

Stope Optimiser - A FORTRAN Program to Optimise Stope Boundaries

E.Y.Baafi & M.Ataee-pour

Faculty of Engineering, University of Wollongong, Wollongong, NSW 2522, Australia

ABSTRACT: A computer program has been developed to implement a heuristic algorithm for optimisation of underground stope geometries. The *Stope Limit Optimiser (SLO)* software integrates *FORTRAN 90* computer program with *WINTERACTER* user interface features to provide a Windows based interactive environment to define and modify the underground mine design parameters such as, block model parameters, stope geometry constraints and economic factors. The heuristic algorithm termed *Maximum Value Neighbourhood (MVN)* and its *FORTRAN 90* implementation procedure are presented with examples.

1 PROBLEM DEFINITION

The problem of finding the best combination of desirable and non-desirable blocks that result in the maximisation of profit of a stope, may be expressed by Equation (1).

$$\left. \begin{array}{l} \text{Objective function:} \\ \text{Maximise } SEV = \sum_{ijk} F_{ijk} BEV_{ijk} \\ \text{subject to:} \\ \text{stope geometry constraints} \end{array} \right\} \quad (1)$$

where

SEV : total stope economic value,

BEV_{ijk} : the economic value of block B, j, k ,

F_{ijk} : an indicator function showing whether the block, B, j, k , is mined or not and defined by Equation (2).

$$F_{ijk} = \begin{cases} 1 & \text{if } B_{ijk} \text{ is selected,} \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

2 STOPE GEOMETRY

The geometry of the working spaces in underground mines is restricted in each orthogonal direction by both a minimum and a maximum size (length, width or height). Figure 1 depicts the minimum stope size constraints in 1D, 2D and 3D directions. Physical parameters, such as the geo-mechanical properties of the ore-body and the surrounding rock, the dip, the depth and the thickness of the ore-body, affect the proposed underground mining methods, which, in turn, impose some practical restrictions on the extraction limit of the orebody. The *block caving* method imposes different constraints to the stope geometry from that imposed by a *cut-and-fill* method. A *cut-and-fill* stoping is flexible allowing the extraction of the high-grade ore whilst leaving the low-grade material in the stope as fill. In practice, the minimum size of the stope must be designed so that sufficient space is provided for the activities of drilling, blasting and loading, as well as the traffic of personnel. The maximum limits of the stope dimensions are usually dictated by the geo-technical factors.

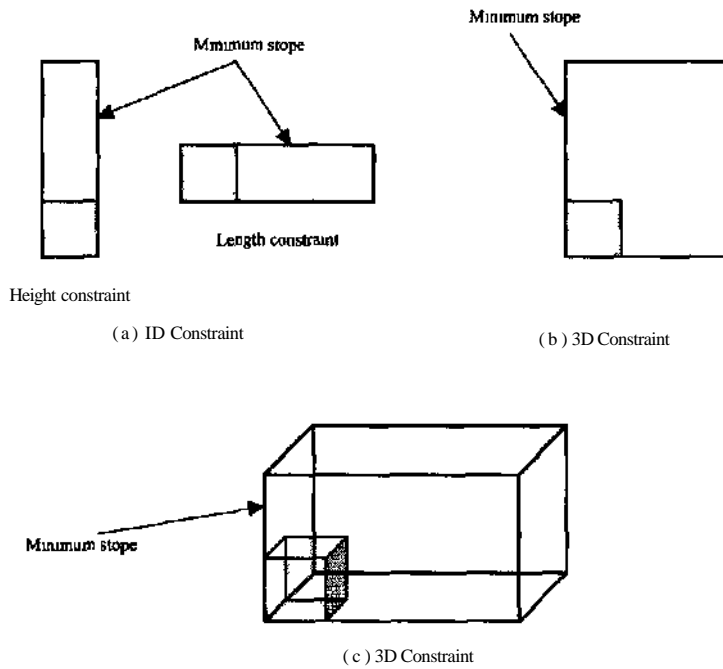


Figure \ Slope Constraints (a) 1D, (b) 2D and (c) 3D Problems

3 THE NEIGHBOURHOOD CONCEPT

The minimum slope size represents by the number of mmeable adjacent blocks of a fixed geological block model. This leads to the definition of the two terms the "neighbourhood" and the "order of neighbourhood". When considering a one-dimensional stope constraint, a "neighbourhood" (NB) is the set of all the sequential blocks, including the block of interest in the specified direction, which may be mined to satisfy the minimum slope size requirement. Figure 2 shows a row of seven blocks (B_j , $j = 1, 2, \dots, 7$). If the mininram slope size is 20 m and the block size of a fixed block model is 5

m, then a possible *neighbourhood* for the block, B_4 , may consist of the set of four sequential blocks $\{B_i, B_4, B_i, B_6\}$. It is clear that the illustrated neighbourhood is not the only possible neighbourhood for the block, B_4 . One can easily assume that the set $\{B_3, B_i, B_4, B_5\}$ could be another possible neighbourhood for the same block, B_4 . Assummg that the stope geometry is restricted in length or height only, then the term "stope block fig/io" *fSBR*) may be expressed by Equation (3).

$$fSBR = \frac{\text{minimum slope size (length or height)}}{\text{fixed block size (length or height)}} \quad (3)$$

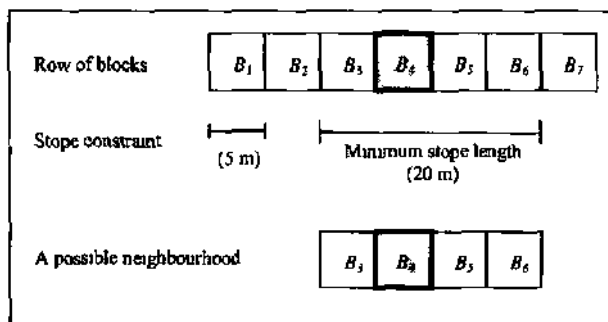


Figure 2 Minimum Slope Size versus the Neighbourhood (NB)

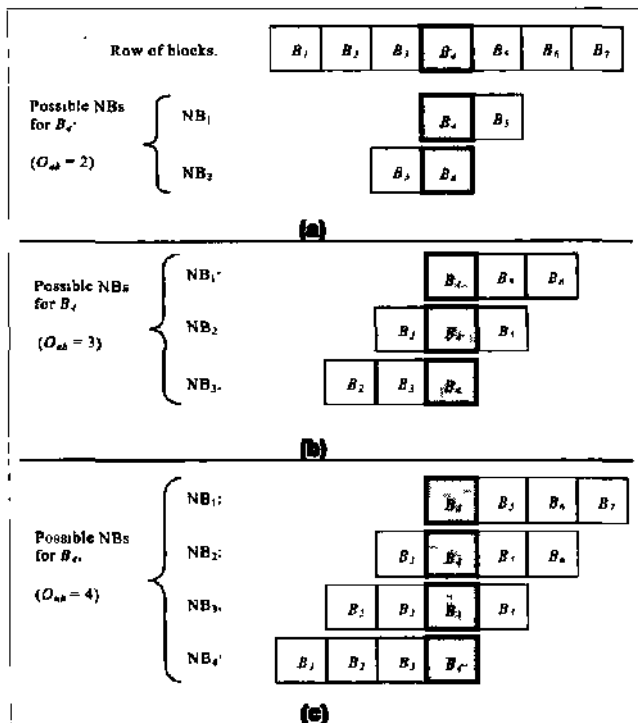


Figure 3. Possible Neighbourhoods of the Block, B_j , for MJ Orders of 2,3 and 4.

A slope block ratio of 3.8 suggests that the minimum slope length (height) must be 3.8 times the block size of the fixed block model. The slope block ratio is a continuous real function, which suggests an inclusion of partial blocks. Since the proposed *MVN* algorithm does not allow partial blocks, the value of *SBR* is rounded up. A positive integer value termed the "order of neighbourhood" (O_{nb}) is used to represent the rounding up value of the slope block ratio. In other words, the *order of neighbourhood* represents the size of the neighbourhood set. That is, the total number of sequential blocks, including the block of interest, in a specified direction, which may be mined to satisfy the minimum slope size requirement. There are a number of combinations of blocks forming a neighbourhood for any given block. Figure 3 shows the possible neighbourhoods of the block, B_4 , for the neighbourhood orders of 2, 3 and 4. For the *NB* order of 4, four neighbourhoods may be defined, so that each neighbourhood contains four members (Figure 3c).

The set of neighbourhoods, $NB_{m,j}$, for any block, B_j , with any order of neighbourhood, m , may be defined by the set of m sequential blocks in ascending order, starting from the block, B_{j-m+1} , and ending with the block, B_{j+m} , where $1 \leq m \leq j$, as

expressed by Equation (4):

$$NB_{m,j} = \{BEV_{\beta} \in U \mid \beta = j-m+1, j-m+2, \dots, j+m\} \quad (4)$$

where

- β : the column number, β , used for blocks in the m^{th} neighbourhood and
- BEV_{β} : the economic value of a typical block, located in the m^{th} neighbourhood.

The definition of any neighbourhood requires'

- (i) the location of the block, for which the neighbourhood is defined (the j address of the block in a ID constraint problem),
- (ii) the order of neighbourhood, " O_{nb} " and
- (iii) the neighbourhood number (that is, which neighbourhood is required).

In general, each neighbourhood, $NB_{m,j}$, is assigned a value, denoted by $NBV_{m,j}$ which is defined by Equation (5)

$$NBV_{m,j} = \sum_{\beta=j-m+1}^{j+m} BEV_{\beta} \quad (5)$$

For any block, B_j , with the neighbourhood order of m , the maximum neighbourhood value

(*MNBV*), among all the elements of the set of neighbourhood values, is defined by:

$$MNBV_{j,t} = \text{Max. } \{NBV_{1,j,t}, NBV_{2,j,t}, \dots, NBV_{m,j,t}, \dots, NBV_{l,j,t}\}$$

or

$$MNBV_{j,t} = \text{Max } NBVS_{j,t} \\ = \text{Max } \{NBV_{m,j,t} \mid m = 1, 2, \dots, l\}$$

As a numerical example, consider the row of blocks shown in Figure 4. If the minimum slope length is ten meters and each block is three meters long, the slope block ratio is 3.3, which results in an order of neighbourhood of 4. All neighbourhood values for the block, B_j , with the order of 4 are collected in the set $NBVS_{j,t}$. Then, the 4th neighbourhood value (*NBV*) provides the maximum neighbourhood value, termed *MNBV* for this example.

4 MAXIMUM VALUE NEIGHBOURHOOD ALGORITHM

The neighbourhood concept locates the optimum neighbourhood for a block. This heuristic optimisation technique requires the following seven basic steps for each block within the economic model, with a value *BEY*:

- (i) determine the neighbourhoods of the block, based on the order of neighbourhood, i.e. construct the set of possible neighbourhoods (*NBS*);
- (ii) evaluate the feasibility of each neighbourhood within the *NBS* set;
- (iii) calculate the economic value for each neighbourhood, i.e. neighbourhood value *NBV*, and determine the set of neighbourhood values (*NBVS*);
- (iv) locate the maximum neighbourhood value (*MNBV*) within the *NBVS* set;
- (v) determine the maximum value neighbourhood (*MVN*);
- (vi) flag the blocks of the *MVN* and
- (vii) update the slope economic value (*SEV*).

The generalised flow-chart for the optimisation procedure in the *MVN* algorithm is illustrated in Figure 5 whereby blocks are taken into consideration one by one, in the order of rows (X direction), columns (Y direction) and finally sections (Z direction). FORTRAN 90 implementation of *MVN* algorithm has been provided by Ataee-pour (2000). If a block has a non-negative value and is not already flagged, the procedure will continue to construct its set of neighbourhoods, (*NBS*), based on the order of neighbourhood "1"; calculate the values of neighbourhoods and finally locate the maximum neighbourhood value (*MNBV*) as well as the maximum value neighbourhood (*MVN*) of the block.

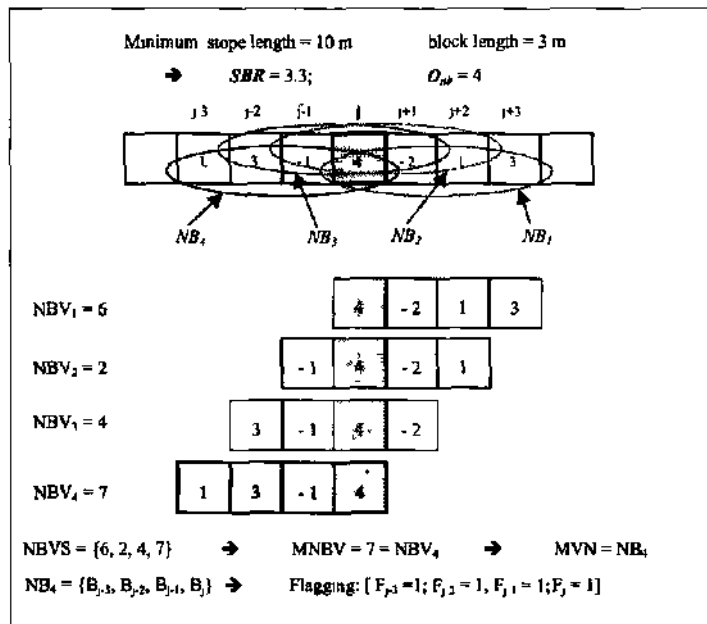


Figure 4. Locating the Maximum Value Neighbourhood.

The algorithm then determines whether or not the maximum neighbourhood value is non-negative, i.e. it contributes to the stope value. Ignoring the blocks with a negative *MNBV*, the procedure continues for the non-negative ones. The *MVN* of each block represents its marginal contribution to the stope value. This marginal value may be negative, zero or positive. Negative marginal values are ignored and the algorithm selects only a block with an *MVN* that provides a non-negative marginal value. The elements of that *MVN* are then flagged and included in the stope. The above process is repeated for all

blocks with the block economic model. After all the blocks are examined, the final optimum stope boundaries are displayed. Four checks are used in this algorithm to exclude unnecessary blocks. A block is ignored:

- (1) if the block is flagged already;
- (2) if the block has a negative dollar value;
- (3) if the maximum neighbourhood value (*MNBV*) of the block is negative; and
- (4) if the marginal value provided by the *MVN* is negative

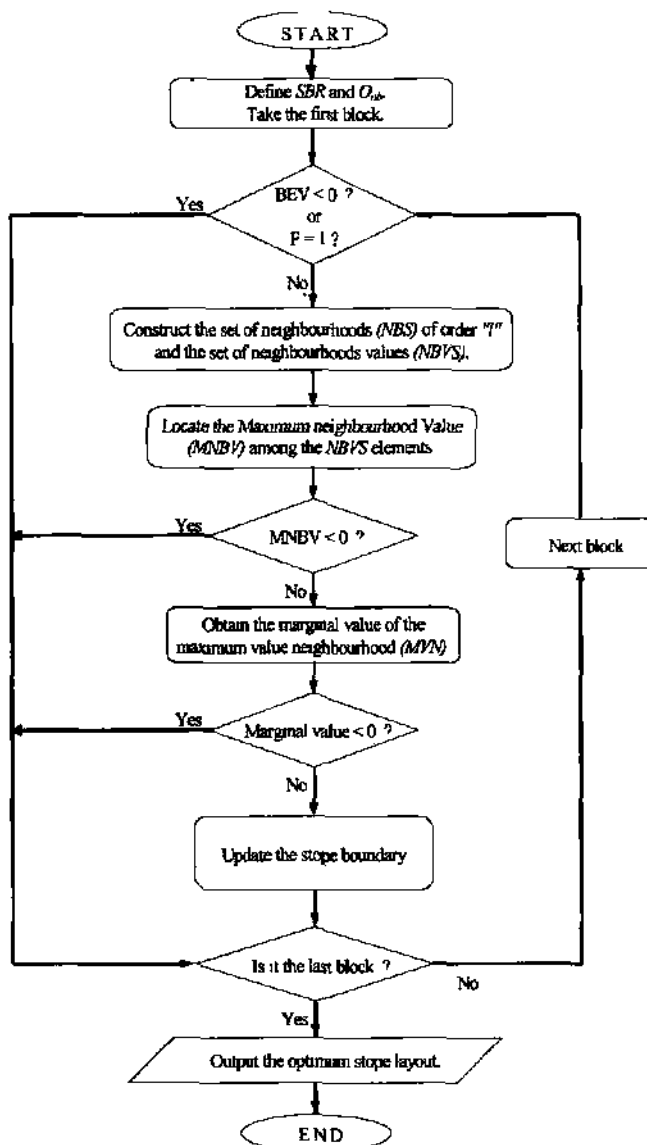


Figure 5. Generalised Flow-Chart for the Optimisation Procedure

5 NUMERICAL EXAMPLE

Consider a row of 10 m blocks, $\{B_j, j = 1, 2, \dots, 10\}$, shown in Figure 6. The number inside each cell represents the economic value, BEV_j , of the

respective block. A 25 m minimum slope length gives a slope block ratio is then 2.5, which results in the order of neighbourhood of 3. The pseudocode of the *MVN* algorithm with this example is provided below:

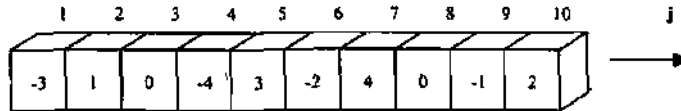


Figure 6 A Row of Blocks with 10 Columns taken from an Economic Block Model

Step 1: Initialise variables.

Block length = 10 m; Minimum slope length = 25

SBR=2.5; $O_{n|>} = 3$

SEV = 0; F = 0; FF = 0

Step 2: Examine the first block.

B_1 : $BEV_1 = -3$; $F_1 = 0$

Step 3: Check the negativity and the flag of the block.

$BEV_1 < 0 \rightarrow$ Block B_1 is exempted from further process.
Go to Step 9.

Step 9: Check the end of the model.

B_1 is not the last block. Continue.

Step 10: Examine the next block.

B_2 : $BEV_2 = 1$; $F_2 = 0$

Go to Step 3

Step 3: Check the negativity and the flag of the block.

$BEV_2 > 0$ and $F_2 = 0 \rightarrow$ Continue.

Step 4: Locate the maximum neighbourhood value (*MNBV*)

$NB_1 = \{1, 0, -4\} \rightarrow NBV_1 = 1 + 0 - 4 = -3$

$NB_2 = \{-3, 1, 0\} \rightarrow NBV_2 = -3 + 1 + 0 = -2$

$NB_3 =$ Not feasible

$\rightarrow MNBV = -2$; $MVN = NB_2$

Step 5: Check the negativity of the *MNBV*.

$MNBV < 0 \rightarrow$ Block B_2 is exempted from further process.
Go to Step 9.

Step 9: Check the end of the model.

B_2 is not the last block. \rightarrow Continue.

Step 10: Examine the next block.

B_3 : $BEV_3 = 0$; $F_3 = 0$

Go to Step 3

Step 3: Check the negativity and the flag of the block.

$BEV_3 \neq 0$ and $F_3 = 0 \rightarrow$ Continue.

Step 4: Locate the maximum neighbourhood value (*MNBV*).

$NB_1 = \{0, -4, 3\} \rightarrow NBV_1 = 0 - 4 + 3 = -1$

$NB_2 = \{1, 0, -4\} \rightarrow NBV_2 = 1 + 0 - 4 = -3$

$NB_3 = \{-3, 1, 0\} \rightarrow NBV_3 = -3 + 1 + 0 = -2$

$\rightarrow MNBV = -1$; $MVN = NB_1$

Step 5: Check the negativity of the *MNBV*.

$MNBV < 0 \rightarrow$ Block B_3 is exempted from further process.
Go to Step 9.

Step 9: Check the end of the model.

B3 is not the last block → Continue

Step 10: Examine the next block

B₄: BEV₄ = -4; F₄ = 0

Go to Step 3

Step 3: Check the negativity and the flag of the block.

BEV₄ < 0 → Block B₄ is exempted from further process.
Go to Step 9

Step 9: Check the end of the model.

B4 is not the last block. T > 0 → Continue.

Step 10: Examine the next block.

B₅: BEV₅ = 3; F₅ = 0

Go to Step 3

Step 3: Check the negativity and the flag of the block

BEV₅ ≥ 0 and F₅ ≠ 1 → Continue.

Step 4: Locate the maximum neighbourhood value (MNBV).

NB₁ = (3, -2, 4) •• NBV₁ = 3 - 2 + 4 = 5

NB₂ = (H, 3, -2) → NBV₂ = -4 + 3 - 2 = -3

NB₃ = (0, -4, 3) •* NBV₃ = 0 - 4 + 3 = -1

* MNBV = 5; MVN = NB₁

Step 5: Check the negativity of the MNBV.

MNBV > 0 → Continue.

Step 6: Calculate the marginal value.

MVN = NB₁ = {BEV₁, BEV₆, BEV₇}

F₅ = 0 → FF₅ = 1; F₆ = 0 → FF₆ = 1; F₇ = 0 → FF₇ = 1

Marginal Value = BEV₅ · FF₅ + BEV₆ · FF₆ + BEV₇ · FF₇
= (3 × 1) + (-2 × 1) + (4 × 1) = 5

Step 7: Check the negativity of the marginal value.

Marginal Value > 0 → Continue.

Step 8: Update the slope.

SEV = SEV + Marginal Value = 0 + 5 = 5 → SEV = 5

F₃ = 1; F₆ = 1; F₇ = 1

Step 9: Check the end of the model.

B₅ is not the last block. → Continue.

Step 10: Examine the next block.

B₆: BEV₆ = -2; F₆ = 1

Go to Step 3

Step 3: Check the negativity and the flag of the block.

BEV₆ < 0 → Block B₆ is exempted from further process.
Go to Step 9.

Step 9: Check the end of the model.

B₆ is not the last block. → Continue

Step 10: Examine the next block.

B₇: BEV₇ = 4; F₇ = 1

Go to Step 3

Step 3: Check the negativity and the flag of the block.

F₇ = 1 → Block B₇ is exempted from further process*
Go to Step 9.

Step 9: Check the end of the model.

B₇ is not the last block. → Continue.

- Step 10: Examine the next block.
 B_7 : $BEV_7 = 0$, $F_7 = 0$
 Go to Step 3
- Step 3: Check the negativity and the flag of the block.
 $BEV_7 > 0$ and $F_7 \neq 1 \rightarrow$ Continue.
- Step 4: Locate the maximum neighbourhood value (*MNBV*).
 $NB_1 = \{0, -1, 2\} \rightarrow NBV_1 = 0 - 1 + 2 = 1$
 $NB_2 = \{4, 0, -1\} \rightarrow NBV_2 = 4 + 0 - 1 = 3$
 $NB_3 = \{-2, 4, 0\} \rightarrow NBV_3 = -2 + 4 + 0 = 2$
 $\rightarrow MNBV = 3$; $MVN = NB_2$
- Step 5: Check the negativity of the *MNBV*.
 $MNBV > 0 \rightarrow$ Continue.
- Step 6: Calculate the marginal value.
 $MVN = NB_2 = \{BEV_7, BEV_8, BEV_9\}$
 $F_7 = 1 \rightarrow FF_7 = 0$; $F_8 = 0 \rightarrow FF_8 = 1$; $F_9 = 0 \rightarrow FF_9 = 1$
 Marginal Value $= BEV_7 \cdot FF_7 + BEV_8 \cdot FF_8 + BEV_9 \cdot FF_9$
 $= (4 \times 0) + (0 \times 1) + (-1 \times 1) = -1$
- Step 7: Check the negativity of the marginal value.
 Marginal Value $< 0 \rightarrow$ Block B_7 is exempted from further process.
 Go to Step 9.
- Step 9: Check the end of the model.
 B_7 is not the last block. \rightarrow Continue.
- Step 10: Examine the next block.
 B_9 : $BEV_9 = -1$; $F_9 = 0$
 Go to Step 3
- Step 3: Check the negativity and the flag of the block.
 $BEV_9 < 0 \rightarrow$ Block B_9 is exempted *Ètm fàrtfeerpöeess*.
 Go to Step 9.
- Step 9: Check the end of the model
 B_9 is not die last block. \rightarrow Continue.
- Step 10: Examine the next block.
 B_{10} : $BEV_{10} = 2$; $F_{10} = 0$
 Go to Step 3.
- Step 3: Check the negativity and die flag of die block.
 $BEV_{10} > 0$ and $F_{10} \neq 1 \rightarrow$ Continue.
- Step 4: Locate tile maximum neighbourhood value (*MNBV*).
 $NB_1 =$ Not feasible
 $NB_2 =$ Not feasible
 $NB_3 = \{0, -1, 2\} \rightarrow NBV_3 = 0 - 1 + 2 = 1$
 $\rightarrow MNBV = 1$; $MVN = NB_3$
- Step 5: Check the negativity of the *MNBV*.
 $MNBV > 0 \rightarrow$ Continue.
- Step 6: Calculate the marginal value.
 $MVN - NB_3 = \{BEV_8, BEV_9, BEV_{10}\}$
 $F_8 = 1 \rightarrow FF_8 = 1$; $F_9 = 0 \rightarrow FF_9 = 1$; $F_{10} = 0 \rightarrow FF_{10} = 1$
 Marginal Value $= BEV_8 \cdot FF_8 + BEV_9 \cdot FF_9 + BEV_{10} \cdot FF_{10}$
 $= (0 \times 1) + (-1 \times 1) + (2 \times 1) = 1$
- Step 7: Check the negativity of the marginal value.
 Marginal Value $\geq 0 \rightarrow$ Continue.
- Step 8: Update the stoep.

$$SEV = SEV + \text{Marginal Value} = 5 + 1 = 6 \quad \rightarrow \quad SEV = 6$$

$$F_a = 1; \quad F_c = 1; \quad F_{s_0} = 1$$

Step 9: Check the end of the model.

Bio is the last block. \rightarrow Go to Step 11.

Step 11: Output the results.

**Stop»» (B* Bs, Bfc Bs, Bs*»815)î Stope Value-6
STOP**

END

Figure 7 shows the optimised row of blocks, in which the flagged blocks are shaded. The first four blocks, including one positive and one zero value block, are excluded from the final stope as they do not improve the stope economic value. In addition, two negative blocks have been included in the optimum stope to insure that the minimum stope size constraints are met.

The following example (Figure 8a) shows a section of six rows with six columns of blocks, that has been optimised using the algorithm. In this example, the order of neighbourhood equals 3 in the vertical direction, that is, the stope has a minimum height restriction of three blocks. The optimised section has been shown in Figure 8b, where the flagged blocks have been shaded. Table 1 summarises the step-by-step application of the *MVN* algorithm to the blocks in the third column of Figure 8a.

6 CONCLUDING REMARK

The "maximum value neighbourhood" concept uses a fixed economic block model to outline the optimum geometry of a stope. After expressing the optimisation objective function as maximization of net stope economic value, the imposed stope geometry constraints were formulated in terms of "neighbourhood" and "order of neighbourhood". For an individual block, the *MVN* approach outlines a small island of mineable blocks within the whole ore-body.

REFERENCE

Ataee-pour, M 2000 A Heuristic Algorithm to Optimise Stope Boundaries, PhD Thesis, University of Wollongong, 313 pp

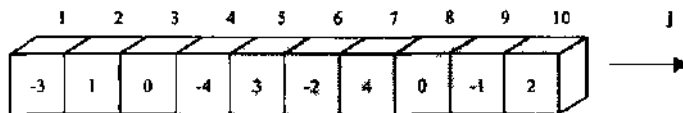


Figure 7. The Optimised Example Manually using the *MVN* Algorithm.

3	-2	3	-2	1	3
4	-1	2	4	3	5
-2	5	4	5	-1	3
3	-2	3	-2	6	-2
4	1	3	2	3	2
2	-3	-2	5	-2	3

(a) Model Section

3	-2	3	*2	1	3
4	-1	2	4	3	5
-2	5	«	5	-1	3
3	-2	3~	-2	6	-2
4	1	3	2	3	2
2	-3	-2	5	-2	3

(b) Optimum Stope Section

Figure 8. Optimising a Stope Section using the *MVN* Algorithm with a 1D (height) Constraint.

Table 1 Summary of the Algorithm Applied for Column 3 of Figure 8

block (i, j)	BE V	negative?	flag?	Neighbourhoods			old F	F F	BEV	FF BEV	old SEV	new SEV	new F	
				1	2	3								
(1, 3)	3	no	no	3 2 6			$F_{13}=0$ $F_{23}=0$ $F_{33}=0$	1 1 1	3 2 6	3 2 6		0	11	$F_{13}=1$ $F_{23}=1$ $F_{33}=1$
(2, 3)	2	no	yes											
(3, 3)	6	no	yes											
(4, 3)	3	no	no	3 3 -2	6 3 3	2 6 3	$F_{33}=1$ $F_{43}=0$ $F_{53}=0$	0 1 1	6 3 3	0 3 3		11	17	$F_{43}=1$ $F_{53}=1$
(5, 3)	3	no	yes											
(6, 3)	-2	yes												