# Development of an Algorithm to Find the Optimum Dredging Region for Short Term Scheduling

* Fernando JMMP, Cooray TMJA and Dissanayake DMDOK
*Corresponding author - marlonfdo@yahoo.com

**Abstract:** This research focuses on the short-term scheduling of the vertical slicing method applied dredge mine sites. An algorithm was developed to outline the region to mine, when the block model, optimum pit limit, topography limit and the market demand are given. The algorithm is based on the concepts of dynamic programming and zero-one integer programming to avoid repetitive solutions and memorize the previous stage outcome for the next stage process. It utilizes 3D matrix to store previous and next stage solutions with three integers representation for "possible future mining", "already mined" and "never mine" conditions. The algorithm is used on 2D resource block model, which is obtained by pre-processing optimized 3D block model to 2D plan-view block model. The developed algorithm was faster and required less data storage over the conventional method due to exclusion of repetitive solutions in the processing.

**Keywords:** Block Model, Dynamic Programming, Integer Programming, Mining

## 1. Introduction

Dredge mining is underwater excavation method of a placer deposit. It is done using a floating vessel called dredge. Mine region selection for the dredge is a part of mine short-term scheduling. This research serves as an operational improvement on finding the optimum dredging region at the bottom to top vertical slicing method applied dredge operations. Dredge mine site is represented by a collection of cells called 3D block model. Each cell carries details of the mineral respect to the cell location. The topography of the site is given as a CAD file generated from GPS coordinates of surveying. The optimum pit depth for the dredge mine site is an outcome of pit optimization algorithms.

*DMDOK Dissanayake, C. Eng., MIE(SL), B.Sc. Eng. (Hons) (Moratuwa), Ph.D. (Seoul), Senior Lecturer in Department of Earth Resources Engineering, University of Moratuwa.*
*TMJA Cooray, B.Sc. (Sp) Mathematics (SL), PG.Dip. Maths (Peradeniya), M.Sc. (Colombo), M.Phil (Moratuwa), Senior Lecturer in Department of Mathematics, University of Moratuwa.*
*JMMP Fernando, AMIE(SL), B.Sc. Eng. (Hons) (Moratuwa), M.Sc. (Moratuwa), Mining Engineer in MineMap Pty. Ltd., Colombo.*

This research focused on finding the region of the best fit for the customer demand on the optimized pit depth and topography data carrying 3D block model. The access a vertex (the collection of same x-y coordinates having cells) for mining will be possible, if at least one of four of it's adjacent vertices is already excavated in the plan-view 2D grid which was built from the 3D model.

The objective of this research was to develop an efficient algorithm to find the optimum region with faster processing capabilities requiring lesser intermediate solutions in storage capacity over the conventional algorithm. It searches for all possible solutions from the last vertex mining to next vertex mining, reproducing the same solutions.

## 3. Material and Methods:

### 2.1 Data Used:

A real world data set of a South African continent dredge mine site 3D optimum pit block model called "SA Dataset" was used for analysing the developed algorithm. Further, auto generated 2D plan-view resource matrices were used for testing purposes. Unit value resource matrix of 15*15 with centre cell as initial location was used to test performance of the algorithm with increment of cells in the solution.

### 2.2 Pre-processing of industry data

The actual mine site 3D block model data has to be treated so that it can be represented as a plan-view 2D resource matrix. It was done by getting the summation of mineral commodity tonnages of stratigraphies of each plan view location. Figure 3.8 shows an example of a pre-processing for a section of a 3D block model.

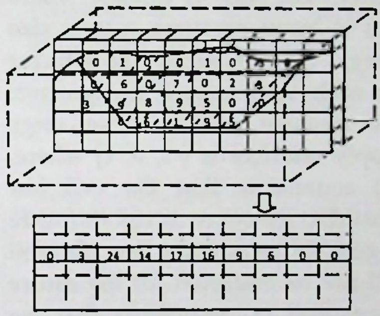

*Figure 1 - Pre-process of 3D to 2D.*

### 2.3 Mathematical representation of data and process

Input data are;

- 2D matrix A(i,j) , where A(i,j) is matrix with real numbers, i=1,2,3,... m and j=1,2,3,... n
- Market demand D, Where D is single value real number
- Dredge initial excavation location (p,q), Where $1 \le p \le m$, $1 \le q \le n$

Process of minimization of region;

Min $Z = \sum X(i,j)$ , where $X(i,j)=1$(mine) or 0(leave)

Subjected to:

- $\sum A(i,j)X(i,j) \geq D$ (To supply become greater than or equal to the demand)
- $X(p,q)=1$(To initial location become an excavating vertex)
- If $X(i,j)=1$ then $X(i+1,j)+X(i-1,j)+X(i,j+1)+X(i,j-1)\geq 1$ (To output region to become a single island)

## 2.4 Proposed Algorithm:

Use two 3D matrices, one as present stage supply matrix and the other as previous stage supply matrix. The size is *mxnxs*, where *mxn* is input resource matrix size and *s* is the number of solutions for the each stage and it is dynamic. The element set of those stage supply matrices is {-1, 0, 1} where, (-1) represents that the cell has possibility to be extracted in future stages, (0) represents that the cell will not be extracted for the future stages and (1) represents that the cell is to be extracted in the concerned stage. Each *mxn* matrix of present supply matrix scalar multiplication with input resource matrix provides present resource supply matrix. The summation of each *mxn* matrix of the present resource supply matrix produces 1x1x*s* size present total supply matrix. Until one or more elements of the present total supply matrix exceeds the demand, the process of finding the optimum region has to be continued with replacing the

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

**Figure 2 – Stage 1 output mxnx1.**

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 1 | -1 | -1 |
| -1 | -1 | 1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 0 | -1 | -1 |
| -1 | -1 | 1 | 1 | -1 |
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 0 | -1 | -1 |
| -1 | -1 | 1 | 0 | -1 |
| -1 | -1 | 1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 0 | -1 | -1 |
| -1 | 1 | 1 | 0 | -1 |
| -1 | -1 | 0 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

**Figure 3 – Stage 2 output mxnx4.**

| | | | | |
|---|---|---|---|---|
| -1 | -1 | 1 | -1 | -1 |
| -1 | -1 | 1 | -1 | -1 |
| -1 | -1 | 1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | 0 | -1 | -1 |
| -1 | -1 | 1 | 1 | -1 |
| -1 | -1 | 1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | 0 | -1 | -1 |
| -1 | -1 | 1 | 0 | -1 |
| -1 | -1 | 1 | 1 | -1 |
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | 0 | -1 | -1 |
| -1 | -1 | 1 | 0 | -1 |
| -1 | -1 | 1 | 0 | -1 |
| -1 | -1 | 1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | 0 | -1 | -1 |
| -1 | -1 | 1 | 0 | -1 |
| -1 | 1 | 1 | 0 | -1 |
| -1 | -1 | 0 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | 0 | -1 | -1 |
| -1 | 1 | 1 | 0 | -1 |
| -1 | 0 | 1 | 0 | -1 |
| -1 | -1 | 0 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 0 | 1 | -1 |
| -1 | -1 | 1 | 1 | -1 |
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 0 | 0 | -1 |
| -1 | -1 | 1 | 1 | 1 |
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 0 | 0 | -1 |
| -1 | -1 | 1 | 1 | 0 |
| -1 | -1 | -1 | 1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 0 | 0 | -1 |
| -1 | -1 | 1 | 1 | 0 |
| -1 | -1 | 1 | 0 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 0 | 0 | -1 |
| -1 | 1 | 1 | 1 | 0 |
| -1 | -1 | 0 | 0 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 0 | -1 | -1 |
| -1 | -1 | 1 | 0 | -1 |
| -1 | -1 | 1 | 1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 0 | -1 | -1 |
| -1 | -1 | 1 | 0 | -1 |
| -1 | -1 | 1 | 0 | -1 |
| -1 | -1 | 1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 0 | -1 | -1 |
| -1 | -1 | 1 | 0 | -1 |
| -1 | 1 | 1 | 0 | -1 |
| -1 | -1 | 0 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 0 | -1 | -1 |
| -1 | 1 | 1 | 0 | -1 |
| -1 | 0 | 1 | 0 | -1 |
| -1 | -1 | 0 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 0 | -1 | -1 |
| -1 | 1 | 1 | 0 | -1 |
| -1 | 1 | 0 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | 0 | -1 | -1 |
| 1 | 1 | 1 | 0 | -1 |
| -1 | 0 | 0 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

| | | | | |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| -1 | 1 | 0 | -1 | -1 |
| 0 | 1 | 1 | 0 | -1 |
| -1 | 0 | 0 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 |

**Figure 4 – Stage 3 output mxnx18.**

Previous stage supply matrix by the present stage supply matrix and computing present stage supply matrix. 1st, 2nd and 3rd stages of the algorithm are graphically represented for 5x5 matrices as Figure 2, 3 and 4.

### 2.5 Testing and analysis

The developed algorithm was coded using Java programming language and tested for "SA Dataset", auto generated data sets and unit matrices.

## 3. Results

The results from both conventional and developed algorithms are shown in Table 1. The "SA Dataset" input 2D matrix size is 47x49 and (6,6) cell location was the initial dredger location.

*Table 1 – No. of solutions*

| Dataset | No. of solutions | | No. of Repetitions |
|---|---|---|---|
| | Conventional algorithm | Developed algorithm | |
| SA(47x49) | 43 | 23 | 20 |
| Auto1 (5x6) | 2 | 2 | - |
| Auto2(3x3) | 3 | 3 | - |
| Auto3 (3x4) | 5 | 3 | 2 |
| Auto4 (3x3) | 2 | 2 | - |
| Auto5 (4x4) | 5 | 4 | 1 |

### 3.1 Stage solutions

Number of intermediate solutions for stage 1, 2 and 3 for each algorithm is graphically shown in Figure 5 and 6.

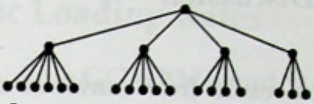

*Figure 5 – Stage solutions' tree diagram by the developed algorithm.*

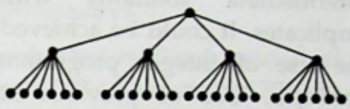

*Figure 6 – Stage solutions' tree diagram by the conventional method.*

Solution storage capacity and processing time requirements for conventional and developed algorithms were studied using 16x16 size unit matrix and the results are shown in Table – 2 and 3.

*Table 2 – Solutions storage capacity*

| No of cells extract (stage) | Required matrix size | |
|---|---|---|
| | Conventional algorithm | Developed algorithm |
| 1 | 1 | 1 |
| 2 | 4 | 4 |
| 3 | 24(=4x6) | 18 |
| 4 | 192(=24x8) | 102 |
| 5 | 1920(=192x10) | 672 |
| 6 | 23040(=1920x12) | 5124 |

*Table 3– Processing time*

| No of cells extract (stage) | Processing Time (ms) | |
|---|---|---|
| | Developed algorithm | Conventional Algorithm |
| 1 | 8330 | 8402 |
| 2 | 8435 | 8446 |
| 3 | 11638 | 19223 |
| 4 | 29047 | 49031 |
| 5 | 123428 | 374521 |
| 6 | 1025391 | 4481368 |

## 4.Discussion

The results from conventional and developed algorithm show that the developed algorithm delivers the intermediate solutions without duplicates. It could be achieved by the use of integer programming concepts. Use of previous stage solution for the next stage solution building which is a concept of dynamic programming, has reduced the processing time of the developed algorithm.
The developed algorithm is a shapes generating algorithm, which is isolated from the input data. The algorithm can be run for unit matrix and the generated output regions (shapes) can be utilized with input data to find the optimum regions.

## 5.Conclusion

The developed algorithm is efficient over the conventional method due to the faster process and the requirement of less solution storage capacity. The process of nominating the cells of 2D grid as 'mined', 'never mine' and 'possible mine' gives flexibility for non mining areas. Commencing of finding the optimum region by the algorithm for any scale of dredge mine, creates positive financial impact by minimizing mineral storage and labour force, particularly global financial crisis periods.

## Acknowledgements

The authors wish to acknowledge the assistance received from many individuals in the department of mathematics at the University of Moratuwa, Sri Lanka, and thankful to Mr. Seldon Mart - CEO, MineMap Pvt Ltd. and Mr. Layne Botes Planning Engineer, Moma mine site, Mozambique for providing exposure to global mining industry and also appreciate Miss S. Wanniarachchi and Mr. H. Liyanachchi for helping code generation for the algorithm.

## References

De Kock, P., (2007). A back to basics approach to mining strategy formulation, *The 6th international heavy mineral conference 'Back to basics'* The Southern African Institute of Mining and Metallurgy, Richard bay minerals.

Hahlbrock, U., (1985). Dredging of heavy minerals, *Bulk handling in open pit mines & quarries* 1981-1985-volume F/86 Transtech, Germany.

Ramazan S., Dimitrakopoulos R. (2004). *Recent applications of operations research and efficient MIP formulations in open pit mining*, Society for Mining, Metallurgy, and Exploration, Inc.

Taha, H. A., (2006). *Operations Research an Introduction*, 8th ed. Prentice Hall.