Global Value	112.8	78.8	74.6
Stope Fraction	0.08	0.08	0.07

MS - Results Mine

- Is a greedy modification beneficial?



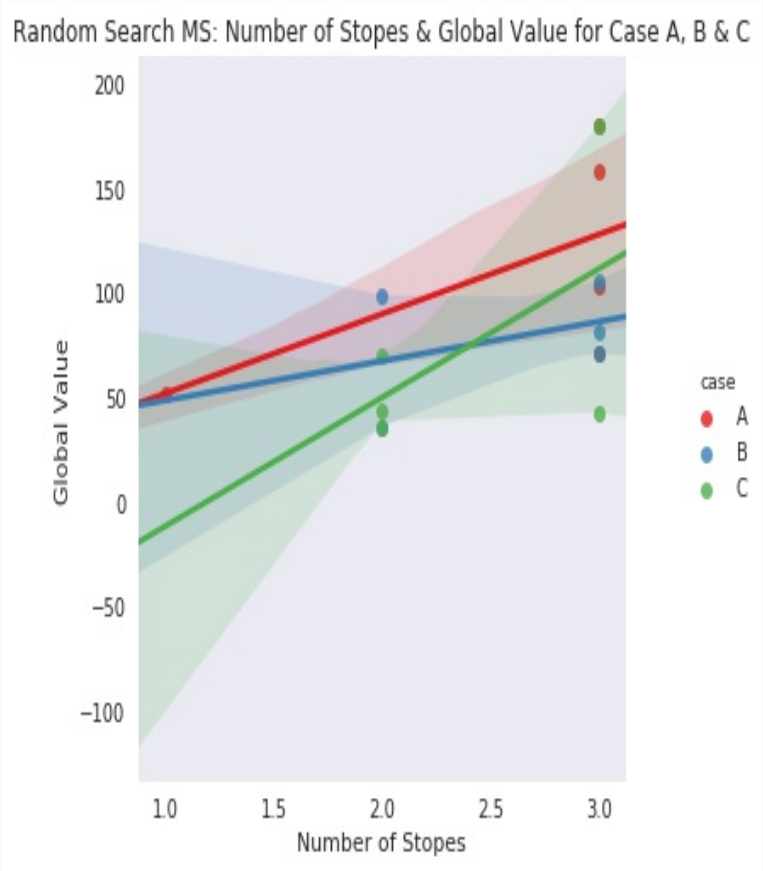
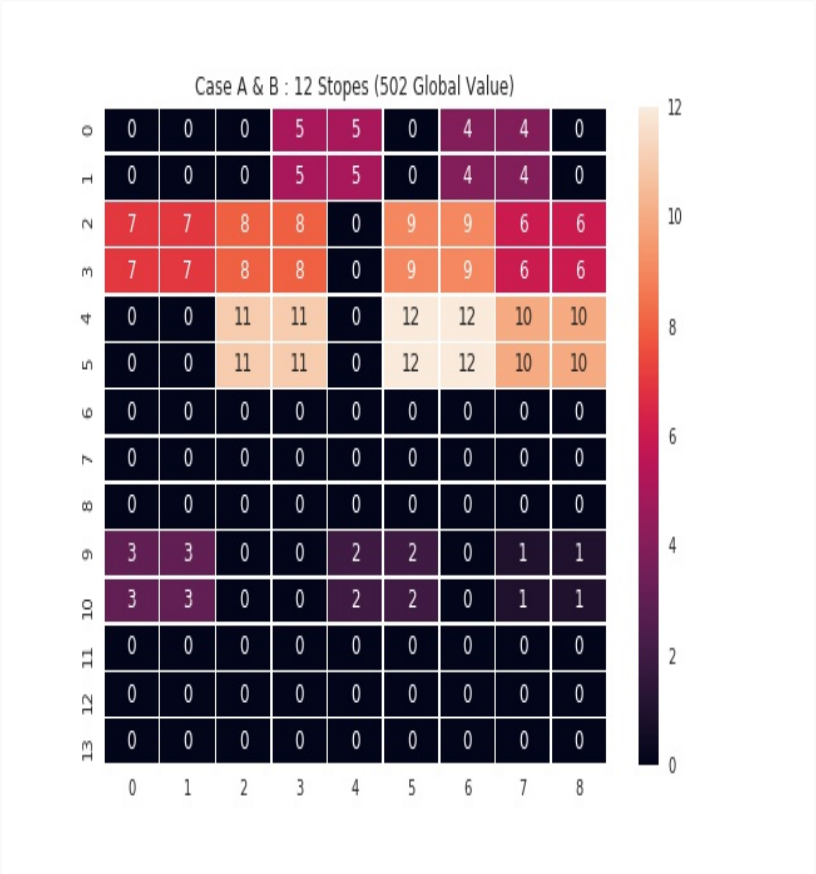


Figure: MS Evaluation using Max Search and Random Search

MS - Results Mine



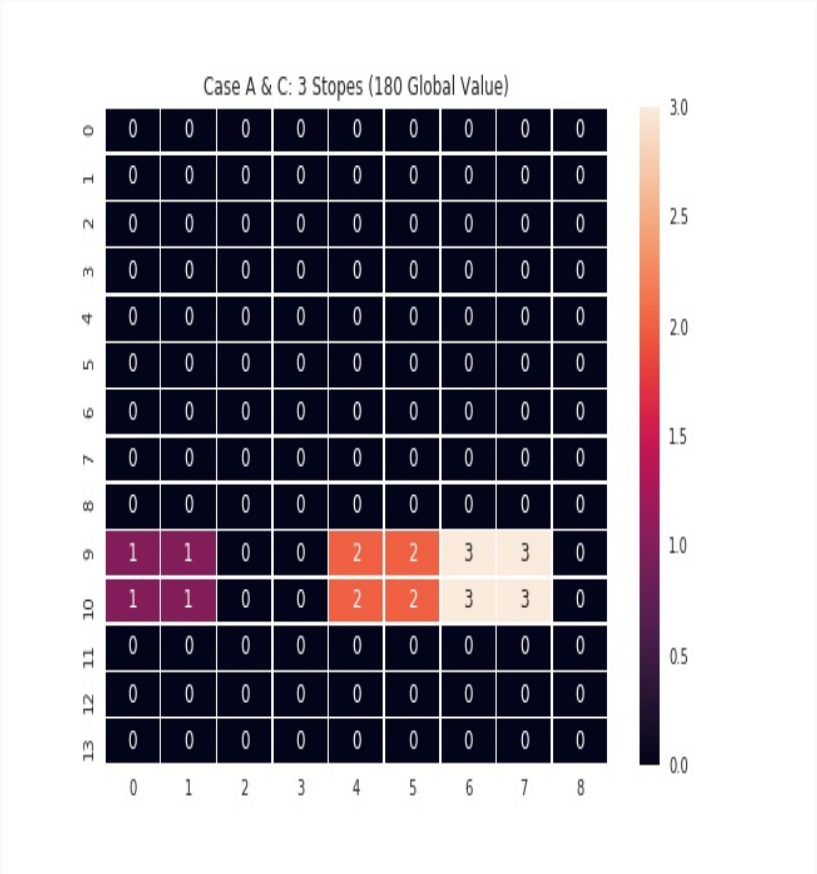


Figure: MS Evaluation using Max Search and Random Search

MS - Overview

- Modification of Pure Random MS algorithm improves solutions.
- Number of search iterations influences the outcome.
- Possible improvement implementations:
  - Parallel MS
  - Alternate local search techniques

Particle Swarm Optimisation

- Simulates the motion of flocking birds
- *Population* - Initial positions of each particle in the swarm
- P_k^i - Personal best position of the i^{th} particle after the k^{th} time step
- g_k - Global best position of all particles at k^{th} time step.
- $Population_{k+1} = Population + Vel_{k+1}$
- $Vel_{k+1} = \omega * vel_k + c_1 * r_1 * (P_k^i - Pop) + c_2 * r_2 * (g_k - Pop)$

PSO - Encoding Details

- Stope fraction = $\frac{No.\ of\ stopes\ used}{Total\ number\ of\ allowed\ stopes}$.
- Stope fraction was fixed - binary decision matrices were initiated.
- 1000 population members - 1000 matrices
- Each member of the population was a set of co-ordinates representing the ones in their respective matrix.
- $Fitness = Config\ Value - k * (Overlap\ Penalty) - k * (Level\ Penalty)$

PSO - But does it work? - Yes

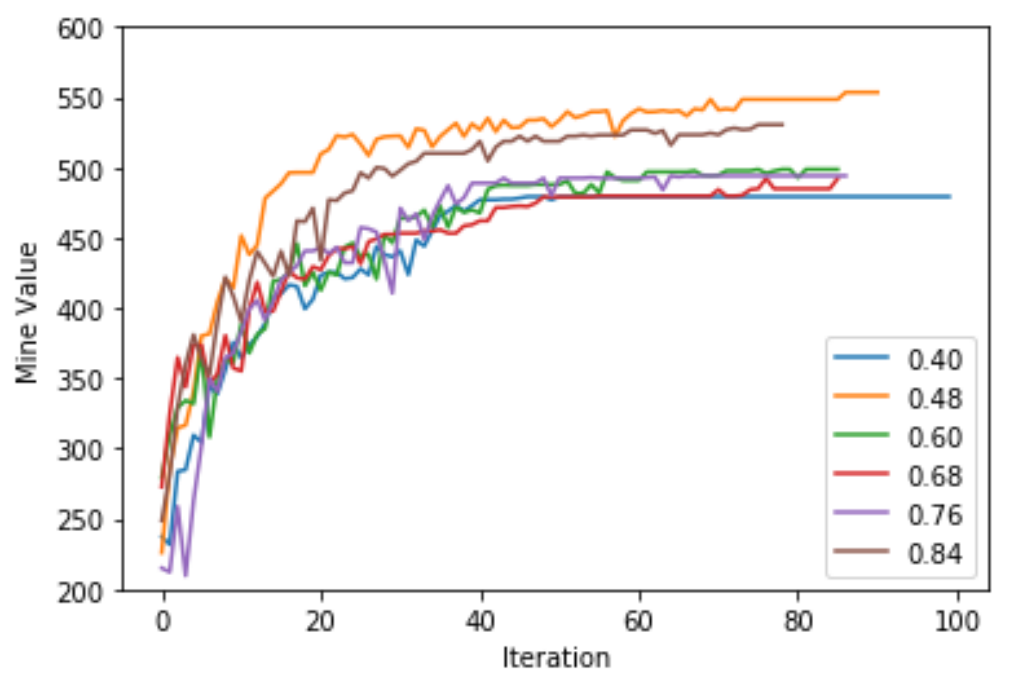
-20	-20	2	7	9	-1	4	-5	7
9	6	3	2	3	-13	23	56	-21
1	-2	4	5	7	-40	0	-11	51
3	0	5	4	-60	30	14	1	31
-4	-7	6	3	-10	4	12	14	34
3	0	7	2	-23	2	4	22	-15
-4	-7	8	1	12	12	3	1	11
5	2	6	2	3	0	1	-50	-20
11	8	0	-1	17	-42	11	0	17
17	0	3	4	23	-2	2	65	23
-3	-6	2	-60	18	11	1	54	12
8	5	6	3	1	0	3	3	2
10	7	-7	-2	18	4	14	9	-57
-8	-11	0	-50	12	18	-45	21	7

PSO - Problem Specification

- **Constant** stope size was considered (2 x 2)
- Various stope fractions were considered.
- For each stope fraction, five experiments were run, with 100 iterations per experiment.
- For each iteration, the maximum fitness value of that population is taken and stored.
- After each experiment, the mean maximum population values were considered, for every iteration.
- These results are presented for various stope fractions.

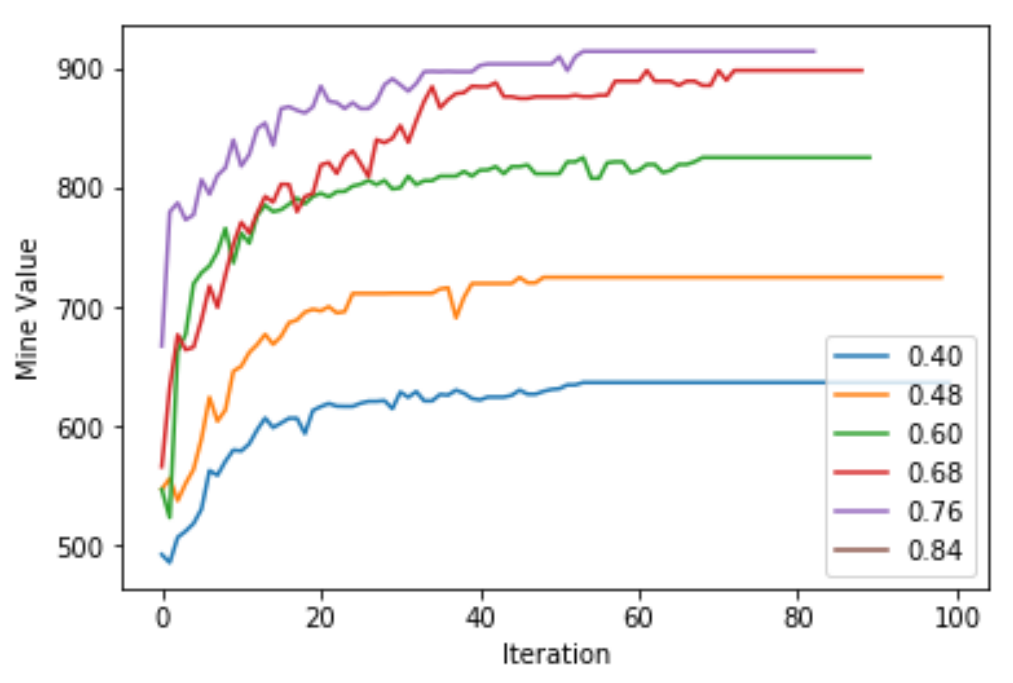
PSO - Results Mine 1

- Different configuration values for different stope fractions.
- Upon convergence, 0.4→worst, 0.48→best.
- Is there a relationship between stope fraction and mine value? Consider mine 2...



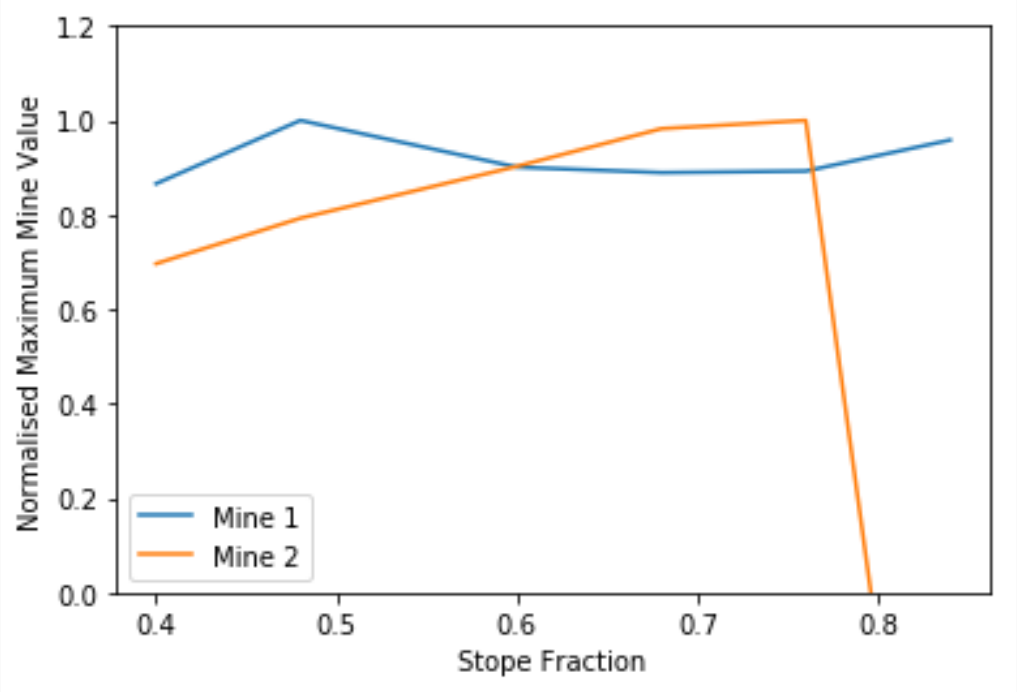
PSO - Results Mine 2

- Upon convergence, 0.84→worst, 0.76→best.
- No clear relationship - maybe some intrinsic properties of each mine that dictate optimal stope fraction (BEV distribution?)
- Looking at the data slightly differently...



PSO - Results Stope Fraction

- Variation of maximum values as a function of mass fraction
- Since these curves are completely uncorrelated, the BEV distribution per mine plays a large role in stope configuration design.



Hybrid Algorithm

The pseudo-code for the hybrid algorithm which we have developed is as follows:

- Input: Mining site dimensions, array containing mining data with BEVs for each block, stope dimensions.
- STEP 1: For *i* = 1 to *swarmSize*
- *dpSolution* = DPH() swarm.add(*dpSolution*)
- STEP 2: PSO(*swarm*)
- Output PSO.gbest as best solution
- Output *swarm* as set of alternative solutions.

Conclusion

In conclusion, we have been able to successfully achieve the following:

- Develop Mathematical model for the 2D SBOP
- Develop Mathematical model for the 3D SBOP
- Develop and test DP-esque heuristic, Multi-Start, and PSO

What we haven't been able to do (for shortness of time):

- Extend solution methods to 3D case
- Implement hybrid DP-esque and PSO algorithm