Application of the Push-Relabel Algorithm to Lignite Surface Mine Optimisation

Panagiotis Mpinos[1], Ioannis Kapageridis[1]

[1]Technological Educational Institute of Western Macedonia, Koila, Kozani, 52100, Greece

ABSTRACT

Open pit optimisation is a process aiming at the determination of the extents of the optimum pit such that the profit made in mining the pit is maximised. The choice of blocks to mine for the optimum pit is an example of the selection problem. A selection problem is based on a set of tasks, where each task has a value or a cost. In most cases, there are certain relationships between tasks, such that in order to perform one task a number of prerequisite tasks must be performed. The solution of the selection problem is the subset of tasks that the sum of their value is the maximum possible within the set when performed. In terms of mining, each block (a task in the selection problem) in a 3D block model is assigned either a profit or a loss based on the revenues and costs associated with mining the block. Geologic constraints are used to establish slope requirements for each block which are used to determine the blocks which must be removed prior to the removal of any given block. Traditional methods of solving the selection problem in open pit optimisation included the floating cone algorithm and the Lerchs-Grossman algorithm based on graph theory. The latter dominated open pit optimisation software products and solutions in the 80s and 90s and offered mining engineers a solid solution to the pit optimisation problem. A decade later from the first implementation of the Lerchs-Grossman algorithm, Picard proved that the pit optimisation problem could be solved with more efficient maximum flow algorithms. In 1988 Goldberg and Tarjan published the first paper describing the Push-Relabel algorithm for solving the maximum flow problem. Later in 1997, Cherkassky and Goldberg published a paper describing a very efficient implementation of the more general Push-Relabel algorithm. This algorithm is used in our case study of surface lignite mine optimisation. A lignite deposit from the region of Kozani as well as all associated technical and financial parameters are used as input to the Push-Relabel implementation provided by a mine planning software package, and the optimisation output is analysed in order to assess the benefits of applying the Push-Relabel algorithm to lignite deposits.

1 INTRODUCTION

Open pit optimisation is a process commonly applied in mine planning of surface mines to produce optimum pit limits to use as a guide for pit design. The optimisation step is also considered an efficient way to convert mineral resources to mineral reserves as it allows the enforcing of financial and technical constraints and parameters to the mine design process in an automated and mathematically robust way. It is commonly used even at the mineral resources estimation stage to limit the reported quantities inside a conceptual pit and raise the confidence in the mineral resources report.

Surface coal and lignite mines have been commonly modelled in the past using a more two-dimensional approach, based on grid or triangulation models that did not allow the application of open pit optimisation algorithms, normally requiring a three-dimensional blocks model of the deposit. The financial aspects of coal deposits are also considered stable along the Z axis, in most cases where the deposit consists of a small number of coal horizons with standard qualities, leading to the conception that pit optimisation is an unnecessary effort. The lignite deposits in Greece, however, normally consist of multiple lignite layers with varying quality parameters in all three dimensions, making them ideal targets for computerised open pit optimisation.

The case study presented in this paper discusses the application of the Push-Relabel method, one of the more recent optimisation methods, and provides a comparison with the well-established Lerchs-Grossman method which is used by the mining industry the last three decades.

2 HISTORY OF OPEN PIT OPTIMISATION

2.1 Before Computers

Before computers found their way into mine planning, mining engineers relied on manual methods on hand-drawn cross-sections to produce a pit design. A simple optimisation of economic pit depth was usually performed with the aid of a calculator for regular shaped orebodies using incremental cross-sectional areas, for ore and waste, and an overall pit slope. Incremental stripping ratios were calculated and compared against the break-even stripping ratio. The final pit shell was then produced by drawing increasingly larger pit shells on cross section such that the last increment had a strip ratio equal to the design maximum. This was a very labour-intensive approach and could only ever approximate the optimal pit [1].

2.2 Floating Cone Method

The Floating Cone algorithm was introduced by Pana (1965) [2]. The method was developed at Kennecott Copper Corporation during the early 1960s and was the first computerised attempt at pit optimization, based on a three-dimensional block model of the mineral deposit. Final pit limits are developed by using a technique of a moving “cone” (or rather an inverted cone). The cone is moved around in the block model space from top to bottom generating a series of interlocking cone-shaped openings. The disadvantage of this approach is that it creates overlapping cones, and it is incapable of examining all combinations of adjacent blocks. For this reason, the algorithm fails to consistently give realistic results and tends to “mine” more tonnage for less value.

2.3 Lerchs-Grossman Method

The same year the floating cone algorithm was introduced (1965), Lerchs and Grossmann published a paper that introduced two modelling approaches to solving the open pit optimisation problem [3]. The Lerchs-Grossman (LG) algorithm is well documented in the technical literature [4, 5, 6, 7]. Lerchs and Grossmann presented two implementations of the pit optimisation algorithm, the first based on Graph Theory (heuristics) and the second on Dynamic Programming (operations research). They both produced optimum pit limits based on an undiscounted cash flow – an economic block model including both ore and waste. Essentially the methods determined which blocks should be mined to obtain the maximum value from the pit. LG requires a technical and a financial parameter:

1. ***Pit slopes:*** these define the blocks that need to be removed before each block considered in the block model. They are used to generate “arcs” between blocks.
2. ***Block value:*** refers to the economic value of each uncovered block. It will be negative for waste blocks and amount to all waste mining and hauling costs. Ore blocks will have values

based on the mining, hauling, processing, selling and any other costs, and the revenue from the recovered ore.

Working from the lowest positive block(s) and using the block values and structure arcs, the method branches upwards between blocks forming a graph (Figure 1). Branches are flagged based on their total value. Positive branches are worth mining once uncovered. Negative branches are also flagged, and the method looks for positive ones that lie below them. In this case, the two branches are combined in a way to produce a positive total branch. The scanning is repeated until no structure arc goes from a positive branch to a negative. Once this is complete, the complete graph defines the optimum pit. Any negative branches left on their own are not to be mined.

In mathematical terms, the LG algorithm finds the maximum closure of a weighted directed graph [1]. The blocks in the model represent the vertices of the graph, the block values represent the weights, and the mining constraints (i.e. the pit slopes) represent the arcs. The LG algorithm provides a mathematically optimum solution to the problem of pit optimisation. The algorithm itself has no "sense" of the nature of the optimisation problem – it works on a set of vertices and arcs. Whether these are defined in one, two or three dimensions and the number of arcs per block makes no difference to the algorithm. The LG algorithm has been used for over 30 years on many feasibility studies and for many producing mines.

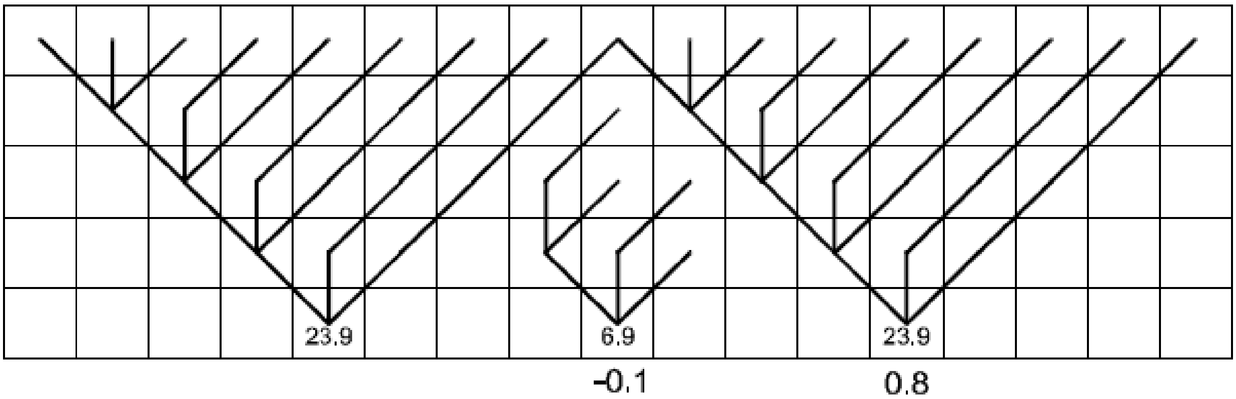

Figure 1. Example of LG optimisation showing three positive blocks surrounded by negative blocks (with a value of -1) linked with branches forming a final graph. The negative branch in the middle is not to be mined [8].

2.4 Network Flow Methods and the Push-Relabel Method

Lerchs and Grossmann suggested that the ultimate-pit problem could be expressed as a maximum closure network flow problem and presented their approach - a method of solving a special case of a network flow problem. Picard proved that a maximum closure network flow problem (like the open pit optimisation) could be reduced to a minimum cut network flow problem which could be solved by an efficient maximum flow algorithm [9]. This meant that network flow algorithms could be used instead of the LG algorithm, and they can calculate identical results in a fraction of the time.

The Push-Relabel algorithm considered in this paper is one of the first efficient maximum flow algorithms used in solving the open pit optimisation problem [10, 11, 12]. It has been shown that the Push-Relabel algorithm outperformed the LG algorithm in nearly all cases [13]. In cases where the number of vertices (blocks in the pit optimisation problem) is greater than a million, network flow algorithms perform orders of magnitude faster and compute precisely the same results [1]. The pit optimisation module of Maptek Vulcan mine planning software is based on implementations of both the LG and Push-Relabel algorithms.

3 THE PUSH-RELABEL METHOD

3.1 Historical Background

The maximum flow problem is a classical combinatorial problem that arises in a wide variety of applications. The basic methods for the maximum flow problem include the network simplex method of Dantzig [14], [15], the augmenting path method of Ford and Fulkerson [16], the blocking flow method of Dinitz [17], and the push-relabel method of Goldberg and Tarjan [10], [18]. Prior to the push-relabel method, several studies have shown that Dinitz's algorithm [17] is in practice superior to other methods, including the network simplex method [14], [15], Ford-Fulkerson algorithm [16], Karazanov's algorithm [19], and Tarjan's algorithm [20]. Several recent studies ([21], [22], [23] and [24]) show that the push-relabel method is superior to Dinitz's method in practice [25].

3.2 The Push-Relabel Method

The definition of a pit with valid slopes is termed a "closed set" or "closure". It consists of a set of nodes *V* that have no arcs initially. Based on the required pit slopes, a set of arcs *E* is defined representing the dependencies between blocks. A closed set of blocks is free to be removed and does not depend on the removal of other blocks. Finding an optimal pit is the process of finding a closure with maximum total value [26]. This problem is called a maximum closure problem. It is easy to observe in Figure 2, that the optimal pit consists of block {b, c, f, g, h, i}, with a total value of 3.

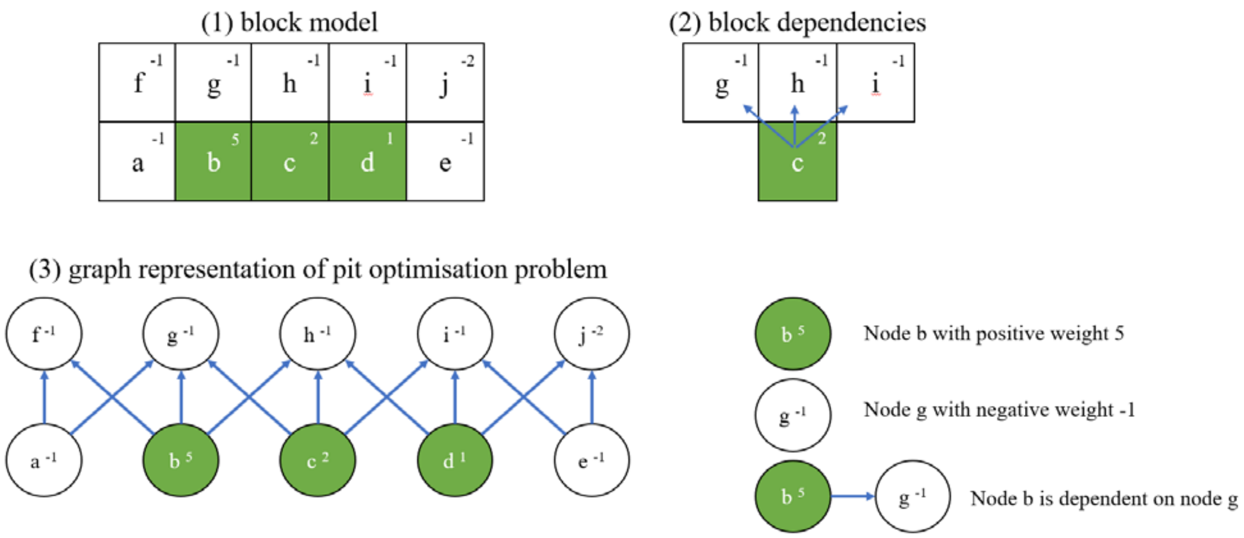

Figure 2. Simple example of block model (1), block dependencies (2) and graph representation (3) for a pit optimisation problem [26].

Two additional special nodes are required: the flow starts from the *source* node and finishes at the *sink* node. Each arc is like a pipe and has a nonnegative capacity function *u* allowing flow up to a limit passing through it. The flow and capacity along an arc must be positive. The nodes (blocks) represent a joining of pipes, so the amount of flow into a node must equal the total flow out of the node, which is called the *conservation constraint*. Each node (block) has a weight value equal to the economic value of the block. Defining a complete flow graph means that we need to make the following changes to the graph of the block model in Figure 2:

- Add two special (virtual) nodes: source *s* and sink node *t*.
- For all the existing arcs (blue), assign infinite capacities.

- Add links from source to all positive nodes, with the capacities equal to the weight of the nodes.
- Add links from negative nodes to sink, with the capacities equal to the absolute weight value of the nodes.
- Remove the weights on nodes.

Figure 3 shows the updated graph once all changes are made. The relation between the flow and mining concepts is not as straightforward as the relation between a closure and a pit [26]. One way to describe this is to consider the ore as the water stored in a source that as much as possible needs to be sent to a destination through a pipe network. The source node connects to all ore blocks, and the destination (sink) connects to all waste blocks. In the network, the economic value of a block is not reflected on a node but is measured by the capacity of the pipe (arc) that connects it with the source or the destination. Since the pipes representing block dependency have unlimited capacity, the bottlenecks of the networks are the pipes connected to the source or destination. Three types of pipes can be identified: “waste-to-destination”, “source-to-ore”, and “block-to-block”.

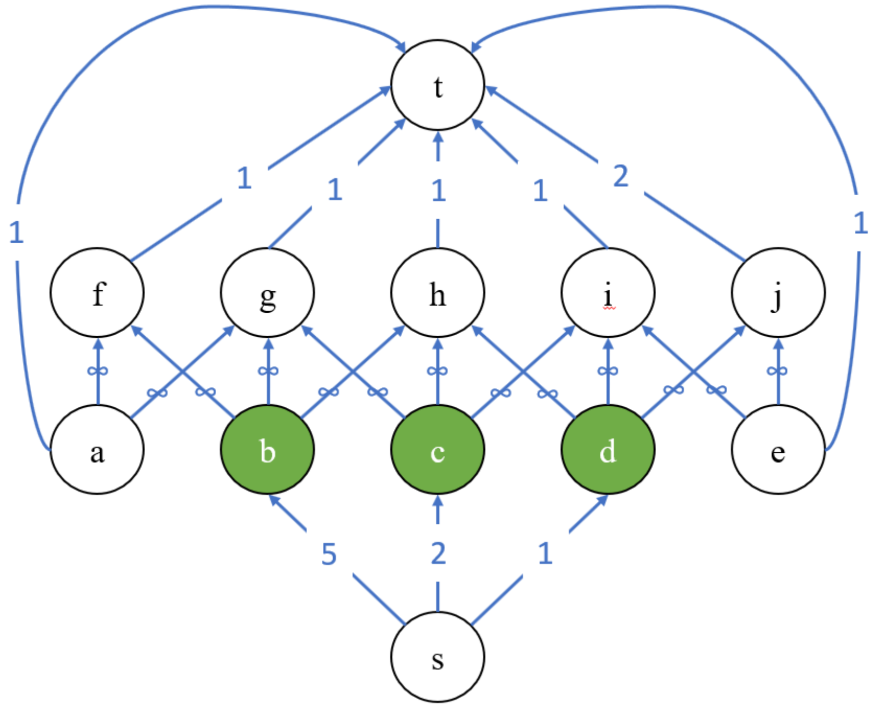

Figure 2. Flow graph representation of the pit optimisation problem.

The conservation constraint at a node v indicates that the excess $e_f(v)$, defined as the difference between the incoming and outgoing flows, is equal to zero. A *preflow* satisfies the capacity constraints and the conservation constraints that requires the excesses to be nonnegative. An arc is *residual* if the flow on it can be increased without exceeding its capacity and *saturated* once the capacity is reached. The residual capacity $u_f(v, w)$ of an arc between nodes v and w is the amount by which the arc flow can be increased. The *distance labelling* d: $V \rightarrow \mathbf{N}$ satisfies the follow conditions: $d(t) = 0$ and for every residual arc (v, w), $d(v) \leq d(w) + 1$. A residual arc (v, w) is *admissible* if $d(v) = d(w) + 1$. A node v is *active* if v is not the source or the sink node, $d(v) <$ number of nodes, and $e_f(v) > 0$.

The push-relabel method maintains a preflow f, initially set to zero an all arcs, and a distance labelling d. The $d(v)$ is initially set to the distance from v to t in the graph. In its first stage, the push-relabel method repeatedly performs the *update operations*, *push* and *relabel* until there are no active nodes left. The update operations modify the preflow f and the labelling d. A *push* from v to w increases $f(v, w)$ and $e_f(w)$ by $\delta = \min\{e_f(v), u_f(v, w)\}$, and decreases $f(w, v)$ and $e_f(v, w)$ by the same amount. A *relabeling* of v sets the label of v equal to the largest value allowed by the valid labeling constraints. The second stage of the method converts f into a flow.

4 A COMPARATIVE CASE STUDY

4.1 Input Block Model

A lignite deposit from the area of West Macedonia in NW Greece was used in the study. It consists of a few lignite layers, and a simpler structure compared to other lignite deposits commonly found in the area. The roof and floor of the mineable lignite area of the deposit was modelled as grid surfaces using inverse distance interpolation. The composited qualities of lignite were also modelled as grids. These grid models were used to generate a stratigraphic block model in Maptek Vulcan as shown in Figure 4. The vertical size of the blocks and their base and top side follows the roof and floor grid models of mineable lignite. The horizontal dimensions of the blocks were 10x10m.

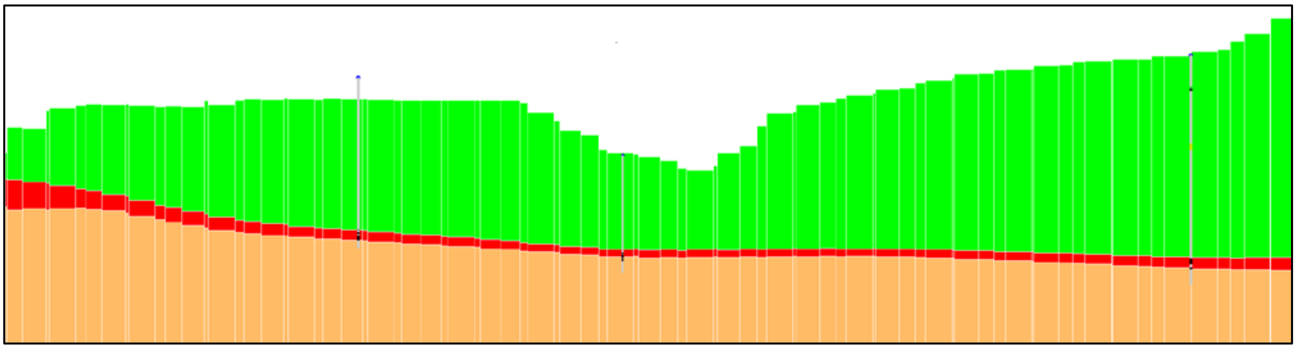

Figure 4. Section through the stratigraphic block model showing overburden (green), mineable lignite area (red), and underburden (orange).

A number of financial parameters were included as variables to the stratigraphic model. These included all mining and processing related costs, and revenue from selling of recovered lignite. The calculation of these parameters was based on the volume of each block, the thickness of mineable lignite and parting, the specific gravity for lignite and waste, and the type of each block (overburden, lignite deposit, underburden). A simple script was used to calculate the necessary parameters in the blocks, as shown in Table 1. The constant values of waste and lignite associated costs and lignite revenue were set using historical information. The script stored the calculated values to corresponding block model variables, including the final block value which represents the undiscounted cash flow of uncovered blocks. This value is necessary as input for the pit optimisation process. The value was positive for the lignite deposit blocks and negative for overburden and underburden blocks.

All current methods of pit optimisation require a regular block model, i.e. a model with equally sized (regular) blocks. This meant that the stratigraphic block model that contained the calculated block values had to be regularised to a standard block size. This size was set to 10x10x8m. Only the block value variable was transferred to the regularised block model as it was the only parameter necessary as input to pit optimisation. This variable was calculated for each block using a sum of the intersecting stratigraphic model blocks' values weighted by their volume inside the regular block. Figure 5 shows the same section shown in Figure 4 but through the regularised model and coloured by block value. Blocks shown in red contain both a lignite and waste component, but the weighted sum of their values results in a positive regular block. These blocks were used as input to the pit optimisation process.

Table 1. Block value calculation script based on mineable lignite thickness, block volume and specific gravities of lignite and waste.

```
if (seam eqs "cx") then
      coal_volume = thickness * 100
      parting_volume = volume - coal_volume
      coal_tonnage = coal_volume * 1.22
      parting_tonnage = parting_volume * 1.6
      revenue = coal_tonnage * 26
      mining_cost = (coal_tonnage * 1.535) + (parting_volume * 0.95)
      processing_cost = coal_tonnage * 1.184
      other_cost = coal_tonnage * 4.054
      block_value = revenue - mining_cost - processing_cost - other_cost
else
      coal_volume = 0
      coal_tonnage = 0
      parting_volume = 0
      parting_tonnage = 0
      waste_volume = volume
      waste_tonnage = volume * 1.6
      revenue = 0
      mining_cost = waste_volume * 0.95
      processing_cost = 0
      other_cost = 0
      block_value = revenue - mining_cost - processing_cost - other_cost
endif
```

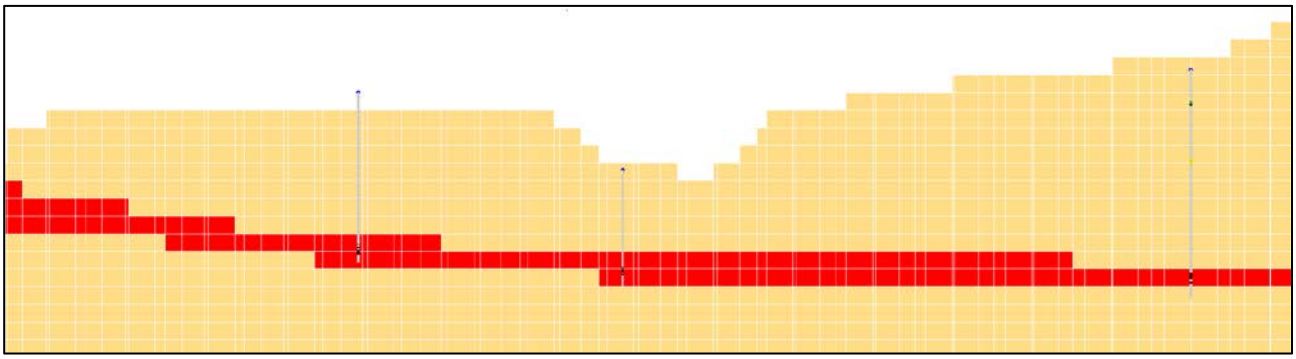

Figure 5. Section through the regularised block model showing positive (red) and negative (orange) blocks passed to pit optimisation.

4.2 Pit Slopes

The second piece of information required by the pit optimisation process is the required pit slopes. In our example, these were based on a conceptual geological model of the deposit area and information related to the stability of different types of rock. The area to be optimised was split into three slope regions based on azimuth as shown in the following figure. A 10^{o} slope interpolation area was used to transit between slope regions. The north-east and east region of the pit (between 0^{o} and 135^{o} azimuth) was considered more stable and was processed with a 45^{o} slope, while the south region (between 135^{o} and 210^{o} azimuth) was considered less stable and was processed with a 33^{o} slope. A 36^{o} pit slope was used in the west and north-west region (between 210^{o} and 360^{o} azimuth).

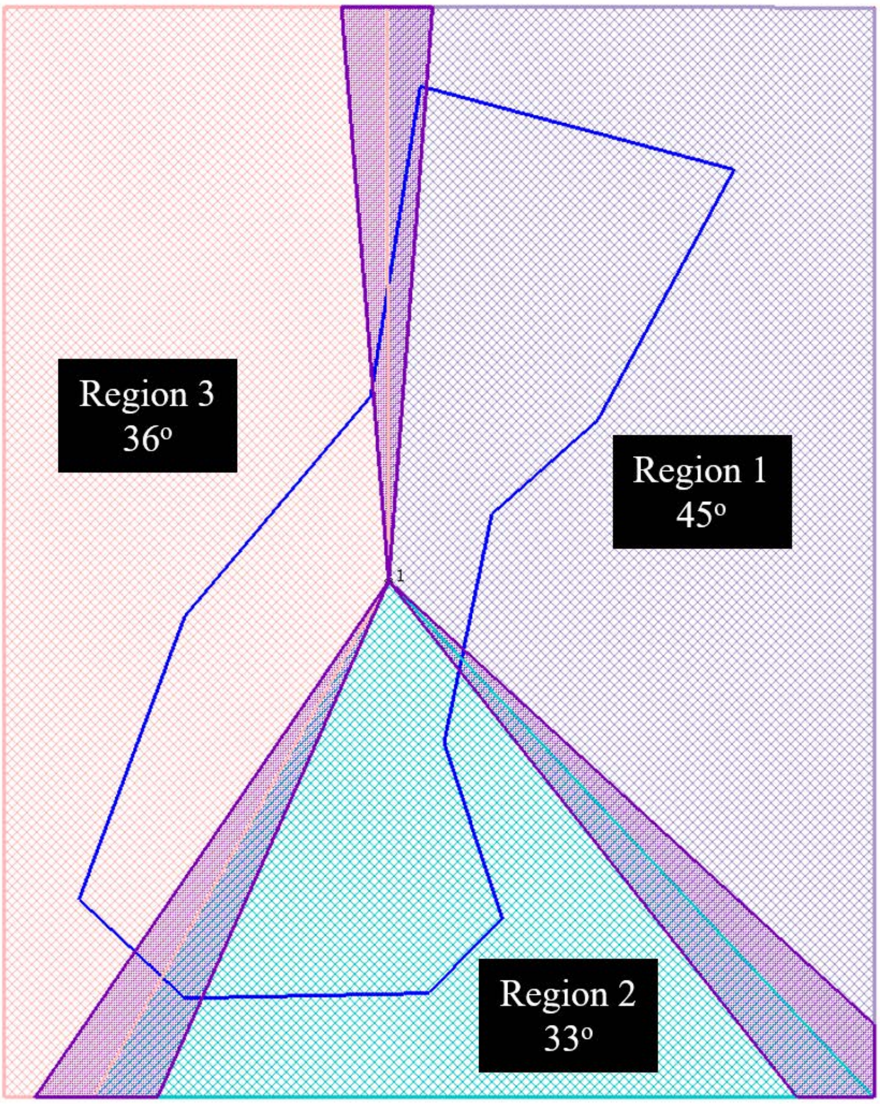

Figure 6. Slope regions and slope interpolation zones shown in plan view over the approximate pit limits and block model extents

4.3 Pit Optimisation

Two separate pit optimisation runs were set up using the same input information (block model and pit slope regions). Two separate block model variables were added to the block model to store the output coding from the two runs – one for the LG method and one for Push-Relabel. Each run produced a separate log file providing details on the input data, optimisation process and output. The following table shows sections from the two log files with information on the blocks used in the optimisation process (feasible blocks), blocks to be mined based on the optimisation (blocks to be mined), ore, waste and air blocks in both cases, the undiscounted economic value of the optimum pit (economic value from the optimum pit) and the time taken to run each optimisation (processing time and total run time respectively).

Once both runs were completed, the optimum pit limits produced by each method were displayed as contours surrounding the blocks to be mined on each bench (level of blocks). The pit limits were 100% identical between the two methods. Figure 7 shows the produced optimum pit limits in plan view. The effect of using different slope regions is clear. The fact that both runs produced identical results was further supported by the information in the log files – both runs produced the same optimum pit economic value based on the same number of blocks to be mined. In other words, the two optimisation runs produced the same result numerically and geometrically to the last block. However, the time spent to produce this result was very different. LG required one hour and 45 minutes to complete the optimisation while Push-Relabel required one minute and 33 seconds!

Table 2. Parts of optimisation log files from the LG (top) and Push-Relabel (bottom) optimisation runs. The first three digits of the economic value of the optimum pit are hidden for confidentiality purposes.

```
- Number of feasible blocks for the optimiser: 856834 blocks.
   -   31916   Ore blocks   (+),
   -  315848   Waste blocks (-),
   -  509070   Air blocks   (0).
 - Maximum arcs for each block................: 87 arcs.
 - Number of arcs evaluated...................: 378776250 arcs.
 - Number of feasible arcs to the optimiser...: 58908200 arcs.
 - Number connections/disconnections made.....: 5349553 connections.
 - Processing time............................: 1:45:42 Hrs.
 - Economic value from the optimum pit........:  xxx94524.69
 - Number of blocks to be mined...............: 484838 blocks.
   -   14533   Ore blocks   (+),
   -  104873   Waste blocks (-),
   -  365432   Air blocks   (0).
```

```
- Number of blocks for the optimiser..........: 856834
 -      31916 Ore blocks   (+),
 -     315848 Waste blocks (-),
 -     509070 Air blocks   (0).
Initialization time...........................: 0:00:01
 - Economic value from the optimum pit.........:  xxx94524.69
 - Number of blocks to be mined................: 484838
 -      14533 Ore blocks   (+),
 -     104873 Waste blocks (-),
 -     365432 Air blocks   (0).
Computation time..............................: 0:00:51
Total time....................................: 0:01:28
 - Pit   0: Factor Index =   0    Value =      0.00000
Total run time................................: 0:01:33 Hrs.
```

5 CONCLUSIONS

Pit optimisation is a process that can automate the definition of pit limits and make open pit design more efficient and less time consuming. The LG algorithm has been well established and accepted by the mining industry as the method for pit optimisation of most mineral deposits. Coal and lignite deposits were not so often approached and designed using pit optimisation. The case study presented in this paper proves that there is value in using pit optimisation for lignite deposits and that the current methods can provide a consistent and efficient way to limit the extents of lignite mines both horizontally and vertically. Speed improvements of the Push-Relabel method open up the opportunity to solve problems consisting of millions of blocks (such as large lignite mines) that were previously too large for the traditional LG method.

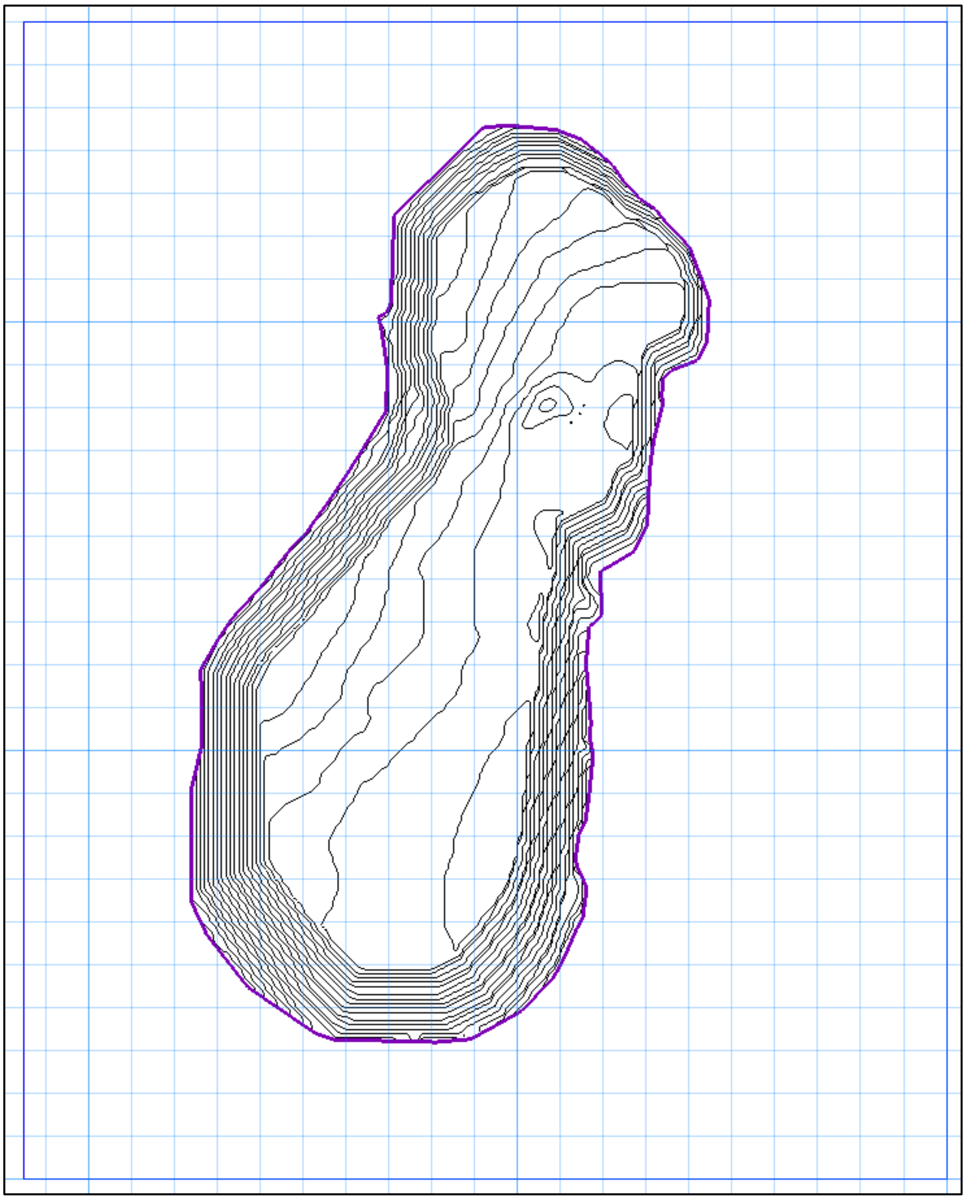

Figure 7. Optimum pit limits produced by both LG and Push-Relabel methods. The effect of applying different slopes in different regions is evident through the change in contour density of the pit walls.

REFERENCES

[1] Poniewierski, J. (2000). Pseudoflow Explained - A discussion of Deswik Pseudoflow Pit Optimization in comparison to Whittle LG Pit Optimization. Deswik Mining Consultants Pty Ltd.

[2] Pana, M. (1965). The simulation approach to open-pit design, In J. Dotson and W. Peters, editors, Short Course and Symposium on Computers and Computer Applications in Mining and Exploration, College of Mines, University of Arizona, Tuscon, Arizona. pp. ZZ–1 – ZZ–24.

[3] Lerchs, H. and Grossmann, I. F. (1965). Optimum design of open pit mines, The Canadian Mining and Metallurgical Bulletin, Vol. 58, January, pp.47-54.

[4] Kim, Y. C. (1978). Ultimate pit design methodologies using computer models the state of the art, Mining Engineering, Vol. 30, pp. 1454-1459.

[5] Seymour, F. (1995). Pit Limit Parameterization from Modified 3D Lerchs-Grossmann Algorithm. SME, Preprint Number 95:96.

[6] Alford, C.G., Whittle, J. (1986). Application of Lerchs–Grossmann pit optimization to the design of open pit mines, In Large Open Pit Mining Conference, AusIMM–IEAust Newman Combined Group, pp. 201–207.

[7] Hustrulid, W.A., Kuchta, M. (2006). Open Pit Mine Planning & Design, 2nd Edition, Taylor & Francis.

[8] Whittle Programming Pty Ltd (1998). Whittle Four-X Strategic Planning Software for Open Pit Mines, Reference Manual.

[9] Picard, J. (1976). Maximal closure of a graph and applications to combinatorial problems, Management Science, Vol. 22, No. 11, pp. 1268–1272.

[10] Goldberg, A.V., Tarjan, R.E. (1988). A new approach to the maximum-flow problem. Journal of the Association for Computing Machinery. 35 (4): 921-940.

[11] King, V., Rao, S. (1992). A faster deterministic maximum flow algorithm. In Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '92, pp 157-164, Philadelphia, PA, USA, Society for Industrial and Applied Mathematics.

[12] Goldfarb, D., Chen, W. (1997). On strongly polynomial dual simplex algorithms for the maximum flow problem, Mathematical Programming 77(2):159-168.

[13] Hochbaum, D., Chen, A. (2000). Performance Analysis and Best Implementations of Old and New Algorithms for the Open-Pit Mining Problem, Operations Research, Volume 48, Issue 6, pp. 823-970

[14] Dantzig, G.B. (1951). Application of the Simplex Method to a Transportation Problem. In T. C. Koopmans, editor, Activity Analysis and Production and Allocation, pages 359–373. Wiley, New York, 1951.

[15] Dantzig, G.B. (1962). Linear Programming and Extensions. Princeton University Press, Princeton, NJ, 1962.

[16] Ford, Jr., L.R., Fulkerson, D.R. (1962). Flows in Networks. Princeton University Press, Princeton, NJ.

[17] Dinitz, E.A. (1970). Algorithm for Solution of a Problem of Maximum Flowing Networks with Power Estimation. Soviet Math. Dokl., 11:1277–1280.

[18] Goldberg, A.V. (1987). Efficient Graph Algorithms for Sequential and Parallel Computers. PhD thesis, M.I.T., Cambridge, MA.

[19] Karzanov, A.V. (1974). Determining the Maximal Flow in a Network by the Method of Preflows. Soviet Math. Dokl., 15:434–437.

[20] Tarjan, R.E. (1984). A Simple Version of Karzanov's Blocking Flow Algorithm. Oper. Res. Lett., 2:265–268.

[21] Anderson, R.J., Setubal, J.C. (1993). Goldberg's Algorithm for the Maximum Flow in Perspective: a Computational Study. In D. S. Johnson and C. C. McGeoch, editors, Network Flows and Matching: First DIMACS Implementation Challenge, pages 1–18. AMS, Providence, RI.

[22] Derigs, U., Meier, W. (1989). Implementing Goldberg's Max-Flow Algorithm—A Computational Investigation. ZOR—Methods and Models of Operations Research, 33:383–403.

[23] Derigs, U., Meier, W. (1992). An Evaluation of Algorithmic Refinements and Proper Data-Structures for the Preflow-Push Approach for Maximum Flow. NATO ASI Series on Computer and System Sciences, vol. 8, pp. 209–223. Nijhoff, The Hague.

[24] Nguyen, Q.C., Venkateswaran, V. (1993). Implementations of Goldberg–Tarjan Maximum Flow Algorithm. In D. S. Johnson and C. C. McGeoch, editors, Network Flows and Matching: First DIMACS Implementation Challenge, pp. 19–42. AMS, Providence, RI.

[25] Cherkassky, B.V., Goldberg, A.V. (1997). On Implementing Push-Relabl method for the maximum flow problem. Algorithmica 19: 390-410.

[26] Bai, V.X., Turczynski, G., Baxter, N., Place, D., Sinclair-Ross, H., Ready, S. (2017). Pseudoflow method for pit optimization, Whitepaper, Geovia-Whittle, Dassault Systems.