Rem: Revista Escola de Minas
ISSN: 0370-4467
editor@rem.com.br
Escola de Minas
Brasil

Ribeiro Souza, Felipe; Melo, Michel; Lopes Pinto, Cláudio Lúcio
A proposal to find the ultimate pit using Ford Fulkerson algorithm
Rem: Revista Escola de Minas, vol. 67, núm. 4, noviembre, 2014, pp. 389-395
Escola de Minas
Ouro Preto, Brasil

A proposal to find the ultimate pit using Ford Fulkerson algorithm

Uma proposta para determinar cava final utilizando Ford Fulkerson

http://dx.doi.org/10.1590/0370-44672014670166

Felipe Ribeiro Souza
Universidade Federal de Minas Gerais
Belo Horizonte – Minas Gerais - Brazil
felipecmc@globomail.com

Michel Melo
Universidade Federal de Minas Gerais
Belo Horizonte – Minas Gerais - Brazil
michelmelo@gmail.com

Cláudio Lúcio Lopes Pinto
Universidade Federal de Minas Gerais
Belo Horizonte – Minas Gerais - Brazil
cpinto@ufmg.br

Abstract

The present study is focused on mining planning with an emphasis on the graph theory model proposed by Lerchs-Grossmann. The original paper published by Lerchs-Grossmann about determination of optimum final pit does not report the computational algorithm to solve the problem. This paper discusses and presents an algorithm based on the maximum flow graph computational work from Ford Fulkerson. The main steps for solving the problem and the results of the two-dimensional models are discussed.

Keywords: Lerchs-Grossmann; Optimum Pit; Graph; Ford Fulkerson.

Resumo

O presente trabalho tem como foco o planejamento de lavra com ênfase no modelo da teoria dos grafos de Lerchs-Grossmann. O trabalho publicado pelo autor em que se discute o teorema para a determinação da cava final ótima não apresenta o algoritmo computacional para resolver o problema. Esse trabalho apresenta um algoritmo baseado no fluxo máximo dos grafos como discutido no trabalho computacional de Ford Fulkerson. Serão apresentados os passos principais para resolução do problema e os resultados dos testes realizados para modelos bidimensionais.

Palavras-chave: *Lerchs-Grossmann; cava final; grafo; Ford Fulkerson.*

1. Introduction

The purpose here is to suggest a logical construction of a system to search for the final pit. This algorithm provides all principles to build the algorithm in 2D and 3D. It begins by constructing the neighborhood matrix that is responsible for indicating to the algorithm which block should be removed in order to mine a lower block.

A depth search engine was the chosen tool to find the validity group of blocks to be mined and, from all the sequences validated, the one that will be, in fact, applied. This searching mechanism reflects the property of this methodology of emulating mining operations. The branches generated by these processes are visually similar to the cuts of mining extractions.

After identifying the graph, Ford Fulkerson theorem will be used to select the feasible ones and, from those, to finding the set of blocks that optimize the "profit" flow.

2. Neighborhood Matrix

The neighborhood matrix is the mathematical argument to identify the neighbors of a certain block. For example, in a block model with N elements, an NxN array can be assembled as shown in figure 1.

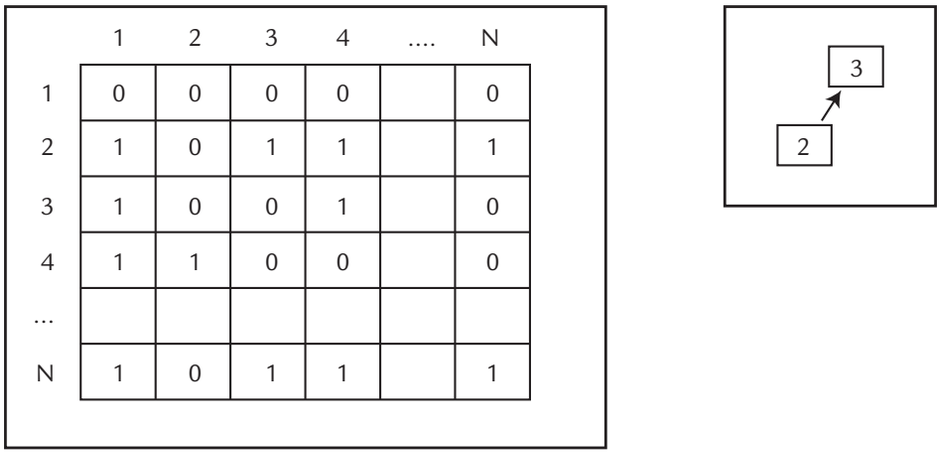

Figure 1
Model of Neighborhood Matrix and One-pointed arc connecting between blocks

The digit 1 seen in the table at position (2, 3) means that there is a one-pointed arc from block 2 to block 3. Otherwise, at the position (3,2), the digit 0 found indicates there will be no arc connecting block 3 to block 2, defining a one-pointed arc as shown in figure above.

3. Search in depth

The concept of stack will be used for searching in depth. This stack will be constructed by superimposing the blocks, starting with the deepest one, using the arcs defined by the neighborhood matrix. Considering the block model of figure 2 as an example, the deepest block will be the base of the stack. The neighborhood matrix will be used to find the adjacent block to the block 1 that will be the next to be place at the stack (block 4)

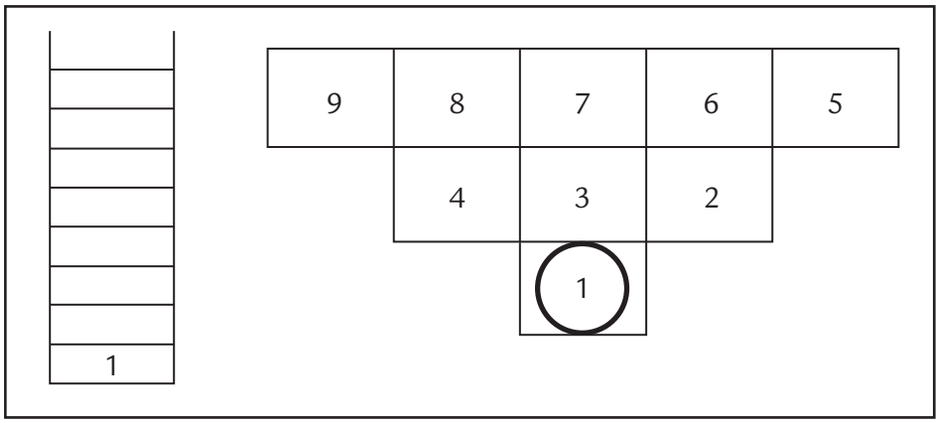

Figure 2
Searching in depth (step 1)

Considering again the neighborhood matrix to verify block 4 nearest neighbors, the block 9 will be placed at the stack and so forth in an interactive process.

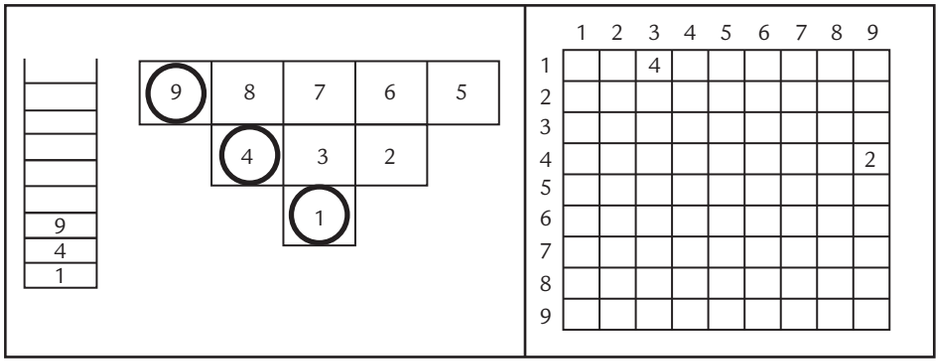

Figure 3
Search step 2 and Neighborhood Matrix filled by flow

In this example, the neighborhood matrix shows that block 9 does not have any neighbor. The algorithm eliminates, then, the block 9 from the stack and restarts the search from the previous position (block 4). Similarly, all blocks already computed will be eliminated from additional searching. In the step illustrated by figure 4, blocks 7 and 8 don't have neighbors, consequently, are eliminated from the stack and return the algorithm to block 4. Furthermore, all neighbors of block 4 have already been visited and likewise are excluded from the stack.

Thus, the algorithm returns to block 1, where neighbors 3 and 4 are still able to be part of the search. This

cyclic mechanism is repeated until all blocks are removed from the stack. The blocks sequence encountered is stored in the neighborhood flow matrix. It should be pointed out that the example used here represents 45°slope angles; if gentler angles are to be modeled the neighborhood matrix should enclose further neighboring blocks.

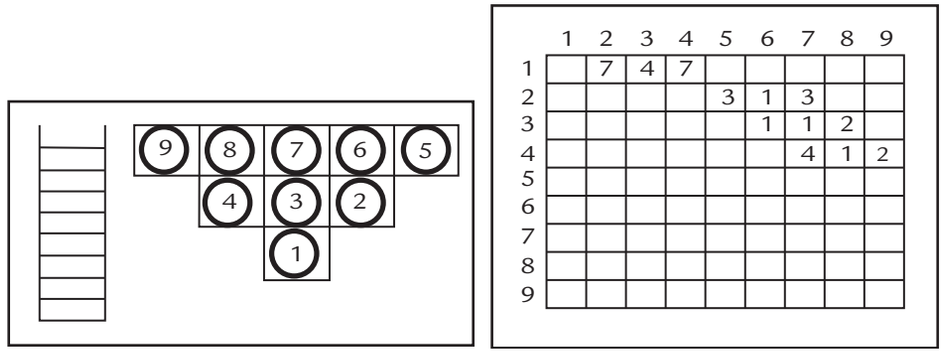

Figure 4
Search final step and
Neighborhood Matrix filled by flow

4. Ford Fulkerson

The algorithm proposed by Ford Fulkerson is very efficient in solving the problem of maximizing flow. It works with a set of directed graphs that represent the flow from one end (source) to the other (destination).

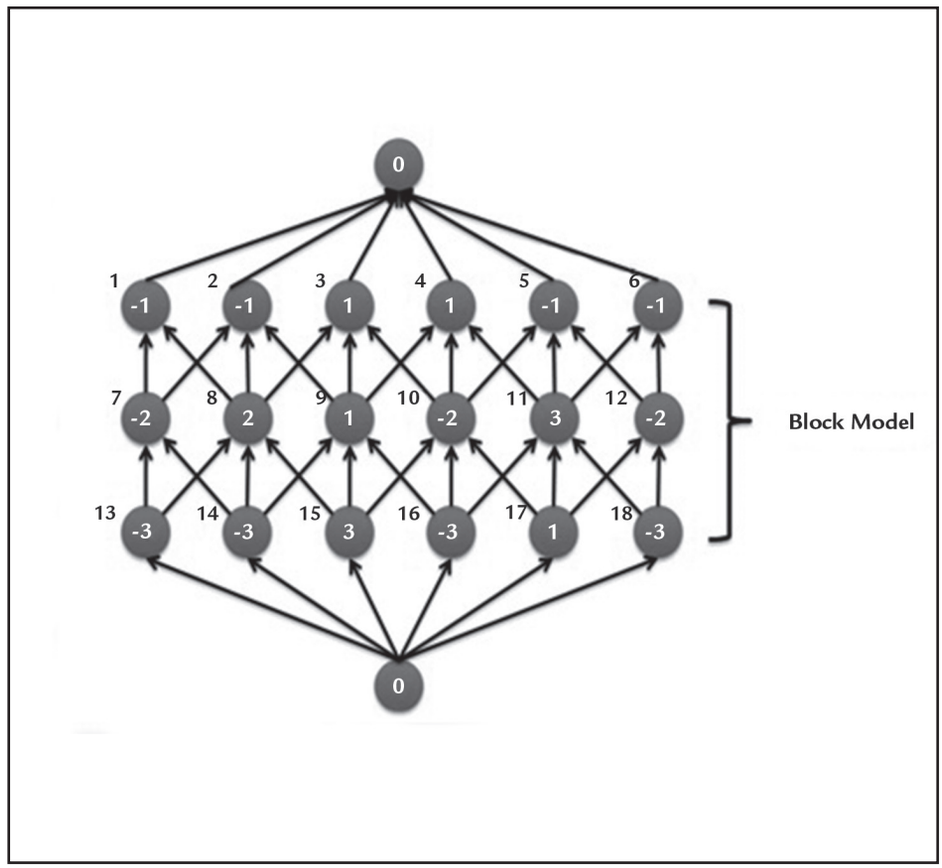

Figure 5
Block Model according
to Fulkerson algorithm

Even though the benefit function can assume values below zero, negative vertices are not allowed in a network flow. All the other vertices are, therefore, increased by the same value, then normalizing the tree. For better understanding of the algorithm theory, some concepts must be accepted:

- Arc capability: Maximum flow carried by the arc (Cij);
- Remaining capacity of an arc: If through the arc(ij) flow a quantity (ϕij) lower than its capacity (Cij) Its remaining capacity is given by Cij – ϕij;
- Saturated Arc: The arc whose remaining capacity is null;
- Saturated Elementary Path: It is a path without repeat vertices; with no negative flow and at least one of its arcs is saturated.
- Saturated Chain: At least one of its arc has zero remaining capacity in the forward direction. At least one of its arc has zero flow in reverse direction.

5. Ford Fulkerson implementation

The example presented in Figure 6 will be used to illustrate the first step of the algorithm. This step consists in saturating the elementary paths.

The values in the block model have been normalized.

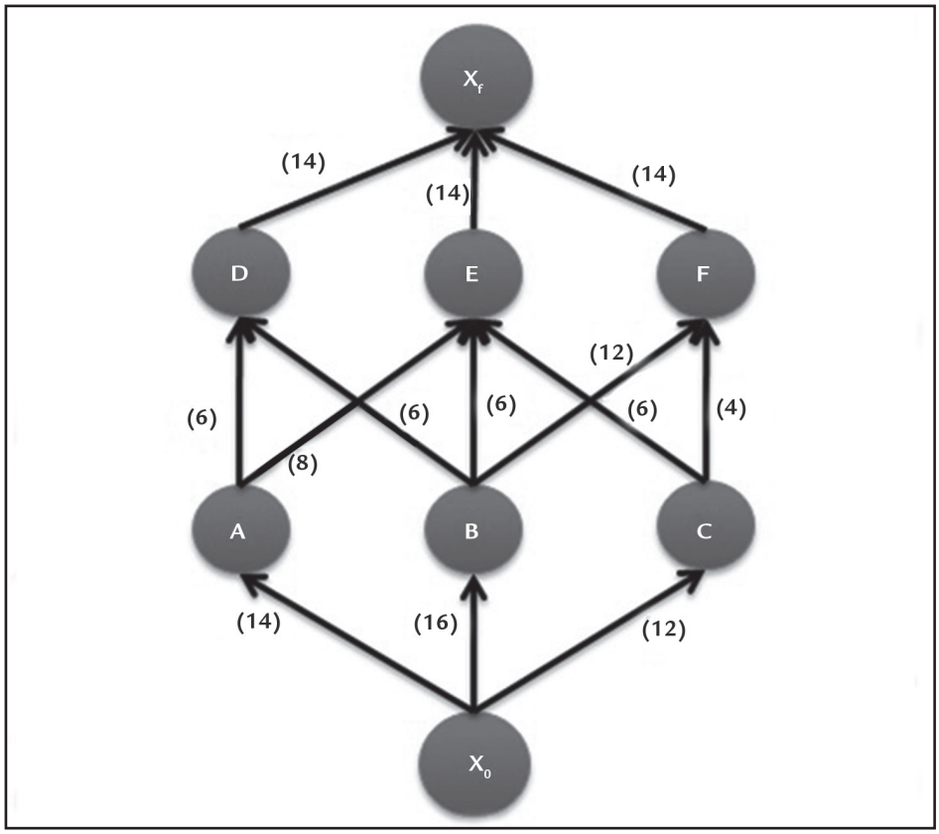

Figure 6
Block Model example for Ford Fulkerson.

All paths must be searched individually and orderly. Path X_0-A-D-X_f must be investigated prior to path X_0-A-E-X_f and so on. In this example, the algorithm would build up the information given in table 1 and the effective flow shown in figure 7.

Here, the flow corresponds to the benefits function.

Path	Arcs with the smallest remaining capacity	Saturating flow	Arc(s) Saturated
X_0-A-D-X_f	(A,D)=6	6	(A,D)
X_0-A-E-X_f	(X_0,A)=8;(A,E)=8	8	(X_0,A);(A,E)
X_0-B-D-X_f	(B,D)=6	6	(B,D)
X_0-B-E-X_f	(B,E)=6;(E,X_f)=6	6	(B,E)(E,X_f)
X_0-B-F-Xf	(X,B)= 4	4	(X,B)
X_0-C-F-X_f	(C,X_f)=4	4	(C,X_f)

Table 1
Saturated elementary paths

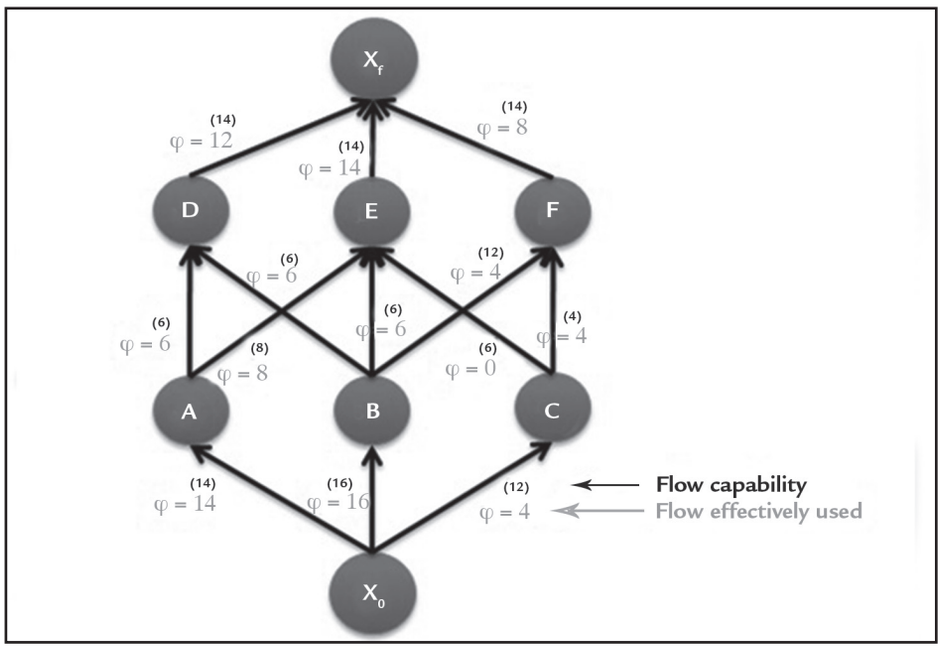

Figure 7
Flow relative to the saturation of the elementary path

The flow entering the system through the node X0 (14 +16 +4 = 34) matches the flow exiting the system through node Nf (12 + 14 + 8 = 34). At this point one has maximized the direct flow of this graph. Nevertheless, the goal of the algorithm is global maximization, which can be seen as the duality problem in linear systems. The algorithm, then, guarantees the minimization of the remaining flow in the opposite direction.

First step (Primal)

One of the most important concepts used in programming is of duality. A problem associated to another problem is classified as Dual where the original problem is called Primal. In this context it is represented by the objective of maximizing the flow of blocks in the forward direction. In this step, the aim is to maximize the matrix related to maximum flow. The associated problem referred is to as Dual, which in this case, is the reverse flow.

$$\text{Maximize: } Z = Cx \text{ Constrained by: } AX \leq b \; X \geq 0$$

Second step (Dual)

The dual stage aims to minimize the remaining flow of the non-saturated chains. To detect an unsaturated chain from X0 to Xf the following steps should be performed.

Since the problem has to do with the reverse flow, the aim is to minimize the transposed matrix related to minimize constraining flow.

$$\text{Minimize: } W = \text{by Constrained by: } ATY \geq C \quad Y \geq 0$$

After applying the procedures to minimize the reverse flow (dual problem) the optimal flow is achieved, which in the example used here is 40 (X_f = 12+14+14) as shown in figure 8.

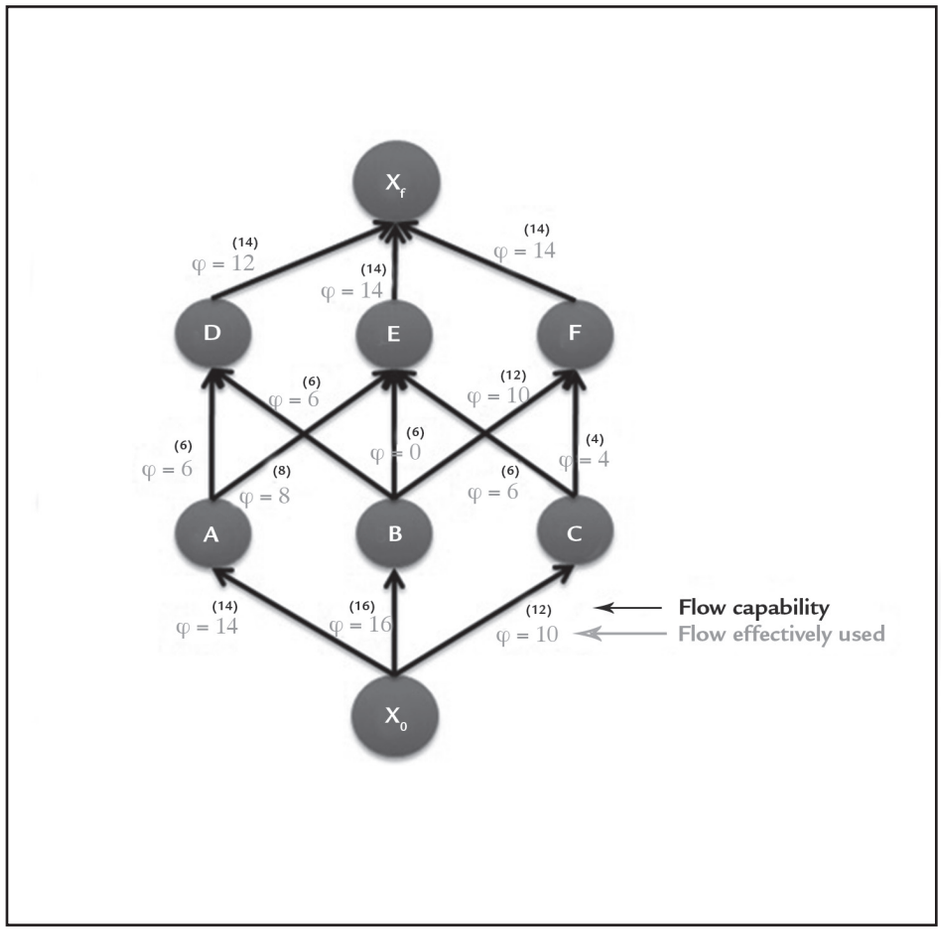

Figure 8
Dual result of Ford Fulkerson.

Third Step (Flow Cutting):

The arcs that does not affect the maximum network flow should be "deleted". In the case of a block model, the blocks that do not contribute to increase the benefit function would be casted off. The final pit not just finds the maximum possible flow, but also should identify which blocks that should be mined and which should not. Thus, the remaining vertices (blocks) in the network flow must comply with two conditions: There would be at least one arc in the forward direction with positive remaining capacity. There would be at least one arc in backward direction with positive flow.

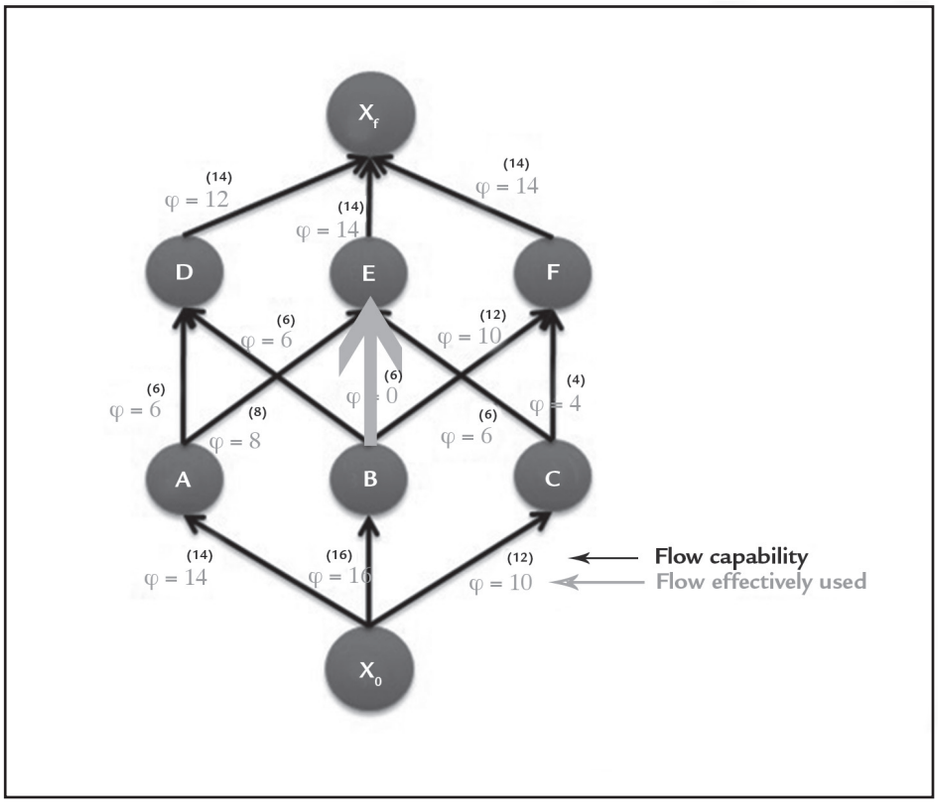

Figure 9
Tree cut

6. Result

To illustrate the applicability of the algorithm discussed in this paper, table 2 shows the results of the Lerch Grossman algorithm applied to the previous section.

-1	-2	-1	0	6	11	24
-3	0	1	7	12	25	23
-6	-1	4	11	24	26	20
-10	-8	-1	20	21	25	16

Table 2
Lerch Grossman final pit

The pit value is 24 financial unities and the blocks at the pit limit are highlighted. The same section was used as input into the algorithm using the Python programing language.

The output is presented in figure 10.

The same blocks are selected to be mined resulting in the same pit value.

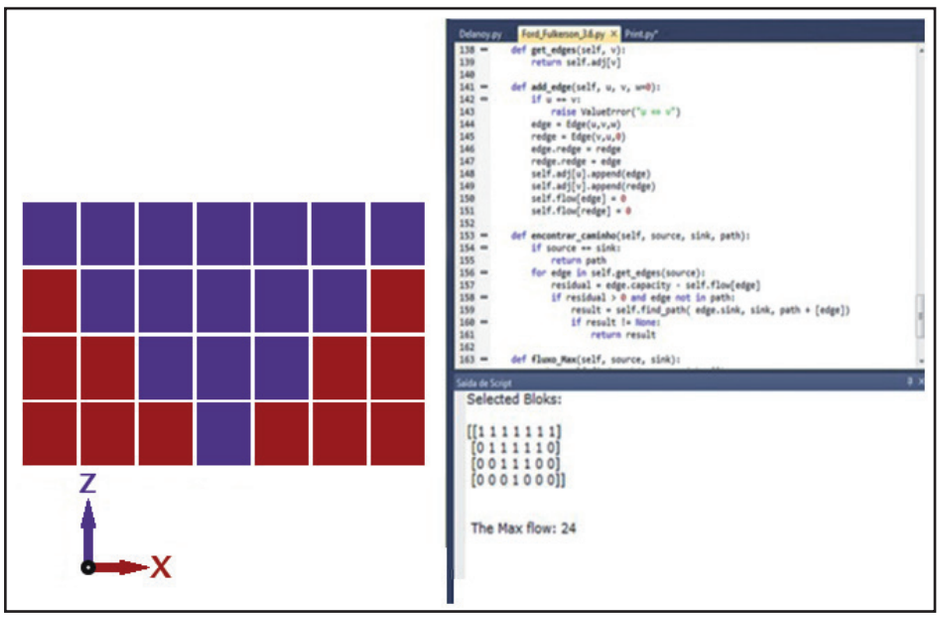

Figure 10
Results classical methodology x Block Model and results of the algorithm

7. Conclusion

Ford Fulkerson algorithm prioritizes the cash flow results to analyze the whole set of blocks. The search engine in depth based on the idea of building stacks to find the pits candidates proved to be very simple For a three dimensional system, the neighborhood matrix needs not be modified. however, an indexing algorithm should be used to identify the sections in the actual block model.

The graph theory showed to be applicable to mathematically represent a geological block model. The matrix was built up in a way that its maximum and its minimized transpose represents the set of blocks that optimize the project cash flow.

The mathematical algorithm finds the true optimum unlikely approximation methodologies. A two-dimensional section was used to describe the application of the algorithm. The results were compared with the traditional Lerch-Grossmman algorithm achieving the same results. Nevertheless, the Graph would be more suited to manage irregular blocks and models with different angles of slope.

8. References

BAYER, D. A., AND LAGARIAS, J. C., The Nonlinear Geometry of Linear Programming, Part I: Affine and Projective Scaling Trajectories. Trans. Amer. Math. Soc. V. 314, n. 2, p. 499–526, 1989.

BLAND, R. G., New Finite Pivoting Rules for the Simplex Method, Math. Oper.Res, p. 103-107, 1977.

DANTZI, G. B Linear Programing and Extensions, United States Air Force Project Rand, 1963, p 94 – 209

ECKER J. G., HEGNER N. S., KOUADA I. A., Generating all maximal efficient faces for multiple objective linear programs. J Optim Theory Appl, v.30, n. 3, p. 353-354, 1980.

EHRGOTT M, PUERTO J, RODRIGUEZ-CHIA AM A primal-dual simplex method for multiob-jective linear programming. Journal of optimization theory and applications, v. 134, n.3, p. 483-497, 2006

FRENCH S., HARTLEY R., THOMAS L. C., WHITE D. J. (EDS). Multi-objective decision making. New York: Academic Press,, p. 13-98, 1983.

HARTLEY R. Survey of algorithms for vector optimization problems in multi-objective decision making. Multi-objective decision making. New York: Academic Press, 1983.

HUSTRULID, W.; KUCHTA, M. Open pit mine planning and design. (2nd. Ed). London: Taylor and Francis, 2006. v. 1 (Fundamentals).

KHALOKAKAIE R. , Computer-Aided Optimal Open Pit Design With Variable Slope Angles. The University of Leeds, 1999. (Ph.D. Thesis,)

LÖHNE A., Optimization with set relations: conjugate duality. Optimization, v. 54, n. 3, p. 265-282. 2005. DOI: 10.1080/02331930500096197

LUENBERGER, D. G., YINYU Y. , Linear and Nonlinear Programming, Springer. V. 116, p. 10-102, 2008

MEHLHORN, K., NÄER, S. LEDA: A platform for combinatorial and geometric computing. Commun: ACM, p 38. 96 –102, 1955

TRECATE, G. F, Notas de aula,Dipartimento di Ingeneria Industriale e dell'Informazione,Università degli Studi di Pavia, 2010

Received: 29 August 2014 - Accepted: 3 October 2014.