

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:

Yarmuch, Juan Luis

Title:

Optimisation in open-pit mine planning

Date:

2020

Persistent Link:

<https://hdl.handle.net/11343/241387>

Terms and Conditions:

Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.

# Optimisation in open-pit mine planning

Juan L. Yarmuch

ORCID: [orcid.org/0000-0002-7067-6765](https://orcid.org/0000-0002-7067-6765)

Doctor of Philosophy

January 2020

Department of Mechanical Engineering

THE UNIVERSITY OF MELBOURNE

Submitted in total fulfilment of the requirements for the degree of  
Doctor of Philosophy.

Copyright © 2020 Juan L. Yarmuch

Published articles within this thesis are reproduced with  
permission from their respective copyright holders.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the author.

# Abstract

Mining has played an important role in the growth of civilisation. Migration, and economic and industrial revolutions have been based on the availability of mineral resources. Open-pit mines are surface excavations created to extract valuable material that is located below the surface by means of an excavation. Open-pit mining is commonly used for extracting near to surface metallic and non-metallic ore deposits. Large scale ore deposits can benefit from the economies of scale offered by the open-pit mining process. Open-pit mines are excavated in phases often referred to as pushbacks or cutbacks, which are mineable units designed to maximise the financial return from the mine.

Strategic mine planning is the process that quantifies the economic value of a mining project over the life of the mine. It aims to answer three main questions regarding an ore deposit: 1) what portion of the orebody is both economically and technically feasible to extract; 2) when to extract that portion, or what is the mining sequence and production scheduling; and 3) where to process the extracted material.

Due to the complexity of the decisions involved in mine planning, standard industry practice is to subdivide the main problem into interrelated sub-problems that have been studied in the literature as independent problems. During the last 60 years, optimisation algorithms and computers have been used to assist the design of open-pit mines. However, existing models used to provide guidelines for open-pit designs avoid the complexity of modelling practical and operational conditions inherent of the extraction process. Consequently, most of the output from those models requires significant intervention.

This thesis addresses a key problem in strategic open-pit mine planning which consists of finding the set of pushbacks, their production schedule and the allocation of the material to an appropriate destination that maximises the discounted cash-flows and satisfies both the practical and operational conditions of the pushback design. This thesis explores different mathematical techniques to formally include these practical and operational conditions into mathematical models that can be solved exactly to optimality.



# Declaration

This is to certify that

1. the thesis comprises only my original work towards the PhD except where indicated in the preface;
2. due acknowledgement has been made in the text to all other material used; and
3. the thesis is fewer than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.



# Preface

This thesis presents original work by Juan L. Yarmuch undertaken during his Ph.D. candidature. The use of existing literature and work is made clear within each of those publications. Juan L. Yarmuch did the majority of the work (substantially more than 50%) and is lead-author of all included publications.

Part of the research conducted during this PhD and presented in this thesis, as well as some relevant work completed during the same period, has been published, patented or communicated elsewhere. In addition, results from Chapters 4, 5 and 6 are in preparation to be published.

## Journal articles

- J. Yarmuch, Brazil, M., Rubinstein, H., Thomas, D.A. (2019) “Optimum ramp design in open-pit mines”. *Computers & Operations Research*. doi.org/10.1016/j.cor.2019.06.013.

## Patents

- University of Melbourne. Inventors: Joachim H. Rubinstein and Juan L. Yarmuch (2018) AU Provisional patent ref. no. 2018901873 “*Mine planning method and system*”.
- University of Melbourne. Inventors: Joachim H. Rubinstein and Juan L. Yarmuch (2019) PCT international patent. International publication number WO 2019/227147 A1. “*Mine planning method and system*”.

## Conferences and workshops

- J. Yarmuch, Brazil, M., Rubinstein, H., Thomas, D.A. (2017) “A mathematical formulation for practical pushback design”. 21<sup>st</sup> conference of the International Federation of Operational Research Societies IFORS. Quebec, Canada.

- J. Yarmuch, Brazil, M., Rubinstein, H., Thomas, D.A. (2018) “A mathematical model for practical pushback design optimisation”. Invitation as a speaker at the workshop *Challenges in mine planning* DEPLAMIN. Santa Cruz, Chile.

The following contributions from third parties and existing works are acknowledged here:

- Prof. Maurice Queyranne (University of British Columbia) suggested the use of variable  $v$  in Chapter 5 to keep record of the distance between blocks.

This work was financially supported by the Gilbert Rigg, George Lansell, and Fee Remission Scholarships from the University of Melbourne; and the National Commission for Science Research and Technology (CONICYT-Chile) grant no. 72160405, and FONDEF project IT16M10010.

# Acknowledgements

Many thanks to everyone who have been with me in one way or another during my PhD candidature. You have made this a profoundly fulfilling experience.

I am deeply grateful to my advisory committee Doreen Thomas, Hyam Rubinstein and Marcus Brazil for their unwavering support, guidance and constant encouragement.

Special thanks to the committee chair Chris Manzie for diligent advice throughout the candidature. Also, I would like to thank Jan May and Lyn Buchanan for guiding me through the intricate world of the university rules, procedures and formalities.

In Chile, I am greatly in debt to Rafael Epstein, to whom I consider my mentor and friend, for encouraging me to pursue an academic career. Without his generous help and wise advice, this thesis and endeavour would have not been possible.

This acknowledgement would not be complete without expressing my profound gratitude and admiration to my life partner (and wife) Constanza Gamboa for being -together with Juan Ignacio and Celeste- my endless source of happiness.



*Para Constanza, Juan Ignacio y Celeste.*



# Contents

|          |                                           |           |
|----------|-------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                       | <b>1</b>  |
| 1.1      | Background and motivation . . . . .       | 1         |
| 1.1.1    | Block model . . . . .                     | 3         |
| 1.1.2    | Pit limit . . . . .                       | 4         |
| 1.1.3    | Mining sequence . . . . .                 | 4         |
| 1.1.4    | Pushback design . . . . .                 | 4         |
| 1.1.5    | Haulage road design . . . . .             | 5         |
| 1.1.6    | Mine production scheduling . . . . .      | 6         |
| 1.1.7    | Financial evaluation . . . . .            | 6         |
| 1.2      | Research problem . . . . .                | 6         |
| 1.2.1    | Definitions: . . . . .                    | 6         |
| 1.3      | Aims . . . . .                            | 7         |
| 1.4      | Contribution of the thesis . . . . .      | 8         |
| 1.5      | Structure of the thesis . . . . .         | 10        |
| <b>2</b> | <b>Literature review</b>                  | <b>11</b> |
| 2.1      | Ultimate pit limit . . . . .              | 12        |
| 2.1.1    | Multiple-cones . . . . .                  | 12        |
| 2.1.2    | Exact algorithms . . . . .                | 13        |
| 2.2      | Mining sequence and pushbacks . . . . .   | 15        |
| 2.2.1    | Pushbacks and the final pit . . . . .     | 16        |
| 2.2.2    | Parametrisation methods . . . . .         | 16        |
| 2.2.3    | Resource constrained pits . . . . .       | 17        |
| 2.2.4    | Pushbacks as <i>nested-pits</i> . . . . . | 18        |

|          |                                                                        |           |
|----------|------------------------------------------------------------------------|-----------|
| 2.2.5    | Semi-practical pushbacks . . . . .                                     | 19        |
| 2.2.6    | Ramp design . . . . .                                                  | 21        |
| 2.3      | Open-pit mine production scheduling problem . . . . .                  | 23        |
| 2.4      | Aggregation of blocks . . . . .                                        | 24        |
| <b>3</b> | <b>Optimum ramp design in open pit mines</b>                           | <b>27</b> |
| 3.1      | Optimisation of low stripping roads . . . . .                          | 29        |
| 3.1.1    | Shortest path formulation . . . . .                                    | 29        |
| 3.2      | Optimisation of high stripping ramps . . . . .                         | 31        |
| 3.2.1    | Minimum cost ramp formulation . . . . .                                | 31        |
| 3.3      | Solution method . . . . .                                              | 35        |
| 3.3.1    | Mutually exclusive greedy adaptive path (MEGAP) algorithm . . .        | 35        |
| 3.3.2    | Local search . . . . .                                                 | 36        |
| 3.4      | A case study . . . . .                                                 | 37        |
| 3.4.1    | Ex-pit road . . . . .                                                  | 38        |
| 3.4.2    | In-pit ramp . . . . .                                                  | 40        |
| 3.5      | Conclusion . . . . .                                                   | 42        |
| 3.6      | Summary . . . . .                                                      | 43        |
| <b>4</b> | <b>A mathematical model for the practical pushback design problem</b>  | <b>45</b> |
| 4.1      | The mathematical model . . . . .                                       | 46        |
| 4.2      | Closeness factor . . . . .                                             | 50        |
| 4.3      | Case study . . . . .                                                   | 52        |
| 4.3.1    | Optimal practical pushbacks . . . . .                                  | 53        |
| 4.3.2    | Traditional methodology . . . . .                                      | 58        |
| 4.3.3    | Schedule of aggregated optimised pushbacks . . . . .                   | 62        |
| 4.4      | Conclusions . . . . .                                                  | 63        |
| 4.5      | Summary . . . . .                                                      | 64        |
| <b>5</b> | <b>Two mathematical models for the semi-practical pushback problem</b> | <b>65</b> |
| 5.1      | The problem . . . . .                                                  | 66        |
| 5.2      | The mathematical models . . . . .                                      | 66        |
| 5.2.1    | The mathematical model 1: Connectivity to a ramp . . . . .             | 67        |

|          |                                                                                     |            |
|----------|-------------------------------------------------------------------------------------|------------|
| 5.2.2    | The mathematical model 2: Horizontal network connectivity . . .                     | 72         |
| 5.3      | A computational comparison between models $SPPM1_{BLP}$ and $SPPM2_{BLP}$           | 77         |
| 5.4      | Solution method . . . . .                                                           | 78         |
| 5.4.1    | Early and late start . . . . .                                                      | 78         |
| 5.4.2    | Sliding window heuristic . . . . .                                                  | 78         |
| 5.5      | Computational experiments . . . . .                                                 | 81         |
| 5.5.1    | Sensitivity analysis of parameters $I_{win}$ and $C_{win}$ for instance $I_1$ . .   | 82         |
| 5.5.2    | Larger computational instances . . . . .                                            | 82         |
| 5.6      | Conclusion . . . . .                                                                | 87         |
| 5.7      | Summary . . . . .                                                                   | 89         |
| <b>6</b> | <b>A mathematical formulation for the long-range open-pit mine planning problem</b> | <b>91</b>  |
| 6.1      | The mathematical model . . . . .                                                    | 92         |
| 6.1.1    | Objective function . . . . .                                                        | 96         |
| 6.1.2    | Geotechnical constraints . . . . .                                                  | 96         |
| 6.1.3    | Mining width constraints . . . . .                                                  | 96         |
| 6.1.4    | Connectivity constraints . . . . .                                                  | 97         |
| 6.1.5    | Haulage ramp constraints . . . . .                                                  | 97         |
| 6.2      | Solution method . . . . .                                                           | 98         |
| 6.2.1    | Sliding time window heuristic . . . . .                                             | 98         |
| 6.2.2    | Sampling and refining algorithm . . . . .                                           | 99         |
| 6.3      | Computational experiments . . . . .                                                 | 102        |
| 6.3.1    | Optimisation parameters . . . . .                                                   | 103        |
| 6.3.2    | Results . . . . .                                                                   | 104        |
| 6.4      | Conclusions . . . . .                                                               | 106        |
| 6.5      | Summary . . . . .                                                                   | 107        |
| <b>7</b> | <b>Conclusions</b>                                                                  | <b>109</b> |
| <b>A</b> | <b>PCT patent: Mine Planning Method and System</b>                                  | <b>121</b> |



# List of Figures

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1.1 | Example of an open-pit mine. Chuquicamata copper mine, Chile (source: Google Earth). . . . .                                                                                                                                                                                                                                                                                                                                                                   | 2  |
| 1.2 | Unit operations and basic elements in a mining pushback. Radomiro Tomic copper mine, Chile. . . . .                                                                                                                                                                                                                                                                                                                                                            | 3  |
| 2.1 | Example of 4 macro-blocks each of which is a 2x2x2 reblocking of the original size blocks. . . . .                                                                                                                                                                                                                                                                                                                                                             | 25 |
| 3.1 | Open pit ramp. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                         | 28 |
| 3.2 | Representation of a simple 4 choice of directions graph $\mathcal{G}$ . 1, 2, 3 and 4 represent North, East, South and West directions, respectively. Vertices in white represent incoming directions, while vertices in grey colour outgoing directions. Solid arcs connect all possible incoming directions to all possible outgoing directions at the same cell. Dashed arcs represent the feasible connections from a cell to a neighbouring cell. . . . . | 30 |
| 3.3 | Illustration of the wall slope precedences. Blocks in grey need to be mined before the block in black to ensure slope constraint represented by the two inclined lines that start at the black block. . . . .                                                                                                                                                                                                                                                  | 32 |
| 3.4 | Example of a simple 2D case of $\mathcal{G}_3$ . The vertices in grey illustrate the blocks that can be selected as ramp, the edges in solid black represent the possible connections of the path. The initial ( $S$ ) and final ( $T$ ) vertices are represented in black and the edges from and to those vertices are illustrated with dashed lines. . . . .                                                                                                 | 33 |
| 3.5 | Illustration of Scenario 1 (a) and Scenario 2 (b). The grey zone in Figure (b) represents a forbidden area for the roads. Roads in red represent paths obtained with the high stripping model, while roads in blue show paths obtained with the low stripping model. . . . .                                                                                                                                                                                   | 38 |
| 3.6 | Comparison of the performance (cost and time) of the binary linear programming (BLP) model and the proposed local search method (Local Search). . . . .                                                                                                                                                                                                                                                                                                        | 41 |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |    |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.7 | Illustration of the in-pit ramps. Figures (a), (b) and (c) represent the engineer design, the ramp obtained with the local search method and the ramp obtained solving the BLP model, respectively. Ramps are represented in blue lines, while the stripping associated with the ramps is represented with blue dots. . . . .                                                                                                                                                                                                                                                                                                                                       | 41 |
| 4.1 | Example of a <i>ramp segments template</i> composed of 16 <i>ramp segments</i> with a 1:4 gradient constraint, plan view, considering block 122 as a <i>ramp entrance</i> . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 46 |
| 4.2 | Example of a 45° geotechnical slope upward precedence. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 48 |
| 4.3 | Illustration of the matching factor (plan view). (a) Guidance shells. (b) Manual design. (c) Matching blocks, $g_b = d_b$ and $ g_b - d_b  = 1$ are represented in dark and light grey, respectively. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 52 |
| 4.4 | Modified slope $y^\circ$ after a ramp of width $D$ has been included ([100]). . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 55 |
| 4.5 | Isometric view of the guidance and optimised practical pushbacks, colours represent different pushbacks (a) Guidance pushbacks. (b) Optimised practical pushbacks. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 56 |
| 4.6 | Isometric view of the guidance and optimiser practical pushbacks, colours represent different pushbacks (a) Guidance shells. (b) Manual practical pushbacks. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 61 |
| 5.1 | Example of horizontal rectangular templates, each template is represented as a black rectangle. If the block filled in red is chosen to belong to a pushback then either all blocks marked in black or all blocks marked in grey (or both) must belong to the same pushback. . . . .                                                                                                                                                                                                                                                                                                                                                                                | 67 |
| 5.2 | Illustration of a block precedence relationship to model a wall slope of 45°, assuming cubic blocks. Blocks 1-5 must be mined in order to mine block 6. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 68 |
| 5.3 | Example of a <i>quick &amp; dirty</i> pushback schedule to estimate the value of parameters $\delta_p$ . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 69 |
| 5.4 | (a) Example of the relationship between variables $x$ and $u$ . If the block $b$ (in red) belongs to pushback $p$ , then at least one of the rectangular templates $w$ which contains $b$ must belong to the same pushback; forcing all the blocks within that (those) template(s) (blocks marked in grey and black) to belong to $p$ as well. (b) Illustration of the relationship between variables $v$ , $x$ and $z$ . The variable $v_{bdp}$ is one if there is a path of blocks (marked in grey) that belong to pushback $p$ of length less than or equal to $d$ from the selected coordinate $c$ (marked as black) and the block $b$ (marked as red). . . . . | 70 |
| 5.5 | (a) Example of a set of lines $l$ following directions $\vec{X}$ and $\vec{Y}$ for a level. (b) Example of a simple graph $G_l$ . Arcs from $S_l$ to the blocks and from the blocks to $T_l$ are represented as bowed arrows, while arcs that connect blocks with their immediate neighbours are drawn as straight arrows. . . .                                                                                                                                                                                                                                                                                                                                    | 73 |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |     |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 5.6  | (a) Example of a non-convex shape (black) that cannot be obtained by imposing directions $\vec{X}$ and $\vec{Y}$ simultaneously. The resultant shape when using directions $\vec{X}$ and $\vec{Y}$ adds the section marked in red. (b) Example of disconnected shapes in direction $\vec{XY}$ when only directions $\vec{X}$ and $\vec{Y}$ are considered.                                                                                                                    | 77  |
| 5.7  | Illustration of the SWH with $I_{win} = 3$ (the number of consecutive pushbacks in each integer window) and $C_{win} = 2$ (the number of pushbacks in each continuous window, with the integrality of the variables relaxed), black, green and red colours represent $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ , respectively. . . . .                                                                                                                                    | 79  |
| 5.8  | Mining input data for instances $I_2$ and $I_3$ , where colour scale represents the economic value of the blocks. Plots (a), (c) and (e) are isometric, YZ and XY views of the input data for instance $I_2$ , respectively. Similarly, plots (b), (d) and (f) are isometric, YZ and XY views of the input data for instance $I_3$ , respectively. . . . .                                                                                                                    | 84  |
| 5.9  | Visualisation of the $I_2(2,3)$ plots ((b),(d),(f)) compared to the instance $I_2$ without considering the mining width and connectivity constraints (plots (a),(c),(e)). Plots (a), (c) and (e) are isometric, YZ and XY views of the relaxation of the practical constraints for instance $I_2$ , respectively. Similarly, plots (b), (d) and (f) are isometric, YZ and XY views of the output of instance $I_2(2,3)$ , respectively. . . . .                               | 86  |
| 5.10 | Visualisation of the $I_3(1,2)$ (plots (b),(d),(f)) compared to the instance $I_3$ with a relaxation of the mining width and connectivity constraints (plots (a),(c),(e)). Plots (a), (c) and (e) are isometric, YZ and XY views of the relaxation of the practical constraints for instance $I_3$ , respectively. Similarly, plots (b), (d) and (f) are isometric, YZ and XY views of the output of instance $I_3(1,2)$ , respectively. . . . .                              | 87  |
| 6.1  | Example of the <i>sampling and refining</i> algorithm. (a) blocks marked in grey represent the reduced number of coordinates $\hat{\mathcal{C}}$ at the first iteration of the algorithm. (b) the blocks in red represent the coordinates selected as reference points for the pushbacks. (c) the blocks in grey represent the new set of reduced coordinates which correspond to the neighbours of the coordinates selected as a solution in the previous iteration. . . . . | 99  |
| 6.2  | Blocks within the larger ultimate pit. The colour scale represents the value of the blocks in dollars. Plots (a), (b) and (c) represent isometric, XY and YZ views, respectively. . . . .                                                                                                                                                                                                                                                                                     | 103 |
| 6.3  | Pushbacks found using the <i>sampling and refining</i> algorithm. Colours represent different pushbacks. Figures (a), (b) and (c) show the solutions at different view points for the first, second and third iteration of the algorithm, respectively. . . . .                                                                                                                                                                                                               | 105 |



# List of Tables

|      |                                                                                                                                                      |    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.1  | Minimal paths for scenario 1. . . . .                                                                                                                | 39 |
| 3.2  | Minimal paths for scenario 2. . . . .                                                                                                                | 39 |
| 3.3  | In-pit ramp cost comparison between the BLP model, the local search model and the engineer's design. . . . .                                         | 42 |
| 4.1  | Economic, design and scheduling parameters. . . . .                                                                                                  | 53 |
| 4.2  | Maximum distance parameter $D^*$ for each pushback used in the optimisation. Values showed here are the result of a trial and error process. . . . . | 55 |
| 4.3  | Output for the guidance pushbacks for the case study illustrated in Figure 4.5(a). . . . .                                                           | 56 |
| 4.4  | Material content of optimised practical pushbacks illustrated in Figure 4.5(b). . . . .                                                              | 57 |
| 4.5  | Material mined each year from each optimised practical pushback. . . . .                                                                             | 57 |
| 4.6  | Output of the schedule for optimised practical pushbacks in terms of ore, waste and profit (based on a 12.5% discount rate). . . . .                 | 57 |
| 4.7  | Material content inside each pit shell. . . . .                                                                                                      | 59 |
| 4.8  | Material and profit of each pushback used as pit guidance for the traditional methodology. . . . .                                                   | 60 |
| 4.9  | Material inside each manual practical pushback. . . . .                                                                                              | 60 |
| 4.10 | Material mined each year from each pushback for the manual practical pushbacks. . . . .                                                              | 61 |
| 4.11 | Output of the manual practical pushback schedule in terms of ore, waste and profit (based on a 12.5% discount rate). . . . .                         | 62 |
| 4.12 | Material mined each year from each optimised practical pushback. . . . .                                                                             | 62 |
| 4.13 | Output of the schedule for optimised practical pushbacks in terms of ore, waste and profit (based on a 12.5% annual discount rate). . . . .          | 62 |
| 5.1  | Computational comparison of models $SPPM1_{BLP}$ and $SPPM2_{BLP}$ for instance $I_1$ . . . . .                                                      | 78 |

|     |                                                                                                                                                                                                             |     |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 5.2 | Experiment over parameters $I_{win}$ and $C_{win}$ for instance $I_1$ . . . . .                                                                                                                             | 82  |
| 5.3 | Computational experiments for problem instances $I_2$ and $I_3$ . The values presented in $I_2 Opt^*$ and $I_3 Opt^*$ are the best solutions found by the solver within the time limit of 24 hours. . . . . | 85  |
| 6.1 | <i>Sampling and refining</i> algorithm results. . . . .                                                                                                                                                     | 104 |
| 6.2 | Material sent to process and to waste dump per period. . . . .                                                                                                                                              | 104 |
| 6.3 | Pushback schedule. . . . .                                                                                                                                                                                  | 106 |

# Acronyms & Initialisms

|               |                                                                                   |
|---------------|-----------------------------------------------------------------------------------|
| 2D            | two dimensions.                                                                   |
| 3D            | three dimensions.                                                                 |
| 4D            | four directions.                                                                  |
| BLP           | binary linear program.                                                            |
| BZ            | Bienstock-Zuckerberg algorithm.                                                   |
| CAD           | computer assisted design                                                          |
| CAPEX         | capital expenditure.                                                              |
| CPU           | computer processing unit.                                                         |
| HS            | high stripping ramp.                                                              |
| IBM           | international business machines.                                                  |
| MILP          | mixed integer linear program.                                                     |
| LG-2D         | Lerchs and Grossmann algorithm for the two-dimensional ultimate pit problem.      |
| LG-3D         | Lerchs and Grossmann algorithm for the three-dimensional ultimate pit problem.    |
| LG-UPL        | Lerchs and Grossmann three-dimensional ultimate pit problem.                      |
| LP            | linear programming.                                                               |
| LS            | low stripping roads.                                                              |
| MEGAP         | mutually exclusive greedy adaptive algorithm.                                     |
| MPPSP         | multi-period production scheduling problem.                                       |
| MSRP          | minimum stripping ratio pushback.                                                 |
| NPV           | net present value.                                                                |
| OPEX          | operational expenditure.                                                          |
| OPLRMPP       | open-pit long-range mine planning problem.                                        |
| OPLRMPP-Mod   | open-pit long-range mine planning problem mixed integer linear programming model. |
| OPMPSP        | open-pit mine production and scheduling problem.                                  |
| UPL           | ultimate pit limit.                                                               |
| P.T.          | pronounced turn.                                                                  |
| RAM           | random access memory.                                                             |
| R.T.          | right turn.                                                                       |
| $SPPM1_{BLP}$ | semi-practical pushback binary linear programming model 1.                        |
| $SPPM2_{BLP}$ | semi-practical pushback binary linear programming model 2.                        |

|               |                                                                                                                                 |
|---------------|---------------------------------------------------------------------------------------------------------------------------------|
| $SPPM2_{sub}$ | mixed integer linear programming model for the sub-problem associated to the semi-practical pushback model 2 ( $SPPM2_{BLP}$ ). |
| S.T.          | slight turn.                                                                                                                    |
| STWH          | sliding time window heuristic.                                                                                                  |
| SWH           | sliding window heuristic.                                                                                                       |

# Nomenclature

Note that separate notation has been defined and used in the re-printed published manuscripts included within this thesis.

$b \in \mathcal{B}$  : blocks in the block model  $1, \dots, B$ .

$c \in \mathcal{C}$  : set of XY-coordinates that act as a reference point to ensure the connectivity of the blocks within a pushback. A priori the set  $\mathcal{C}$  is formed by the XY-projection of the  $(x, y, z)$  coordinates associated to all blocks  $b \in \mathcal{B}$ .

$t \in \mathcal{T}$  : periods  $1, \dots, T$ , where  $T$  is the maximum period considered for the schedule.

$p \in \mathcal{P}$  : pushbacks  $1, \dots, P$ .  $P$  correspond to the maximum number of pushbacks to be considered.

$d \in \mathcal{D}_p$  : set of distances from the reference point associated to pushback  $p$ .  $0, \dots, D$ .

$\hat{b} \in \hat{\mathcal{B}}_b$  : vertical precedences of block  $b$ .

$l \in \mathcal{L}$  : levels in the block model;  $1, \dots, L$ .

$\check{b} \in \check{\mathcal{B}}_l$  : blocks that belong to a level  $l$ .

$\breve{b} \in \breve{\mathcal{B}}_b$  : blocks that share the same level  $l$  with block  $b$ .

$b_s^+ \in \mathcal{B}_s^+$  : set of blocks immediately above blocks in ramp segment  $s$ .

$b_s^- \in \mathcal{B}_s^-$  : set of blocks immediately below blocks in ramp segment  $s$ .

$s' \in \mathcal{S}'_b$  : ramp segments that start at block  $b$ . The block  $b$  act as a *ramp entrance* of the segments  $s' \in \mathcal{S}'_b$ .

$s'' \in \mathcal{S}''_b$  : ramp segments that end at block  $b$ .

$s \in \mathcal{S}$  : all ramp segments ( $\mathcal{S} = \bigcup_{b \in \mathcal{B}} \mathcal{S}'_b$ ).

$\bar{b} \in \bar{\mathcal{B}}_b$  : set of horizontal adjacent blocks of block  $b$ .

$w \in \mathcal{W}$  : set of mining width rectangular templates.

$w' \in \mathcal{W}'_b$  : set of mining width rectangular templates that contain block  $b$ .

$b' \in \mathcal{B}'_w$  : set of blocks contained in the mining width rectangular template  $w$ .

$V'_b$  : economic value per ton [US\$/ton] of block  $b$  sent to process.

$V''_b$  : economic value per ton [US\$/ton] of block  $b$  sent to waste dump.

$W_b$  : tonnage of block  $b$  [tonnes].

$\overline{W}_t^{ML}$  : maximum mining limit [tonnes] for period  $t$ .

$\overline{W}_t^{PL}$  : maximum processing limit [tonnes] for period  $t$ .

$\underline{W}_t^{ML}$  : minimum mining limit [tonnes] for period  $t$ .

$\underline{W}_t^{PL}$  : minimum processing limit [tonnes] for period  $t$ .

$T_p^{ini}$  : minimum extracting period for pushback  $p$ .

$T_p^{fin}$  : maximum mining period for pushback  $p$ .

$\delta_t$  : discount factor for period  $t$ .  $\delta_t = (1 + \zeta)^{-t}$ , where  $\zeta$  is the discount rate.

$D_{b\check{b}}$  : Euclidean distance from block  $b$  to block  $\check{b}$  [metres].

$D_p^*$  : maximum distance from the ramp entrance to every block at the same level as the ramp entrance that belongs to pushback  $p$  [metres].

$V_b$  : net profit of block  $b$ , this will be negative for any block that is waste [dollars].

$\underline{W}_p$  : minimum tonnage of pushback  $p$  [tonnes].

$\overline{W}_p$  : maximum tonnage of pushback  $p$  [tonnes].

$\lambda_b$  : compactness factor of pushback  $p$ .

$\delta_p$  : value discount factor for pushback  $p$ .

$\mathcal{F}$  : a mapping function  $\mathcal{F} : \mathcal{B} \rightarrow \mathcal{C}$  such that  $\mathcal{F}(b) = c$  is the 2D projection over the XY plane of the spatial location of block  $b$ .

# Chapter 1

## Introduction

**M**INING has played an important role in the growth of civilisation. Migration, and economic and industrial revolutions have been based on the availability of mineral resources. The Victorian gold-rush (1850-1860) which led the development of what was once called *Marvellous Melbourne* is a palpable example of this. Today, mining is still one of the most important sectors for the Australian economy. In 2017-18, minerals and resources accounted for 55% of the total exports in Australia, making up 8% of Australia's economy and employing more than 250,000 people ([www.industry.gov.au](http://www.industry.gov.au)).

### 1.1 Background and motivation

Surface mining is a category of mining that differs from underground mining in which surface mining requires the removal of the material overlying the mineral. The most common methods among surface mining are open-pit mining, strip-mining and quarry mining. Open-pit mining is different from quarry mining in that open-pit mines selectively extract the ore, while quarries extract aggregates or dimensional stone products, such as granite and marble. The main difference between open-pit mining and strip-mining is how the waste material is disposed. In open-pit mines, the waste material is generally disposed outside the pit, while in strip-mining the waste material is disposed inside the pit. Most of the surface coal mines are strip mines.

Open-pit mines are surface excavations created to extract valuable material (ore) that is located below the surface by means of an excavation or cut. The ore body is mined from top-down in a series of benches (horizontal layers). Mining starts by excavating a level that is accessible with for topography and after a sufficient area has been exposed the mining of the following level begins. This process continues until the mine outline is reached [48]. Ore is usually mined together with a considerable amount of material with little or no value (waste) which must be extracted before reaching the ore; this process is known as stripping. The ore is processed at the processing plants or it is stockpiled, while the waste material is disposed of in dedicated areas commonly called waste dumps. Figure 1.1 illustrates the general layout of an open-pit mine.

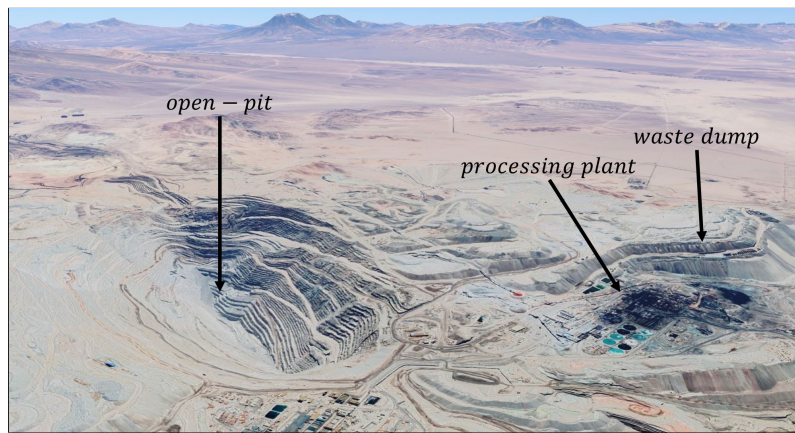


Figure 1.1: Example of an open-pit mine. Chuquicamata copper mine, Chile (source: Google Earth).

Open-pit mining is commonly used for extracting near to surface metallic and non-metallic ore deposits. Large scale ore deposits can benefit from the economies of scale offered by the open-pit mining process [98]. Open-pit mines are excavated in phases often referred to as pushbacks or cutbacks, which are mineable units designed to maximise the financial return from the mine. Pushbacks are volumes with sufficient bench (see Figure 1.2) area to allow the safe operation of the equipment in the mining process.

The mining unit operations are the basic steps necessary for the extraction of the material. In open-pit mines, the most common unit operations are: drilling, blasting, loading and hauling. In general, these unit operations occur simultaneously in the same bench. Benches can be divided into working (or active) benches and inactive benches. Active benches are in the process of being excavated, whereas inactive benches are left to ensure the pit-wall slope stability. To access different benches in-pit haul roads or ramps are created. Thus open pit mines require a comprehensive road network which connects the extraction sectors inside the pit to the possible destinations outside the pit. Figure 1.2 illustrates the main units operations and the basic features of a pushback.

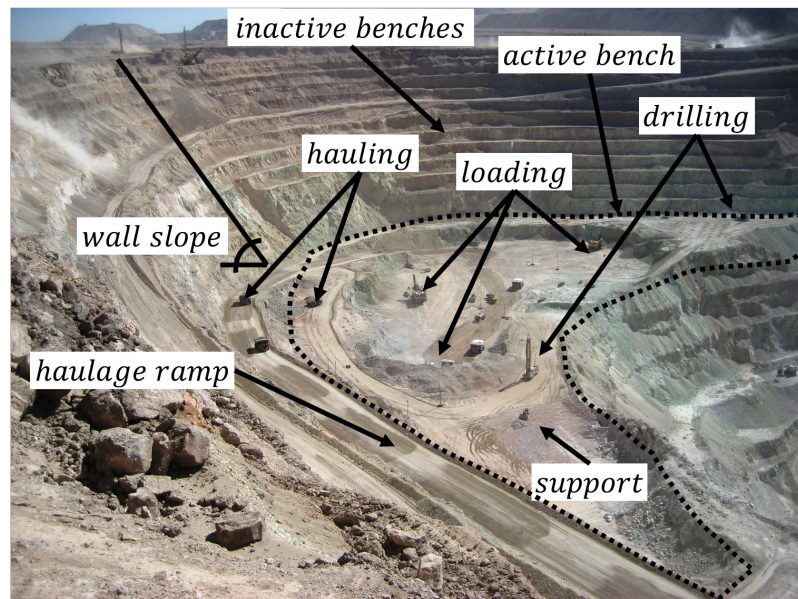


Figure 1.2: Unit operations and basic elements in a mining pushback. Radomiro Tomic copper mine, Chile.

In general, open-pit long-range (strategic) mine planning is the process that quantifies the economic value of a mining project over the life of the mine. It aims to answer three main questions regarding an ore deposit: 1) what portion of the orebody is both economically and technically feasible to extract? (also known as the pit limit), 2) when to extract that portion, or what is the mining sequence and production scheduling?, and 3) where to process the extracted material? Mine planning also produces a cost-benefit analysis of the ore extraction and the infrastructure required for the development of the project taking into account the interaction between the economic-performance, safety, availability of labour and the mining equipment [98]. Due to the complexity of the decisions involved in mine planning, standard industry practice is to subdivide the main problem into the following sequential stages which are solved independently: 1) pit limit definition, 2) mining sequence optimisation, 3) pushback, dump and road design, 4) mine production scheduling, 5) mining fleet requirement estimation, and 6) operational and capital expenditure calculation.

### 1.1.1 Block model

Geological exploration and modelling is the first stage of the mine evaluation process. Once the geological, geo-technical and metallurgical models have been completed, technical and economic studies are conducted before an open pit mine begins development and production.

The geological model is typically represented as a 3D regular grid where each cell is called a block. Each block is associated with information about its spatial location and

a number of attributes regarding the geology of the mineral deposit such as the valuable material content, contaminants, specific gravity, rock type, geotechnical domain and profit amongst others. The number of blocks in a typical block model of a large open-pit mine is between 500,000 and 10,000,000.

### 1.1.2 Pit limit

The ultimate pit limit (UPL) is the region of the geological deposit that is technically and economically viable to mine. In the 60's Lerchs and Grossmann (see [60]) introduced an efficient optimisation algorithm to define the UPL by maximising the un-discounted cash-flows of the excavation, subject to satisfying geotechnical constraints on the slopes of the pit walls (slope constraints). The PL problem (as described in [60]) is a static problem that simplifies the mining process to that of deciding which material to extract over the life of the mine. Time decisions and production capacity constraints are not taken into account. The Lerchs and Grossmann approach and algorithm is the prevailing method for solving the UPL problem. However, recent approaches that use maximum flow algorithms for the UPL problem have been shown to outperform the Lerchs and Grossmann algorithm (see [46]).

### 1.1.3 Mining sequence

Taking advantage of the efficiency of the Lerchs and Grossmann algorithm, a common approach to find a mining sequence consists of iteratively generating a set of ultimate pit limits where the value of each block is weighted. This approach is based on the suggested *parametric analysis* (see [60]) which was later formalised by G. Matheron in 1975 [100]. Thus, the output of the parametric analysis is a set of *nested-pits* which gives a sequence from the most to the least profitable regions inside the ultimate pit limit, when weights between 0 and 1 are used.

The parametrisation method is simple and fast. However, in most cases, shells obtained with this parametric analysis are not practical to use directly as pushbacks. There are three major issues with the nested-pits: the big difference in volume between two adjacent shells, also known as the *gap problem* (see [80]), the disconnected regions within the same pit shell and the lack of an operational mining width between shells. In practice, the set of shells needs to be manually modified by mining engineers to incorporate these operational conditions using CAD tools [29].

### 1.1.4 Pushback design

Engineers mainly consider three operational conditions when they design practical pushbacks. First, every pushback must satisfy a minimum width to allow safe operation for

the mining equipment. Second, all the regions of a pushback must be connected. Third, every pushback needs to have at least one haulage ramp to access the pushback from top to bottom.

Current mine planning software and capabilities are not adequate to achieve optimal designs which simultaneously satisfy operational constraints and maximize the net present value (NPV). The current methodology for designing pushbacks relies on quality manual interpretation of the nested pits, which might provide some guidance to optimise theoretical NPV, but cannot give practical designs without considerable and subjective judgement from an experienced mine designer. Pushbacks are later scheduled to produce the mine production schedules.

### 1.1.5 Haulage road design

In-pit ramps are roads that connect the working phases to the pit exits; these require the removal of a great amount of material for their construction. The total tonnage of ore and waste and the pit shape can change dramatically after the addition of the in-pit ramps. Ex-pit roads, on the other hand, connect pit exits to processing plants or dumps, and their construction usually requires removing much less material than in-pit ramps.

The ramp design process begins with the determining the pit exits, the ramp width and gradient. Pit exits are usually located close to the processing plants or waste dumps to reduce the total haulage cost, while, the ramp width is defined by considering operational factors, safety conditions and the stripping cost. Examples of operational factors are the size of the haulage equipment, the number of traffic lanes, the traffic density and safety elements (centre line piles, road edge barriers and ditches). A comprehensive description of these factors can be found in [8]. As a rule of thumb, the minimum ramp width should be at least four times the width of the hauling equipment. The ramp gradient is constrained by a maximum gradient determined by diverse factors such as the type of haulage trucks, the mining operation conditions and the road surface quality. Mining trucks work best when the maximum ramp gradient is between 8% and 10% [95].

Once the design parameters have been defined, engineers must decide whether to use spiral ramps, switchbacks (180° turns) or a combination of both. Because ramps can be used as catch berms to control local rock failures, a design that includes switchbacks is generally more stable, in geotechnical terms, than spiral ramps. However, a significant number of switchbacks over the same wall increases the amount of stripping of the pit, compromising the profitability of the mine. Also, switchbacks affect the cost and safety of the haulage truck's operation [95]. For instance, switchbacks reduce the truck speed and consequently the truck productivity, they require periodic maintenance increasing the operational cost, and they generate visual problems for drivers increasing the risk of collision. Nowadays, most mining operations rely on their engineers for the haulage road design, and they mostly do this work manually by trial-and-error.

### 1.1.6 Mine production scheduling

After the pushback design stage, scheduling and cut-off grade optimisation (classifying each block as either waste or ore) is done in the next stage. Other constraints such as, production targets and blending constraints over time are included. Long-range planning takes into account the pushbacks and their level-by-level subdivisions, known as benches, as the minimum mineable units. Thus, the pushback design is the key component in determining the optimal schedule. Several optimisation software packages have been developed to assist engineers in this task. Software like Comet, Geovia Whittle (Dassault), Vulcan Chronos (Maptek), Minemax Scheduler, and MineSight Economic Planner (Mintec) are commonly used in the mining industry for mine production scheduling optimisation.

### 1.1.7 Financial evaluation

The mine schedule gives details of the mining sequence as well as the physical destination of the material mined. The mine equipment requirement is now calculated based on the production rates and the haulage distances (fifth stage). Finally, a purchase plan of mining equipment is computed (CAPEX), and the operational cost (OPEX) is calculated taking into account the labour costs, consumables, equipment maintenance, shift roster and other factors [102].

## 1.2 Research problem

The majority of the models used to describe the pushback design problem avoid the complexity of modelling real operational constraints. Consequently, the output from these models generally requires significant intervention by mining engineers. At present, there is a clear gap between the literature and the pushback design problem. Moreover, even though this is an old problem, there is no consensus in defining what a pushback is. The key missing parts in the literature will be highlighted in Chapter 2.

### 1.2.1 Definitions:

The following definition of pushbacks, which will be used in this thesis, is given to avoid confusion with other definitions from the literature.

- **Pushbacks:** Pushbacks are a set of disjoint, mineable volumes aimed to maximise the financial return from a mine. The union of the pushbacks forms the ultimate pit. Each pushback is a connected volume (aggregation of blocks) with sufficient

operational width, and as such the slope conditions are honoured. Each pushback is designed with a haul-road that connects all the benches from top to bottom.

- **Semi-practical pushback:** A pushback such that the haul-road condition is relaxed.

This thesis addresses a key problem in strategic open-pit mine planning which consists of finding the set of pushbacks, their production schedule and the allocation of the material to an appropriate destination that maximises the discounted cash-flows and satisfies the operational conditions of the pushback design (this problem is hereafter called the open-pit long-range mine planning problem, or the OPLRMPP). The OPLRMPP is strategic, and so the time horizon considered in this research is the life of the mine. The solution of the OPLRMPP defines the set of pushbacks and, for each period, determines the extracted blocks and their destination.

The OPLRMPP, described in this thesis, contains the following assumptions and considerations:

- All parameters are deterministic and there is no stochastic element in the model.
- The minimum amount of time for scheduling purposes is 1 year.
- No stockpiling/re-handling will be incorporated into the mine production schedule.
- The block model, mining and processing capacities, economic and technical parameters for pushback design and scheduling are given.
- The ramp system and mining width considered assume a truck and shovel operation.

## 1.3 Aims

This research aims to develop a new model for the open-pit long-range mine planning problem by considering operational conditions and proposing decomposition methods and heuristics to solve large scale problems in a reasonable time. The aims of the research are as follows:

1. Develop a mixed integer linear programming model to generate pushback designs that take the main operational factors into consideration. Associated research questions are:
  - (a) Which algorithms are suitable for solving the OPLRMPP?
  - (b) If it's not possible to model all the operational features of pushbacks in a single mathematical model, should separate models be created?

- (c) How do we incorporate shape control and haulage ramp design into the pushback design model?
2. Implement and analyse methods to solve this model, ranging from exact solutions for a small problem to heuristic approaches or approaches with partially relaxed constraints for large scale problems. Associated research questions are:
  - (a) Does the model produce an operationally feasible solution and guarantee optimality for the pushback problem?
  - (b) Could that model be implemented and solve large-scale problems?
3. Make a comparison of the new model and its results with current practice. Associated research questions are:
  - (a) How much value can be added by the new model compared with traditional methodology based on nested pits?

## 1.4 Contribution of the thesis

This thesis makes the following contributions:

Chapter 2:

- A comprehensive literature survey about the ultimate pit limit problem and the mining sequencing (pushback design) problem is provided, and
- A classification of various definitions of a pushback from the literature is given.
- Different approaches to find the *next best ore* such as the mine production scheduling and the pushback problem are analysed.

Chapter 3:

- A shortest path formulation, which can be efficiently solved, is adapted to solve the ex-pit road design problem.
- A mixed integer linear programming model to solve the in-pit ramp design problem is introduced.
- A heuristic approach to solve realistically sized instances -within a reasonable time scale- of the high stripping ramp design problem is developed
- A set of computational experiments based on real mine data is presented.

Chapter 4:

- A formal description of the practical pushback problem is given.

- A binary linear programming model for the practical pushback problem is introduced.
- A suitable aggregation method is used to solve realistically sized problems (with millions of blocks).
- A *closeness factor* is introduced to measure the similarity between a computer-generated solution and a corresponding manual design that includes practical constraints.
- Real mine data is used to compare the use of the new formulation to the traditional methodology based on nested-pits.

#### Chapter 5:

- Two binary linear programming models for the semi-practical pushback problem are introduced. Both models explicitly account for the connectivity and mining width conditions as linear constraints.
- A variable reduction pre-processing routine and an adaptation of the Sliding Time Window Heuristic are implemented to solve medium size problem instances.
- Real mine data and two instances from the Minelib mining problem library are used to test the models and heuristics.

#### Chapter 6:

- A mixed integer linear programming (MILP) model for the OPLRMPP is developed. The model is aimed to maximise the time-discounted cash-flow of the mine (also known as NPV). The model is capable of generating a mine production schedule that satisfies the processing and mining capacities, as well as the cut-off grade definition.
- The mathematical model incorporates the connectivity, mining width and haulage ramp conditions as linear constraints.
- A series of pre-processing routines, model simplification and heuristics are implemented to solve medium size problem instances.
- Real mine data is considered to validate the model.

#### Appendix A:

- A binary linear programming (BLP) model for semi-practical pushbacks, and a BLP model for the mine production schedule are introduced.
- The mathematical models approach the connectivity and mining width conditions by using physical behaviour of soap bubbles.
- Traditional algorithms in mine planning (such as LG-3D[60] and BZ[12]) can be used to solve large scale problem instances for the new mathematical formulations.

## 1.5 Structure of the thesis

This introduction forms Chapter 1 of the thesis, and the remaining five chapters are structured as follows:

Appropriate methods and approaches available in the literature for the open-pit long-range mine planning problem (OPLRMPP) are introduced in Chapter 2, as well as providing a more thorough motivation for our interest in the addition of geometrical and operational conditions to the OPLRMPP. A review is also given of the models, methods and approaches for optimising pushbacks and mine production schedules.

In Chapter 3, the problem of adding a haulage ramp to a given semi-practical pushback is studied. Additionally, the problem of finding the minimum cost ex-pit haulage road is addressed. This chapter is based on a published manuscript.

Chapter 4 explores the use of the in-pit ramps as an element to control the shape of a pushback; and thus generate more operationally feasible designs. A mathematical model that simultaneously optimises a pushback and the correspondent in-pit ramp is introduced. The output of the proposed model is compared to the traditional approach based on nested-pits to measure the value added by the new model.

Chapter 5 describes two approaches aimed at incorporating some of the operational conditions -block connectivity and a minimum mining width- into a single model. Both models generate operationally feasible designs that relax the in-pit ramp condition. The use of heuristic algorithms allows medium-size problem instances to be solved close to optimality in a reasonable time.

Chapter 6 considers the question of how it is possible to incorporate the main operational conditions for the OPLRMPP into a single model; and also simplifications and heuristics to solve real mine datasets. This work introduces a unique mixed integer linear programming (MILP) model for the OPLRMPP that uses some of the techniques presented in Chapters 4 and 5 to model the operational conditions. This new formulation answers the questions of what and when to extract, and where to process.

Chapter 7 draws the conclusions from Chapters 3-6 together, summarises the findings and discusses future work.

Finally, in Appendix A a patented method that considers the physical properties of soap bubbles to model block connectivity and operational width conditions is presented. New models for semi-practical pushbacks and mine production scheduling can be solved for large problem instances (consisting of millions of blocks) by using well known algorithms in mine planning.

## Chapter 2

# Literature review

*Anyone who is not familiar with many of these [optimisation techniques in open-pit mine planning] will be bewildered by all the articles in the literature, each claiming to either be in the process of solving the problem or to have solved the problem.*

—Young C. Kim and Yixian Zhao [54]

THE main task of open-pit long-range mine planning is to define the strategy that maximises the financial return of the extraction of a mineral deposit while satisfying -among others- technical, operational, legal and financial conditions. This task is completed by determining which portion of the geological resource (orebody) is technically and economically viable for extraction -the ore considered within this portion constitutes the mining reserves. The financial return of a mine is usually measured by net present value (NPV), which is the summation of the time-discounted cash-flows. To estimate the cash-flows of a mine -and consequently the NPV- it is necessary to develop a production schedule of the mine.

Production planning is a complex process that takes into account a variety of aspects such as the geometry of the orebody, the processing and mining capacities, the interaction of the unit operations and types of equipment, and the production targets. The mine production schedule defines the cash-flows which are essential in achieving the highest financial return of the mine [98]. Pushbacks are the foundation for the production schedule and play a crucial role in defining annual cash-flows [40]. Realistic pushback designs -those that account for enough operational area and access roads- are fundamental for a proper evaluation of a mine. Production plans not based on realistic designs typically underestimate the waste and overestimate the ore material, this often leads to wrong decisions in project evaluation [29]. The final pit outline is the resultant pit after the mining of the scheduled ore and waste. Computing the final pit is required to estimate the mining reserves [86].

As Kim and Zhao stated, the monolithic approach to -what we called- open-pit long-range mine planning problem (OPLRMPP) has not previously been considered in the literature due to the size and complexity of various mine planning aspects [54]. This observation is still valid today. Instead, three interrelated components of the OPLRMPP have been studied in the literature as independent problems: 1) the definition of the

final pit or the ultimate pit limit (UPL) problem, 2) finding the mining sequence, or the pushback design problem, and 3) determination of the production planning or the open-pit mine production scheduling problem (also referred as OPMPSP).

## 2.1 Ultimate pit limit

Early work in open-pit mine planning -from 1960 to 1990- primarily focussed on finding methods to define the ultimate pit limit. Before the use of computers, the final pit outline was defined by hand over a series of 2D cross-sections. Each of these must at least have information regarding the geological inventory and the surface topography. Then, a pit boundary was locally defined for each section by moving the limits of the pit until the calculated *stripping ratio* (amount of waste divided by the amount of ore) reached a break-even ratio, previously defined by the practitioner. This process was done by hand in a trial-and-error process. Finally, the interpolation (or smoothing) of the 2D pit boundaries resulted in the final pit (for a detailed explanation of the method, see [56]). Although the hand-method was useful in the early days of mine planning, the method suffered from two main weaknesses: it was error-prone, and it was a time-consuming process [18]. For example, according to Robb [86], the time required for a team of engineers involved in the detailed design of a large open pit was between 12 and 14 person-months.

Contrary to modern practice, the use of regular-sized block models to represent the geological inventory was not standard practice in the early days. Instead, continuous representation of the orebody and the use of irregular-sized blocks were the preferred choices. The main reasons for using the latter over the former were to reduce the storage space and to minimise the number of calculations in planning tasks such as the final pit calculation. Therefore, some of the first computer algorithms developed to assist mine planning assumed continuous or irregular-sized block representations of the orebody (for example, see [18] [75], [68] and [58]).

### 2.1.1 Multiple-cones

From an historical point of view, the computer program developed by Kennecott Copper Corp. in 1963 was arguably the first successful case of the use of computers to assist open-pit mine-planning. This computer program allowed users to compute a final pit automatically. The core of the program was based on the multiple-cone (also known as a floating cone) algorithm developed by Milton Pana, which simulates the extraction of multiple cones and analyses the economic potential of the incremental removal of each cone, keeping track of the contribution to the total net value of the pit. The algorithm stops when the sum of the net value of all the increments reaches a maximum. Although the computer program was initially aimed to assist the economic evaluation of new projects, it was later modified and successfully implemented in Kennecott's Chino Mines and Bingham mine [18].

The use of multiple-cone techniques to compute final pits and mining sequences rapidly gained popularity within the mining industry, replacing the hand methods. The main advantages of the use of multiple-cones are ease of implementation; low memory requirements and a reduced number of calculations crucial for early-computer systems; the use of cones allowed the modelling of a variety of wall slope conditions and minimum areas for mining equipment (mining width).

However, the resultant final pit obtained using this method depends on the order in which the incremental cones have been considered over the geological inventory. The reason for this variability is that the proposed algorithm cannot adequately accommodate the cost of the waste material in the intersection of overlapping cones. In the following decades, researchers proposed new cone-based methods to deal with overlapping cones. For example, Lemieux [59] introduced an optimising floating cone algorithm that combines two algorithms (a positive floating cone and negative floating cone algorithm). The optimising algorithm uses the results from the positive and a negative cone algorithms to identify the overlapping cones, and then reallocates the costs of the waste associated with the overlapping cones (see [58] for details). Also, Dowd and Onur [28] proposed an alternative floating cone algorithm, which modifies the algorithm previously introduced by Korobov (see [26]) by introducing a rule that redefines the order in which the cones should be evaluated, starting with the cones that do not overlap and then evaluating the overlapping cones. Dowd and Onur claim that the resulting algorithm finds the optimal final pit. However, no rigorous proof of optimality has been provided for this or any other floating cone algorithm mentioned here.

### 2.1.2 Exact algorithms

In 1965 a couple of researchers from IBM, Lerchs and Grossmann [60], introduced two algorithms; one of them later becoming the basis of modern mine planning software. The first algorithm consisted of a dynamic programming method for the optimal final pit in 2-D cross-sections (LG-2D algorithm), and the second corresponded with a graph-theoretic algorithm for finding the final pit in 3-D (LG-3D algorithm). The problem formulated in the paper (hereafter called the LG-UPL problem) assumed the geological inventory is represented by regular blocks; the profits, grades and density is known for each block; the profit is independent of the mining sequence; and the only operational consideration is the wall slope condition, which is imposed by the relation between the blocks (precedence relationship). The authors provided mathematical proofs of optimality and convergence for the graph-theoretic algorithm. However, this 3-D algorithm could not be efficiently implemented for real-size mines until the mid-eighties. The main reasons were the amount of computer memory, and the number of calculations required exceeded the capacities of the early-computer systems [4].

Several authors continued working on improvements to the Lerchs and Grossmann algorithms. For example, Johnson and Sharp [51] proposed a method for the LG-UPL problem that repeatedly uses the LG-2D algorithm over cross-sections, given it is required to mine to a particular level, and then a final application on a longitudinal section

where the blocks represent the optimal section contours down to each level. According to Johnson and Sharp, the implementation of the algorithm allowed solution of mines up to 100,000 blocks, but the largest example provided consisted of 23,010 blocks taking 1,457 seconds to run. However, the algorithm cannot find the optimal pit for the 3D problem in all cases.

Also, Lipkewich and Borgman [61], introduced a procedure called “layer-cake” which reduced the memory requirement of the LG-3D algorithm by solving the LG-3D algorithm over a sub-graph of the complete model. The LG-3D algorithm is run from the top bench, and the remaining graph is augmented with the following layer of data, removing from the memory the blocks within the partial solutions before moving to the next bench, until the bottom bench is reached. The program accepted instances of up to 12,000 blocks because of the computer system limitations. Additionally, Caccetta and Giannini [16] proposed the use of dynamic programming to produce a “bound” for the optimal pit limit and, thus, reduced the complexity of the problem by running the LG-3D algorithm over the blocks within the bound.

Another example is the algorithm introduced by Zhao and Kim [104], which modifies the LG-3D algorithm by operating over directed trees based on the ore-waste relationship. The algorithm presented redefined the original concepts introduced by Lerchs and Grossmann making it easier to understand. Besides, the new algorithm considers a series of preprocessing routines that reduce the size of the memory requirements and increases the speed compared with the LG-3D algorithm. The largest instance tested consisted of 256,000 blocks (52,325 after preprocessing) taking 115 minutes to run.

Other methods, such as linear programming and network flows, have been proposed to solve the LG-UPL problem. Meyer [68] proposed a linear programming model that considered columns instead of blocks to reduce the number of variables. However, the model introduced is constrained to vein-shape ore zones. Picard [76] presented a transformation of the LG-UPL problem to a network flow problem (over a transformed graph). The solution of the LG-UPL problem corresponds to the minimum-cut of the transformed graph which can be found using a variety of max-flow min-cut algorithms such as the Sleator-Tarjan, the Push-relabel [46] and the Pseudo-flow [47] algorithm. Underwood and Tolwinski [96] present a network flow algorithm based on the dual of the LG-UPL problem. The authors provide a theoretical insight of the LG-3D algorithm from a mathematical programming viewpoint. Another interesting approach is the *parametrisation method* proposed by G. Matheron (described in [36]), which replaces the search for the final pit with the determination of a parametrising function which particularly can find different iso-value pits.

In the mid-80's Alford and Whittle [4] gave an efficient implementation of the LG-3D algorithm. The largest instance reported consisted of 500,000 blocks of 5x5x5 m size, taking 100 hours to run and 100 megabytes of disk space; the same instance required 3 hours of computer time after regularisation (aggregation) into 30,000 aggregated blocks of 10x10x10 m size. No details of the preprocessing methods considered or the implementation were provided due to commercial interest. In the following years, Whittle's implementation of the LG-3D algorithm became the industry standard for computing the

ultimate pit (now known as Geovia Whittle); and it has been described as one of the most efficient implementations of the LG-3D algorithm [46].

More efficient algorithms combined with advances in computer systems now allow the solution of the LG-UPL problem for mines containing millions of blocks in just a few minutes. For example, recent studies have shown that the pseudo-flow algorithm outperforms the LG-3D algorithm (see [71]). For instance, Geovia Whittle has reported a time-reduction from 15 hours to 12 minutes for a mine instance with 21.3 million blocks [9]. Other software providers, such as Datamine<sup>1</sup> and Deswik [78], have also opted for the pseudo-flow algorithm as the optimisation engine for the LG-UPL problem.

Despite significant advances in solving the LG-UPL problem to optimality, the solutions will not necessarily result in practical designs because of the simplifications and assumptions of the LG-UPL problem formulation. The addition of further operational constraints such as mining widths and haulage roads is usually done by hand over the optimised shape, which in some cases significantly degrades the value of the solution [55], [28] and [43]. Moreover, other assumptions such as maximising un-discounted cash-flows, assuming profits independent of the mining sequences, considering unlimited mining and processing capacities pose questions of the validity of the optimal solution in “real” terms [43] and [28].

## 2.2 Mining sequence and pushbacks

The ultimate pit limit gives the final contour of the mine, but there are multiple mining sequences that end in the same final pit. Different mining sequences may lead to different mine plans which, in turn, may produce different discounted cash-flows or NPV ([60], [101] and [103]). Although the concept of pushback has not been unequivocally defined, some key features can be extracted from the multiple definitions existing in the literature. Pushbacks form a sequence of mineable pits that contain enough material to support production ([86],[80],[30], [66] and [64]), with enough working area ([48], [23], [86], [103], [84], [22], and [37]), and with haul road access ([48], [40], [84], [22] and [37]) aimed at maximising the NPV of the mine.

Pushbacks are the basis of production plans. The use of non-mineable pushbacks for production planning and project evaluation may lead to wrong decisions in feasibility studies [29]. The size of equipment is crucial to the definition of the minimum working area for the pushbacks [86], and their production rates are inter-related with the shape and size of the pushbacks [6].

---

<sup>1</sup><https://www.dataminesoftware.com/product/npv-scheduler/> consulted on the 29/11/2019

### 2.2.1 Pushbacks and the final pit

The study of the mining sequence is an old problem in mine planning. In some cases, methods for finding a mining sequence have been proposed in articles primarily focussed on the ultimate pit limit problem.

Lerchs and Grossmann [60] proposed a parametric analysis to design intermediate pits (mining sequence). The idea consisted of obtaining pits with a limited volume that represented, for example, one year of production. Robb [86] proposed a heuristic method based on floating cones to design pushbacks that are formed as a series of incremental cones. Alford and Whittle [4] proposed an algorithm that computes the ultimate pit over a partial selection of the block model. At each step the algorithm runs a pit limit adding a bench to the block model from top to bottom until the bottom-most level is reached (this method was later referred to as *vertical bounding* [100]). The Matheron parametrisation method for the final pit can also be used to deliver a set of nested pit shells if different parameter values are considered [36].

### 2.2.2 Parametrisation methods

The main principle behind the parametrisation method is to weight the value of the blocks in order to obtain smaller pits when an ultimate pit algorithm is computed over the weighted-value blocks. Different parametrisation methods will drive different mining sequences. Thus, the challenge is to properly define a weighting procedure such that the pits obtained (or the mining sequence) lead to the best possible NPV. Probably, one of the best-known parametrisation techniques is the *revenue factor* method introduced by Whittle and Rozman [101]. The method consists of iteratively computing the LG-3D algorithm over a series of revenue-factor value block models. The output of this method is a set of “*nested pits*” which gives a sequence from the most to the least profitable pit, when factors between 0.0 and 1.0 are considered (see [100]).

Another -less common- parametrisation approach, known as the *mining direction method* [100], consists of weighting the value of the blocks depending on their spatial location and a given starting point and direction. Poblete et al. [77] give an example of the mining direction method applied to a copper deposit. Ramazan and Dagdelen [82] introduced an algorithm to find a series of minimum stripping ratio pushbacks (MSRP). The authors claim that the algorithm is capable of producing the optimal MSRP. According to the authors, the result of the proposed algorithm can be replicated by the *revenue factor* method if the same grade value (maximum grade in the block model) is assigned to all the ore blocks.

Other authors have proposed methods to incorporate geological uncertainty into the pushbacks obtained through the parametrisation method. For example, Gholamnejad and Osanloo [40] propose the addition of a penalisation factor -associated with the geological variability- to the value of the blocks before using the *revenue factor* method to find a sequence of nested-pits. Thus, according to the authors, the obtained sequence will ac-

count for the geological uncertainty. Dimitrakopoulos, Martinez, and Ramazan [27] use the revenue factor parametrisation method over a series of simulated orebodies to account for geological uncertainty. For each set of nested-pits (each one associated with an orebody simulation) a mine production schedule is obtained. Finally, the criteria to define the best nested-pits is based on the NPV and the risk profile of the production plans. Meagher, Sabour, and Dimitrakopoulos [65] develop a two-stage optimisation method that takes into account market and geological uncertainty. The first stage consists of assigning different aspects of the uncertainty to the economic value of the blocks. The second stage uses a parametric min-cut algorithm for generating lower risk pushbacks considering the values from the first stage.

### 2.2.3 Resource constrained pits

Another approach to defining pushbacks is to assume that pushbacks are resource-constrained pits. An early method to find resource-constrained pits was developed by Wang and Sevim [97]. The authors introduced a heuristic algorithm to find maximum value pits with a given number of blocks. The key idea of the introduced heuristic algorithm is that finding the best pit with  $M$  blocks over a total of  $M+N$  blocks is equivalent to eliminating the  $N$  worst blocks (least metal). Thus, the proposed algorithm uses a modified floating cone procedure to find and eliminate the  $N$  blocks.

A more rigorous approach for the constrained pit problem (called maximal closure on a graph with resource constraints) is presented by Tachefine and Soumis [91]. The authors give an mixed integer linear programming formulation for the constrained pit problem and develop a solution method based on the Lagrangian relaxation to solve instances up-to 100,000 blocks with 3 to 4 resource constraints. The authors tested the bundle and the sub-gradient methods to find the Lagrangian multipliers. Both methods gave satisfactory results for solving the instances considered close to optimality.

Somrit and Dagdelen [87], approximated the pushback problem as a simplified multi-period production scheduling problem (MPPSP). The authors assume that each pushback corresponds to a period of the MPPSP; thus, the mining and processing capacity constraints act as a material content constraint for the pushbacks. Also, some constraints, such as average grade and chemical content, were added to the model. The objective function contains time-discounted cash-flow, but because the model assumes one pushback per period, the time-discount factor is also a pushback discount factor. Somrit and Dagdelen proposed the use of solution strategies such as the Lagrangian relaxation, period (pushback) aggregation and other variable reduction methods to improve the tractability of the problem.

Tabesh and Askari-Nasab [90] presented a binary programming model that imposes upper bounds for both the total tonnage and ore content of each pushback. The formulation maximises the un-discounted cash-flows associated with the extraction of the blocks, and no other constraints apart from the wall slope condition were included. The authors use a Lagrangian relaxation as a bounding method to eliminate some of the problem

variables, and a clustering algorithm (described in [89]) to reduce the number of binary variables and to improve the practicality of the solutions. The method is tested in an ore-deposit with 19,561 blocks and contrasted with traditional methods.

Askari-Nasab, Mieth, and Mohammad Tabesh [7] build on the work presented by Tabesh and Askari-Nasab [90]. The authors introduced a local search and a greedy procedure to improve the tractability of the method studied in [90]. A mine with approximately 177,000 blocks is used to test the heuristics, and the results are compared to the ones obtained from the revenue factor parametrisation method.

Meagher, Dimitrakopoulos, and Vidal [66] presented a binary programming formulation for a single pushback problem. The objective function maximises the total undiscounted cash-flow subject to wall slope conditions and a resource limit constraints (imposed over the material considered as ore). The model considers a dynamic cut-off grade optimisation, in other words, is capable of defining whether the block is ore or waste within the optimisation (based on the resource limit constraint).

#### 2.2.4 Pushbacks as *nested-pits*

Elkington and Durham [30] introduced an MILP formulation to design pushbacks from a series of *panels* (intersection between nested-pits and benches) aimed at maximising the NPV of a production schedule. The nested-pits are previously calculated using the revenue factor parametrisation method. The authors conducted a computational experiment over a -relatively small- problem instance with 34,461 continuous variables, 693 binary variables and 46,007 constraints which took 13 hours to solve.

Jelvez, Morales, and Askari-Nasab [49] proposed a BLP formulation that generates pushbacks as a collection of predefined nested-pits. Similar to Elkington and Durham [30], the nested pits are computed beforehand considering the revenue-factor parametrisation method. The Jelvez, Morales, and Askari-Nasab formulation for pushback selection does not maximise time-discounted cash-flows as in Elkington and Durham; instead, other objective functions are proposed such as minimising tonnage differences between pushbacks, minimisation the stripping (waste/ore) ratio in the first pushbacks and maximising the ore content within each pushback. Nevertheless, it only takes into account the desired content for each pushback.

Wharton and Whittle [99] presented an algorithm implemented in the software Whittle (Geovia) that imposes a mining width on the nested-pits after they are calculated. The authors approach the minimum mining width as a rectangle of  $n$  and  $m$  blocks in the  $x$  and  $y$  direction (hereafter *rectangular template*). All the blocks within the rectangular template, up to a user-defined tolerance, must belong to the same pit-shell.

The algorithm evaluates whether the blocks within the template satisfy the mining width by superimposing the template over all the blocks, level by level, starting from the bottom level of the UPL to the top. In the case that the condition is not satisfied the value of the pit shell associated with the blocks is modified to generate a valid mining

width. For a set of experiments, the authors reported a loss of net present value (NPV) for the whole mine of between 3% and 22% after the inclusion of the mining width. The significant loss in NPV reported is produced by two combined factors. First, there is an inherent loss of NPV due to the imposition of the minimum mining width, since wider pits take more time to be excavated, delaying the extraction of valuable material. Second, the algorithm described adds the mining width constraint as a post-process, after the set of unconstrained pit shells has been obtained.

Consuegra and Dimitrakopoulos [22] proposed a multi-stage approach for pushback definition. Pushbacks are considered as a collection of nested-pits, computed using the revenue factor parametrisation method. The first stage consists of computing the nested-pits; the second stage defines different pushback candidates (groups of nested-pits) according to predefined criteria such as desired tonnage content or stripping (waste/ore) ratio. Finally, in the third stage, a stochastic production plan is considered for each set of pushback candidates, selecting the ones that produce the best-expected NPV with minimum risk.

Goodfellow and Dimitrakopoulos [44] proposed two formulations aimed at modifying pushbacks in order to reduce the geological risk (grade variability) associated with the material sent to processing. The objective function of the first formulation consists of minimising the cumulative deviation (with respect to a target value) over the processing destination. The second objective function consists of the minimisation of a standardised quadratic deviation concerning the target. Only wall slope conditions have been considered for pushback creation. A Simulated Annealing algorithm is developed and implemented to solve the proposed models.

### 2.2.5 Semi-practical pushbacks

There have been a number of earlier attempts to model semi-practical pushbacks in the literature, which we now outline.

Juarez et al. [52] sketched a methodology to define semi-practical pushbacks and their schedule at the same time. The algorithm, described as an approximate dynamic programming approach, creates and explores several mining sequences taking into account geometric considerations. The methodology starts by calculating a set of nested pits that are later used to generate *extraction zones* heuristically (aggregates of blocks that satisfy semi-practical constraints). Each extraction zone is either a conical pit or an expansion from a previous pit. In the latter case, the set of nested pits is used to limit the extraction zone to one of the nested pits that satisfies the minimum width (among other parameters of the software) required to form the pushback. Adjacent *extraction zones* are aggregated and considered as pushbacks.

From this, a tree of possible (and feasible) states for the mine for each period is created. Metrics, like cumulative NPV at each state or predicted NPV from each state, are used to prune the tree to find a solution in a reasonable time. Finally, the path of chronologically sequential states that maximise the NPV over the pruned tree is chosen. Due to

commercial confidentiality, the authors give no details of the scheduling heuristic, nor the aggregation process. This approach has two main drawbacks. First, the use of the nested pits as a guide might bias the solution. Second, since the number of possible states is substantial, the quality of the solution might be severely compromised by the pruning algorithm and the metrics used.

On the other hand, Zhang [103] introduced a simulated annealing meta-heuristic to produce flatter bottom pushbacks that satisfy mining width constraints. The idea of the algorithm is to evolve from a set of pushback designs to another set of pushbacks with higher NPV. Each pushback is characterised by a single four-dimensioned point  $(x, y, z, t)$  that is used to select which blocks belong to each pushback. The blocks selected are those that have a minimum value over a distance metric to the point  $(x, y, z, t)$ . For the first iteration, the time component is calculated from a direct schedule of the blocks (a general formulation for the direct block scheduling problem is described in [13]). For subsequent iterations, the reference point is randomly generated by the perturbation of the reference point obtained in the previous iteration of the algorithm.

For each candidate solution, a smoothing heuristic is used to ensure the minimum mining width and the flatter bottom constraints. The smoothing heuristic employs the rectangular template introduced in [99]. The objective function of the meta-heuristic is maximising the NPV, which is calculated by scheduling the blocks within each pushback from the first to the last and from the uppermost to the lowermost level. Although an improvement in NPV has been shown in two case studies, the heuristic nature of this approach does not guarantee the optimality of the solution. Also, the algorithm does not ensure the connectivity of the blocks within each pushback.

Similarly, Froyland and Menabde [38] proposed a method to produce semi-practical pushbacks, named *Generic Kumpking* (this term only appears in Bhp patents and refers to methods used in internal mine planning softwares), which starts by aggregating the blocks in the block model in clumps, to reduce the number of variables of the problem. Clumps are scheduled with respect to mining and processing capacities and later clustered into pushbacks. The clustering method uses spatial and temporal coordinates, where the emphasis of the time coordinate can be modified to generate more or less mineable clusters. The minimum mining width constraint is imposed on the clusters, as a post-process, to create semi-practical pushbacks. However, only details of the aggregation methods are provided, and no disclosure of the clustering method and the mineable criteria have been given.

Likewise, Farmer and Dimitrakopoulos [35] introduced a framework to create semi-practical pushbacks from a pre-conceived direct schedule of blocks. The method begins by aggregating the adjacent blocks that are scheduled at the same period to form *shapes*. If the number of blocks in a shape is larger than a threshold, the shape is considered as a mineable-shape; otherwise, it is not a mineable shape. Mineable-shapes are then clustered using a variation of the k-means algorithm according to their attributes: extraction period and spatial location, where the emphasis of each attribute can be modified. The clusters are used as *seed pushbacks* to which non-mineable shapes are later attached. Finally, the minimum mining width constraint is added as a post-process using the rect-

angular template introduced in [99]. Since the only step of the procedure that uses optimisation is the direct schedule of blocks, the optimality of resultant pushbacks cannot be guaranteed. Moreover, the incorporation of the mining width as a post-process may severely reduce the quality (regarding NPV) of the solution, as noted in [99].

Recently, Bai et al. [10] introduced an algorithm to generate semi-practical pushbacks using image modification tools. *Resource-constrained pits*, pits obtained by weighting the metal price with a weight such that a resource constraint is satisfied, are the basis for the construction of their semi-practical pushbacks. The algorithm transforms each horizontal section of a pushback into binary images, which are manipulated by using morphological tools to enforce the minimum width, smoothness and continuity constraints. The modification of each pushback proceeds level-by-level, from bottom to top. The objective function of the algorithm is to maintain the NPV close to the one obtained from parametric nested pits. Pushbacks are created individually (one after other) from first to last. Blocks within the pushbacks created are not considered for the calculation of the following resource-constrained pits. However, the sequential generation of pushbacks may preclude the generation of globally optimal pushbacks. Additionally, since the resource-constrained pits are calculated using the parametrisation technique, pushbacks obtained may suffer from the gap problem.

### 2.2.6 Ramp design

In this section, we review the state of the art of the open pit ramp design problem. Although there are many articles about the open pit ramp design problem, the majority of them focus on practical guidelines instead of mathematical formulations or algorithms. We are aware of only a few publications that analyse the in-pit ramp design problem, and none on the ex-pit road design problem. For that reason, we survey some useful literature on road design in forest operations that can be applied to the ex-pit (low stripping) road design problem. Interested readers can find a more complete review of optimisation techniques in forest road network problems in [3].

#### Low stripping roads

If the stripping associated with the construction of a road is not significant, the problem can be reduced to the problem of finding the minimum cost curvature-constrained path between two directed points. Different approaches considering continuous or discrete optimisation have been studied. For instance, Chang et al. [20] introduce a continuous-space formulation to find a least cost curvature-constrained path in a plane where the cost function is anisotropic. However, this technique is only valid when the construction cost is independent of the geographical location. More recently, Casal, Santamarina, and Vázquez-Méndez [19] have presented a sequential quadratic programming method for the optimisation of horizontal alignment roads (which are composed of circular curves and tangent line segments joined by means of transition curves). This model can be

used in road reconstruction projects. Also, Mondal, Lucet, and Hare [69] develop a bi-level programming model for the optimisation of the horizontal alignment in a specified corridor. The proposed method obtains locally optimal horizontal alignment and vertical alignment that is globally optimal.

On the other hand, discrete optimisation techniques have been widely used to design roads in forest operations. One of the most interesting approaches consists of dividing the terrain into cells and then associating with each cell a certain set of nodes to represent the directions in which the cell can be accessed. Thus, the turning radius constraint can be mathematically modelled and discrete optimisation techniques can be applied (for example, see [62], [33], [5], [67], [73] and [79]).

Current digital elevation models allow the discretisation of the terrain in cells of size 10 by 10 meters, and the number of cells selected as neighbours to each cell is 8 or 24. A smaller cell size and a higher number of neighbours increases the resolution of the model, but makes it harder to solve ([45] and [88]). Because roads in forest operations are low stripping, this approach can be implemented to find the minimum cost ex-pit roads.

Interestingly, [15] uses three-dimensional blocks to compute a more accurate cost (compared to the method of estimating the cost by the interpolation of 2D vertical sections) for the Earth Allocation Problem. The partition of the problem domain into 3D regular blocks, known as a block model, is often used in mining optimisation problems (see [74]) and, indeed, is a standard tool in mine planning. This approach is relevant to us since we use a block model structure as the basis for the mixed integer linear programming model introduced in Chapter 3.

## High stripping ramps

A complete description of the in-pit (high stripping) ramp design problem can be found in [23]. In addition, [53], [8] and [92] provide a good explanation of hauling operational issues and ramp design guidelines. Ramp parameters, such as width, super-elevation, gradient and construction materials, and operational considerations, such as safe distances between trucks and the use of traffic signals, are explained well in those articles.

To our knowledge, there are only two papers in the literature that discuss algorithms to solve the in-pit ramp design problem. Dowd and Onur [28] presents a simple algorithm that starts from a specific block at the bottom of the pit and then generates one clockwise ramp and one anti-clockwise ramp from bottom to top. After these ramps are generated, the algorithm iterates to another block in the bottom of the pit until no more blocks can be evaluated. Finally, the algorithm selects the ramp with minimum loss in profit. Depending on the topography surrounding the pit, clockwise and anti-clockwise ramps can differ greatly in the amount of stripping and the pit exit location. However, the algorithm does not consider the addition of switchbacks to the ramp or place a constraint on the pit exit, creating purely spiral maximum gradient roads.

Gill [41] and [42], on the other hand, present a dynamic programming algorithm to

minimise the cost of the ramp construction for a given open pit. Using a graph to represent the pit contour, the algorithm is capable of incorporating gradient constraints as well as minimising the stripping cost. Gill claims that his algorithm can find the optimum ramp under some assumptions, but no details are provided due to commercial confidentiality.

## 2.3 Open-pit mine production scheduling problem

In 1968, Johnson [50] formulated a linear programming (LP) model for the open-pit mine production scheduling problem (OPMPSP). The author proposed a Dantzig-Wolfe decomposition to solve the LP model by separating the multi-period problem into a series of single-period sub-problems. The master problem accounts for the constraints associated with the production capacities, while the subproblems force the wall-slope conditions. The model considers fractional variables associated with the extraction and processing of blocks. Therefore the solutions obtained by this method might need rounding if fractional values are obtained.

Dagdelen [25] proposes a Lagrangian decomposition for the mine production scheduling problem, that decomposes the multi-period problem into a series of single period sub-problems (equivalent to the LG-UPL problem) which can be solved by using the LG-3D algorithm. A sub-gradient method is considered to find the Lagrangian parameters. However, the Lagrangian relaxation method may generate the so-called *gap problem* (a big difference in volume between two adjacent shells) precluding the finding of a feasible solution in all cases.

Akaike [2] worked on the solution strategy introduced by Dagdelen [25]. Akaike proposed a new method to update the Lagrangian multipliers, which consists of modelling the mine production scheduling problem as a 4D-network flow problem. The production capacity constraints are relaxed and penalised in the objective function to create the 4D-network, leaving only the wall slope condition. The problem is then solved using maximum-flow algorithms. The method is reported to be faster than the one proposed by Dagdelen, but it still suffers from the *gap problem*.

Caccetta and Hill [17] redefined the variables of the OPMPSP that account for the extraction period of a block from *at* period  $t$  to *by* period  $t$ . The authors introduced a branch-and-cut strategy, that combines *depth first search* with *breath first search* strategies to solve large problem instances of the OPMPSP. According to the results reported by the authors, problems instances of up to 200,000 blocks can be efficiently solved (in less than 24 hours) close to optimality.

Gaupp [39] introduced a series of techniques to improve the tractability of the OPMPSP such as preprocessing variable elimination routines and generation of LP cuts to strengthen the formulation of the OPMPSP, using the Lagrangian relaxation techniques. Cullenbine, Wood, and Newman [24] builds on Gaupp techniques. A Sliding Time Window Heuristic (STWH) is introduced, which consists of repeatedly solving a partly relaxed sub-problem

of a mixed integer linear programming model. At each iteration of the algorithm, a sub-problem is set-up and solved, then a fraction of the variables of that sub-problem is fixed, and this is repeated until all variables of the mixed integer linear programming model have been fixed.

Bley et al. [13] proposed a method for the determination of clique and cover constraints (cutting planes) which can be added to the OPMPSP to strengthen the formulation. Additionally, the authors proposed a variable reduction method within the branch and bound tree based on the block precedence relationships. Some of the experiments presented show a considerable CPU time reduction compared with the original formulation.

Lambert et al. [57] expanded Cullenbine, Wood, and Newman's work and proposed a series of techniques to improve the tractability of the OPMPSP such as variable elimination, a heuristic to generate initial solutions and the use of Lagrangian relaxation. The authors tested the proposed techniques over problem instances with 10,000 to 20,000 blocks. The solutions time reported are in the range of 924 to 35,449 seconds.

Muñoz et al. [72] studied the Bienstock-Zuckerberg (BZ) algorithm (see [12]), which is presented as a Delayed Column Generation algorithm. The computational performance of the BZ algorithm is compared to a Dantzig-Wolfe decomposition method. The results show that the BZ algorithm outperforms the Dantzig-Wolfe decomposition.

## 2.4 Aggregation of blocks

In some open pit design problems, a change in size and shape of the blocks (by aggregation) has also been used to ensure the fulfilment of some problem constraints such as the mining width (for example, see [58]). Aggregation is a common technique used to reduce the number of variables and constraints in optimisation problems.

*Reblocking* is probably the most common and simple aggregation technique used in open pit optimisation problems. The idea is to create macro-blocks from the union of a set of adjacent blocks. For instance, Figure 2.1 illustrates an example of 4 macro-blocks each of which is the union of 2 by 2 by 2 blocks in  $x$ ,  $y$  and  $z$  coordinates, respectively. However, there are two main issues related to reblocking. First, there is a loss in the quality of the solution due to the aggregation, and, second, larger blocks often violate the slope constraints defined by the disaggregated blocks.

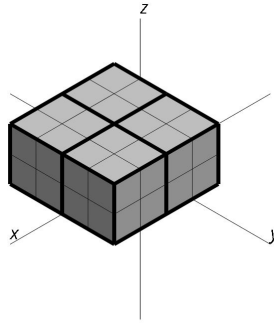


Figure 2.1: Example of 4 macro-blocks each of which is a 2x2x2 reblocking of the original size blocks.

Concerning this latter issue, Ramazan, Dagdelen, and Johnson [81] presented a linear programming methodology to aggregate blocks in subsets with positive value that satisfy the slope constraints of the disaggregated block, so-called *fundamental trees*. Similarly, Froyland and Menabde [38] introduced a linear programming based method to calculate *clumps*, an aggregate of blocks that are spatially connected and have similar properties. Clumps also satisfy the slope constraints of the disaggregated blocks. Other approaches based on agglomerative hierarchical algorithms and Fuzzy C-Means clusterisation algorithms can be found in [89] and [83].



## Chapter 3

# Optimum ramp design in open pit mines

*A pit without roadways cannot be called optimum as roads can change the overall shape of the pit dramatically.*

— Peter Dowd & Ahmet Onur [28]

©Elsevier 2019. Re-printed with permission from:

Yarmuch, J. L., Brazil, M., Rubinstein, H., & Thomas, D. A. (2020). “Optimum ramp design in open pit mines”. In: *Computers & Operations Research*. <https://doi.org/10.1016/j.cor.2019.06.013>. Note that the notation of this chapter differs from the rest of the thesis in order to be consistent to the published article.

**T**HE problem considered in this chapter is the design of the optimal open pit haulage ramp, for a given ramp width and a maximum ramp gradient, which connects two points of the pit, minimising construction and operational costs. Since the stripping cost associated with in-pit ramps can be considerably larger than that of ex-pit roads, we discuss the two different problems: high stripping ramp design and low stripping road design. The first is commonly associated with in-pit ramps, whereas the second is more relevant to ex-pit roads.

In-pit ramps must connect the bottom of the pit to a given pit exit at the top of the excavation (see Figure 3.1). The model should account for the fact that external conditions, such as rock quality, could disqualify some sectors of the pit for ramp construction and that material above the ramp must be extracted so as to satisfy the maximum pit-wall slope conditions. Decision variables of the problem are the ramp starting and final points (from a given set of candidate points), the direction and slope of each section of the ramp, and the number and locations of switchbacks.

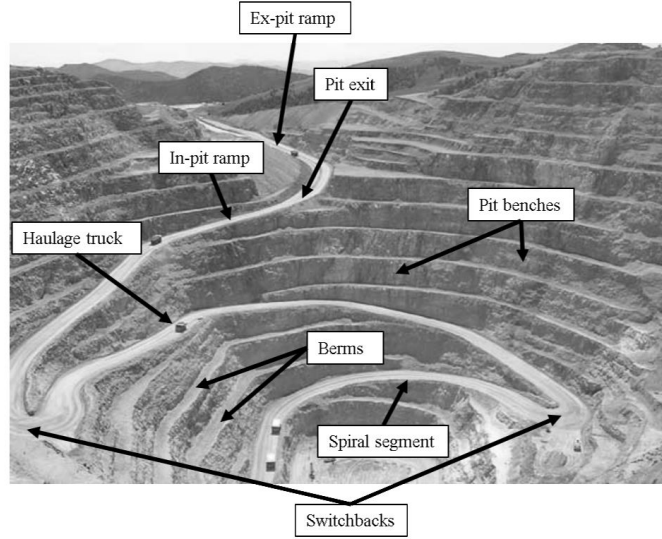


Figure 3.1: Open pit ramp.

The ex-pit road design problem consists of finding the path that connects the pit exit to the mine facilities such as processing plants, stockpiles, dumps and maintenance bays. For this problem, the starting and ending points are given, as well as any forbidden areas. Because the stripping associated with the ex-pit roads is usually not significant, the topography can be considered invariant. Thus, for simplicity, the stripping associated to each segment of the road can be considered as a pre-calculated cost. Even though a change in direction in the path does not have a significant impact in stripping cost, it does increase the road construction and maintenance cost. Therefore, the changes of direction along the road should be taken into account in this problem.

The contributions of this work are first, to adapt the ex-pit road design problem as a minimum distance problem that can be efficiently solved. The second contribution is to develop a new mathematical model to solve the in-pit ramp design problem. In order to be able to solve realistically sized problems within a reasonable time scale, an heuristic approach is developed to solve the high stripping ramp design problem. Additionally, computational results based on real mine data are here presented.

This chapter is organised as follows. In Section 3.1 we present a shortest path formulation to solve the low stripping road problem. In Section 3.2, we give a binary linear programming model for the high stripping ramp design problem. In Section 3.3 we introduce a heuristic approach to solve the high stripping ramp design problem. In Section 3.4 we present a case study for both formulations, and in Section 3.5 present the conclusions.

### 3.1 Optimisation of low stripping roads

Techniques used in the forest industry can be applied to define the optimum ramp in open pit mines. One of the problems that needs to be solved in the planning stage for the forest problem is to determine the layout of the roads that connect the forested (production) areas to the timber mills (processing plants). Road systems of several kilometres in length and at least 10m wide are designed as part of the planning stage. Similarly, the ex-pit haulage roads in the open pit mining problem need to connect the pit exit to the processing plant. The layout selected will define the amount of stripping and the haulage cost, which is proportional to the ex-pit road length.

#### 3.1.1 Shortest path formulation

The problem of finding the minimum path over a weighted graph has been widely studied and efficient algorithms have been developed to find the optimum in polynomial time. According to Bertsimas and Tsitsiklis [11], Dijkstra's algorithm is the most efficient for solving the shortest path problem in dense graphs. However, different implementations of the Dijkstra algorithm improve theoretical computational bounds or empirical running times for special classes of graphs (see [1] for further details).

As mentioned in Section 2.2.6, low stripping roads can be approximated as a road location problem. First, the topography must be projected onto a 2D horizontal plane and divided in regular cells. For each cell we associate a group of nodes that represent the directions in which that cell can be entered and left. The nodes of each cell are then connected to the nodes in the neighbourhood cells, creating a graph that represents possible paths on the topography. Thus, costs for changes in direction and gradient constraints, as well as operational costs depending on the geographical location or characteristics of the ramp can be incorporated into the problem as weights on the arcs between nodes. Note that the representation of directions as nodes in which a cell can be accessed was briefly introduced by Epstein et al. [33] and more described in [32]. The following section explains in detail the construction of the graph for the ex-pit road case.

#### Construction of the graph

First, it is necessary to subdivide the given topography, where the road will be defined, into a set of cells ( $\mathcal{C}$ ). Our approach is to use a regular 2D lattice of cells, where each cell,  $c \in \mathcal{C}$ , represents a rectangular area of the terrain. For each cell  $c$ , we define the set of neighbours cells  $\hat{c} \in \hat{\mathcal{C}}_c$  to generate the number of directions in which the cells will be accessed. For example, in Figure 3.2, the 4 neighbours of the central cell correspond to 4 rectilinear directions. More generally, neighbourhoods of 8 or 24 cells correspond to more directions.

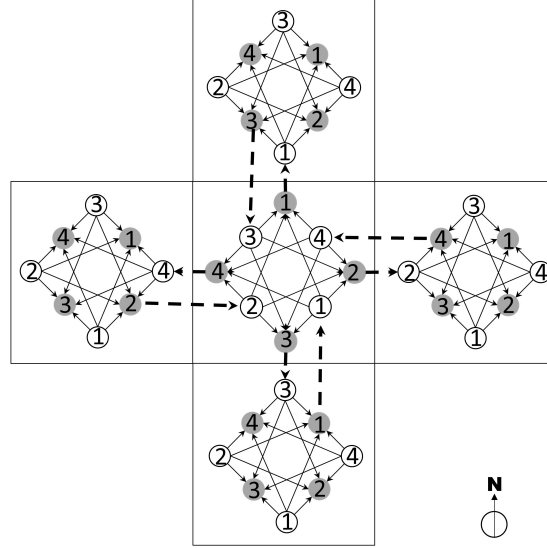


Figure 3.2: Representation of a simple 4 choice of directions graph  $\mathcal{G}$ . 1, 2, 3 and 4 represent North, East, South and West directions, respectively. Vertices in white represent incoming directions, while vertices in grey colour outgoing directions. Solid arcs connect all possible incoming directions to all possible outgoing directions at the same cell. Dashed arcs represent the feasible connections from a cell to a neighbouring cell.

The vertices  $\mathcal{V}$  of the graph  $\mathcal{G}(\mathcal{V}, \mathcal{A})$  represent the possible incoming and outgoing directions of the ramp at each cell. Let  $\mathcal{D}'$  and  $\mathcal{D}''$  represent a discretised set of incoming and outgoing directions respectively. Then, the vertex of an outgoing direction  $d'' \in \mathcal{D}''$  at cell  $c$  is represented as  $v_{d'',c}$ . Similarly,  $v_{d',\hat{c}}$  is the vertex of an incoming direction  $d' \in \mathcal{D}'$  at cell  $\hat{c}$ . Therefore, an arc denoted as  $(v_{d'',c}, v_{d',\hat{c}})$  represents the connection of the outgoing direction  $d''$  at cell  $c$  to the incoming direction  $d'$  at cell  $\hat{c}$ . In Figure 3.2, vertices of incoming directions are illustrated as white nodes, while the vertices of outgoing directions are represented as grey nodes.

Two types of connections are needed to build the graph. To clarify the construction of the graph, we define the disjoint sets of arcs  $\mathcal{A}_d$  and  $\mathcal{A}_c$ , where  $\mathcal{A} = \mathcal{A}_d \cup \mathcal{A}_c$ . The set  $\mathcal{A}_d$  connects all possible incoming directions to all possible outgoing directions at the same cell (solid arcs inside cells in Figure 3.2). Thus,  $\mathcal{A}_d = \{(v_{d',c}, v_{d'',c}), d' \in \mathcal{D}', d'' \in \mathcal{D}'', c \in \mathcal{C}\}$ . The set  $\mathcal{A}_c$  represents the set of permitted connections from a cell to a neighbouring cell (dashed arcs in Figure 3.2). Formally,  $\mathcal{A}_c = \{(v_{d'',c}, v_{d',\hat{c}}), \hat{c} \in \hat{\mathcal{C}}_c, c \in \mathcal{C}\}$ .

Curvature constraints can be modelled within this graph definition. Changes of direction constraints can be included by weighting or cutting off some arcs in  $\mathcal{A}_d$ . Likewise, topographical (or gradient) constraints can be incorporated by adding weights to arcs in  $\mathcal{A}_c$ .

The weights associated with the elements of  $\mathcal{A}_d$  are calculated as the costs of changing the direction  $d'$  to  $d''$  at each cell  $c$ . These costs may vary depending on the angle of the turn, the super-elevation and the maintenance costs. Likewise, weights related to arcs

in  $\mathcal{A}_c$  are designed to reflect the cost of building a section of road from  $c$  to  $\hat{c}$ ; usually this cost is proportional to the Euclidean distance that separates the nodes, adjusted to account for any difference in height.

Once the weights have been calculated and the start and end cells defined, the problem of finding a minimum cost path can be solved using Dijkstra's algorithm for the weighted graph  $\mathcal{G}$  or any other shortest path algorithm that suits graph  $\mathcal{G}$ ). This approach does not take into account a physical modification in topography as we are dealing with the low stripping case. Note that the graph  $\mathcal{G}$  can be also used to solve the problem of finding the minimum cost network when more than two terminals need to be connected through a road network. In this case the problem reduces to the Steiner Tree problem in a graph ([5]), a well-studied problem, for which good exact and heuristic algorithms exist.

## 3.2 Optimisation of high stripping ramps

In this section, we present a binary linear programming (BLP) formulation to find a minimum cost ramp with vertical and horizontal alignment constraints for a given pit, a given set of initial and final points, and a given ramp width. This formulation takes into account the stripping associated with the ramp excavation and the construction and haulage cost.

### 3.2.1 Minimum cost ramp formulation

Here a binary linear model to find the minimum cost ramp is formulated. The inputs for the model are divided into two types: the geological inputs, which will include the geological block model, the topography, the contour of the pit, and the dependencies between blocks based on the required pit slopes; and operational inputs, which include various costs, the ramp width, and the set of possible starting and final points for the ramp. The geological model is assumed to be represented as a regularly spaced set of blocks (equal dimensions in coordinates  $x$  and  $y$ ), and must include the contour of the pit shape we are aiming to achieve. To reduce the size of the problem, blocks that are too far from the contour of the pit can be discarded. We assume the ramp width is equal to the block size in  $x$  ( $dx$ ) or  $y$  ( $dy$ ) and the dimension in  $z$  ( $dz$ ) represents the maximum ramp gradient ( $\max. \text{ slope} \geq \frac{dz}{dx}$  or  $\frac{dz}{dy}$ ). For a standard block model this may necessitate first dividing every block into several horizontal slices (dotted lines in Figures 3.3 and 3.4).

To consider the stripping associated with the ramp, we define a set of vertical precedences (upward), or blocks above the ramp, that need to be extracted to ensure a minimum wall slope (see Figure 3.3). Similarly, to avoid the construction of the ramp above excavated blocks, we consider a set of vertical precedences (downward) to ensure that no block corresponding to the floor of a ramp is extracted.

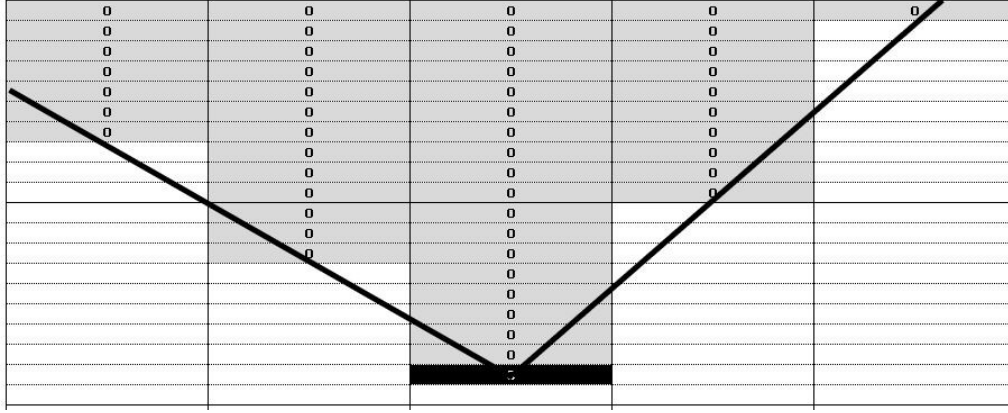


Figure 3.3: Illustration of the wall slope precedences. Blocks in grey need to be mined before the block in black to ensure slope constraint represented by the two inclined lines that start at the black block.

Formally, the precedence relationships are represented in terms of the digraph  $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{A}_1)$  where  $\mathcal{V}_1$  is the set of blocks and  $(u, v) \in \mathcal{A}_1$  only if the block  $v$  must be extracted before the block  $u$ . We assume that digraph  $\mathcal{A}_1$  contains only immediate precedence relationships. That is, if  $u, v$  and  $w \in \mathcal{V}_1$  are such that  $(u, v) \in \mathcal{A}_1$  and  $(v, w) \in \mathcal{A}_1$ , then  $(u, w) \notin \mathcal{A}_1$ . Similarly, the condition that the ramp cannot be built above excavated blocks is represented by a digraph  $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{A}_2)$  where  $(u, v) \in \mathcal{A}_2$  represents the constraint that if the block  $u$  is selected as ramp, then the block  $v$  must not be extracted.

To build  $\mathcal{G}_1$  we add a vertex to  $\mathcal{V}_1$  for each block in the sliced block model, and we add an arc  $(u, v)$  to  $\mathcal{A}_1$  from each vertex  $u \in \mathcal{V}_1$  to all vertices  $v \in \mathcal{V}_1$  such that the minimum wall slope is ensured for the block that is represented by vertex  $u$ . This is a standard method in pit design problems as it can be seen in [74]. The graph  $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{A}_2)$  has a similar structure to  $\mathcal{G}_1$ , but in this case for each vertex  $u$  there is a unique arc  $(u, v)$  in  $\mathcal{A}_2$  such that the block represented by  $v$  is immediately below the block represented by  $u$ .

The ramp is represented as a connected path of adjacent blocks from  $S$  to  $T$  (two artificial blocks that represent the starting and ending points of the ramp). Formally, the possible adjacencies between blocks is represented in terms of the digraph  $\mathcal{G}_3 = (\mathcal{V}_3, \mathcal{A}_3)$ , where there is one vertex  $v \in \mathcal{V}_3$  for each block that can be selected as part of a ramp, plus the initial vertex  $S$  and the final vertex  $T$ . The set of arcs  $(u, v) \in \mathcal{A}_3$  consists of all pairs of vertices that can be selected as adjacent blocks in any feasible ramp. Additionally, we add one edge from  $S$  to every node that can be selected to begin the ramp, and one edge per vertex that can end the ramp to  $T$ . Figure 3.4 illustrates an example of the graph of connections for a simplified case.

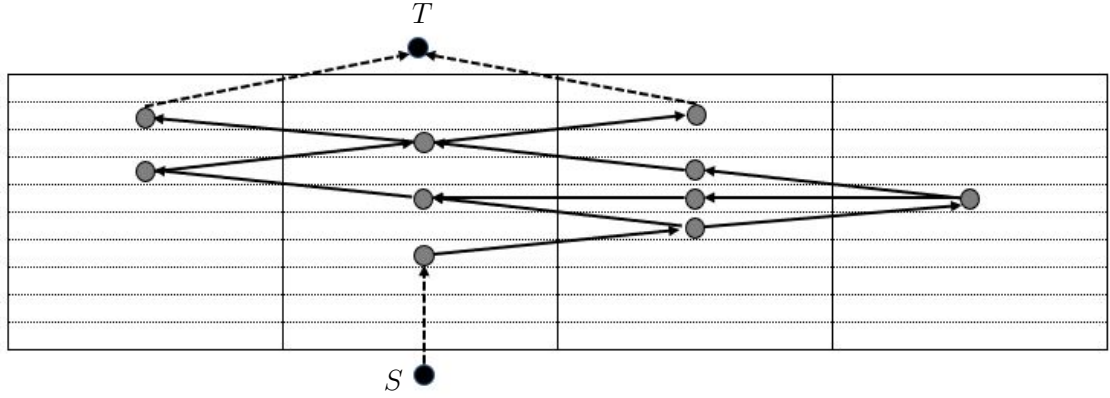


Figure 3.4: Example of a simple 2D case of  $\mathcal{G}_3$ . The vertices in grey illustrate the blocks that can be selected as ramp, the edges in solid black represent the possible connections of the path. The initial ( $S$ ) and final ( $T$ ) vertices are represented in black and the edges from and to those vertices are illustrated with dashed lines.

We now present a formal mixed integer linear programming model for finding the minimum cost in-pit ramp.

### Indices, sets and parameters

In addition to the block model and sets  $\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3$  and  $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$  described above, we have the following sets and given parameters.

$d \in \mathcal{D}$  : types of change of direction (e.g. slight left turn, right sharp turn, switchbacks, etc.).

$\mathcal{A}'_3$  : Set of arcs in  $\mathcal{A}_3$  other than the arcs that end at vertex  $T$ ,  $\mathcal{A}'_3 = \mathcal{A}_3 - \{(i, j) \text{ such that } j = T\}$ .

$\mathcal{A}''_3$  : Set of arcs in  $\mathcal{A}_3$  other than the arcs that start at vertex  $S$  or end at vertex  $T$ ,  $\mathcal{A}''_3 = \mathcal{A}_3 - \{(i, j) \text{ such that } i = S \text{ or } j = T\}$ .

$\mathcal{F}$  : a mapping function  $\mathcal{F} : \{(i, j, k) | (i, j), (j, k) \in \mathcal{A}''_3\} \rightarrow \mathcal{D}$ .

$C_i$  : extraction cost of block  $i$ .

$H_{i,j}$  : haulage cost associated with arc  $(i, j) \in \mathcal{A}''_3$ . Note that the haulage cost  $H_{i,j}$  of any arc  $(i, j)$  should account for the total tonnage that will traverse that arc over the life of the mine.

$P_i^d$  : change of direction cost at block  $i$ , where  $d \in \mathcal{D}$  is the type of change of direction.

### Variables

$r_i$  : binary, equal to 1 if the block  $i$  is selected as a ramp, 0 otherwise.

$x_i$  : binary, equal to 1 if the block  $i$  is extracted, 0 otherwise.

$v_i^d$  : binary, equal to 1 if the ramp undergoes a change in direction of type  $d$  at block  $i$ , 0 otherwise.

$a_{i,j}$  : binary, equal to 1 if the arc  $(i,j) \in \mathcal{A}_3$  is selected as a segment of the ramp, 0 otherwise.

### Objective function

The objective function is defined as follows, subject to a number of constraints:

$$\text{Min: } \sum_{i \in \mathcal{V}_1} C_i \cdot x_i + \sum_{(i,j) \in \mathcal{A}_3''} H_{i,j} \cdot a_{i,j} + \sum_{i \in \mathcal{V}_3} \sum_{d \in \mathcal{D}} P_i^d \cdot v_i^d$$

The objective function minimises the cost of the blocks extracted due to the ramp construction together with the haulage cost and those costs associated with changes of direction of the ramp.

### Geotechnical constraints

$$x_i \leq x_j \quad (i,j) \in \mathcal{A}_1 \quad (3.1)$$

$$r_i \leq x_i \quad i \in \mathcal{V}_1 \quad (3.2)$$

$$r_i \leq 1 - x_j \quad (i,j) \in \mathcal{A}_2 \quad (3.3)$$

Constraint (3.1) ensures the vertical precedences are honoured in the extraction of any block. Constraint (3.2) ensures that every block flagged as part of the ramp must be extracted. Constraint (3.3) avoids building a ramp in the air.

### Ramp constraints

$$\sum_{(i,j) \in \mathcal{A}_3'} a_{i,j} \leq r_j \quad (3.4)$$

$$r_i + r_j \leq 1 + a_{i,j} \quad (i,j) \in \mathcal{A}_3'' \quad (3.5)$$

$$\sum_{(i,j) \in \mathcal{A}_3} a_{i,j} - \sum_{(j,k) \in \mathcal{A}_3} a_{j,k} = \begin{cases} -1, & \text{if } j = S \\ 1, & \text{if } j = T \\ 0, & \text{if } j \neq S \text{ and } j \neq T \end{cases} \quad (3.6)$$

Constraints (3.4) to (3.5) flags the block  $j$  as belonging to the ramp if and only if the block  $i$  belongs to the ramp and the arc  $a_{i,j}$  has been selected. Note that Constraint (3.4) could be also expressed as  $a_{i,j} \leq r_j, \forall (i,j) \in \mathcal{A}'_3$ , but Constraint (3.4) gives a stronger formulation. Constraint (3.6) ensures the connectivity of the ramp.

### Change of direction constraints

$$a_{i,j} + a_{j,k} \leq v_j^d + 1 \quad (i,j), (j,k) \in \mathcal{A}''_3, \quad d = \mathcal{F}(i,j,k) \quad (3.7)$$

Constraint (3.7) identifies the change of direction at each segment of the ramp.

### Scope of the variables

$$x_i, r_i, a_{u,v}, v_j^d \in \{0,1\} \quad i \in \mathcal{V}_1, d \in \mathcal{D}, (u,v) \in \mathcal{A}_3, j \in \mathcal{V}_3 \quad (3.8)$$

Finally, Constraint (3.8) defines the scope of the variables, all of which are binary.

## 3.3 Solution method

With the mathematical formulation presented in Section 3.2.1, the high stripping ramp model can be implemented and solved directly with a commercial MIP-solver. In this section we introduce much faster heuristic solution approach based on backtracking algorithms. A new algorithm to find feasible paths with mutually exclusive ramp segments is introduced in Section 3.3.1. In Section 3.3.2 we describe a local search method to improve the solution found by the algorithm presented in Section 3.3.1. Throughout this section we use the same notation as in Section 3.2.1.

### 3.3.1 Mutually exclusive greedy adaptive path (MEGAP) algorithm

This new algorithm is an adaptation of the depth-first search algorithm presented in [93]. The idea of the MEGAP algorithm is to traverse the graph  $\mathcal{G}_3$  from the source node  $S$  to the sink node  $T$  following a greedy procedure. The way in which  $\mathcal{G}_3$  is traversed ensures that the nodes in the solution ramp  $\mathfrak{p}$  satisfy the constraints introduced in Section 3.2.1. In other words, the resultant path will be a feasible solution for the In-pit ramp problem.

Before introducing the MEGAP algorithm, define  $\mathcal{G}_{tc} = (\mathcal{V}_{tc}, \mathcal{A}_{tc})$  as the transitive closure of  $\mathcal{G}_1 \cup \mathcal{G}_2$ . Also, the stripping associated with each node  $u \in \mathcal{V}_3$  is calculated as the out-degree of  $u \in \mathcal{G}_{tc}$ .

**Algorithm 1:** MEGAP, Mutually Exclusive Greedy Adaptive Path

---

**Data:**  $\mathcal{G}_3$  with edges sorted by stripping in increasing order,  $\mathcal{G}_{tc}$ ,  
starting node  $S$ , final node  $T$

**Result:** A path  $p$  that is a feasible solution for In-pit ramp problem

```

1 begin
2    $p \leftarrow \emptyset$ ;
3    $cN \leftarrow \emptyset$ ;
4   foreach  $u \in \mathcal{V}_3$  do
5      $u.visited \leftarrow \text{False}$ ;
6      $u.predecessor \leftarrow \text{null}$ ;
7   MEGAPvisit( $S$ )
8    $p \leftarrow p \cup \{T\}$ 
9    $pred \leftarrow T.predecessor$ 
10  while  $pred \neq \text{null}$  do
11     $p \leftarrow p \cup \{pred\}$ 
12     $pred \leftarrow pred.predecessor$ 
13  Output  $p$ 
14  Function MEGAPvisit ( $u$ ):
15    foreach  $v$  such that  $(u, v) \in \mathcal{A}_{tc}$  do
16       $cN \leftarrow cN \cup \{v\}$ 
17     $u.visited \leftarrow \text{True}$ ;
18    foreach  $v$  such that  $(u, v) \in \mathcal{A}_3$  do
19      if  $(v.visited = \text{False})$  and  $(v \notin cN)$  and  $(T.visited = \text{False})$ 
20        then
21           $v.visited \leftarrow \text{True}$ ;
22           $v.predecessor \leftarrow u$ ;
23          MEGAPvisit( $v$ )

```

---

The MEGAP algorithm works as follow. Lines 2–6 initialise the variables  $p$  (ramp),  $cN$  (nodes in conflict with the ramp segments already selected), and the node attributes  $visited$  and  $predecessor$  for all nodes in  $\mathcal{G}_3$ . Line 7 begins the exploration of  $\mathcal{G}_3$  from node  $S$ . Lines 10–12 store the solution ramp in the variable  $p$ . Line 13 returns the solution ramp  $p$ .

The function MEGAPvisit (Lines 14–22) consists of a greedy recursive exploration of  $\mathcal{G}_3$  from node  $u$  taking into account the current list of conflicted nodes  $cN$ . Lines 14–15 update the conflicted node list  $cN$ . Line 16 marks node  $v$  as visited. Lines 17–21 explore the non-visited and non-conflicted nodes  $v$  such that  $(u, v) \in \mathcal{A}_3$ , following a recursive greedy type procedure until the node  $T$  is reached.

The computational complexity of MEGAP is bounded by  $O(2 \cdot |\mathcal{V}_3| + \Delta(\mathcal{G}_{tc}) \cdot |\mathcal{A}_3|)$ , where  $\Delta(\mathcal{G}_{tc})$  is the maximum degree of  $\mathcal{G}_{tc}$ .

### 3.3.2 Local search

The local search ( $LS$ ) procedure consists of a partial exploration of all possible paths from  $S$  to  $T$  that prunes the paths that “promise” poor values for the optimisation problem.

At each stage of the *LS* algorithm, the best feasible solution (incumbent) is stored and used to prune the exploration of the paths. More specifically, the *LS* procedure prunes all paths that have a value less than an incumbent value plus a given tolerance function ( $tol(i)$ ) which decreases proportionally with the number of iterations ( $i = 0, 1, \dots, I$ ) of *LS*.

The *LS* procedure relies on a sub-routine called `allComb2( $u$ )` which finds all feasible combinations of length 2 from node  $u$  (denoted by  $\mathcal{C}_u^2$ ) in the graph  $\mathcal{G}_3$ . For each combination  $c_u \in \mathcal{C}_u^2$  a path from  $S$  to  $T$  is calculated using `Algorithm 1`, with the condition that the nodes in  $c_u$  are in the resultant ramp. The *LS* procedure begins by running `Algorithm 1`, the resultant ramp ( $R_0$ ) is stored as the incumbent and the cutoff value is calculated as  $cutoff_0 = \text{cost}(R_0) \cdot (1 + tol(0))$ . Then, a limited branching from node  $S$  is calculated using the procedure `allComb2( $S$ )`, which gives at most  $|\mathcal{C}_S^2|$  feasible ramps ( $\mathcal{R}_S$ ). All the paths which give a cost greater than  $cutoff_0$  are pruned. The procedure repeats the branching at all nodes in the feasible paths at distance  $2 \cdot i$  updating  $cutoff_i$  at each iteration and updating the incumbent if a better value is found, until the maximum number of iterations  $I$  is reached.

### 3.4 A case study

Data from a copper mine was used to test our mathematical models. The mine is located in the Antofagasta province, Chile. The mining reserves within the pit are composed of 84.2 million tonnes of ore and 52.6 million tonnes of waste. For planning purposes, the final pit is subdivided in 4 operational pushbacks, or inner phases. For simplicity, the first pushback was selected to test the performance of the high stripping model.

Two ramp location problems, ex-pit and in-pit, were studied to test our low and high stripping models respectively. For the ex-pit road case, we compared the low stripping to the high stripping model for a given topography. On the other hand, for the in-pit ramp case, we compared the high stripping model to a ramp designed by an experienced mining engineer.

Additionally, we compared the roads obtained by imposing a forbidden region to those with no such region. The model introduced in Section 3.1 was implemented in Matlab R2016a. The model proposed in Section 3.2.1 was implemented in Python 2.7 and solved using Gurobi 7.5 MILP-solver. All the pre-processing routines, such as the definition of the sets and variables, as well as the Local search method introduced in Section 3.3.2 were coded in Python. Execution of the code was done on an Intel Xeon machine with 32 gigabytes running Windows 10. This machine has eight processors that run at 2.40 gigahertz.

### 3.4.1 Ex-pit road

Two models, a low stripping and a high stripping formulation, and two scenarios, one with no obstacles and the second avoiding the grey area showed in Figure 3.5, were considered to test our approaches on real mine data. The task is to find minimum cost paths from the pit exit  $A$  to four possible locations for a processing plant ( $L1 - L4$ ).

In the first case, a nearly flat surface with an extension of 570 m by 600 m and a regular grid of 30 m was used to test the low stripping formulation. Eight incoming and eight outgoing directions were considered for each cell of the grid. The order of the graph  $\mathcal{G}$  is  $20 \times 21 \times 16 = 6,720$ . We defined a vertical gradient constraint of 10% and a horizontal change of direction cost proportional to the degree of the change. The horizontal change of direction was divided into three categories: slight turns (changes up to  $45^\circ$ ), right angle turns (changes of  $90^\circ$ ) and pronounced turns (changes from  $90 - 180^\circ$ ). The costs considered for each category were US\$14,580, US\$21,870 and US\$29,160, which correspond to a factor of 2, 3 and 4 times the mining cost for a ramp segment, respectively. Additionally, we assumed a ramp construction cost of US\$340.2 per linear meter.

For simplicity, the minimum cost path was solved using a Matlab implementation of the Dijkstra shortest path algorithm. However, other algorithms might be faster than Dijkstra for the graph  $\mathcal{G}$ . According to [1] Dial's implementation of the Dijkstra algorithm has excellent empirical results (see [1] pages 113-114 for Dial's implementation details).

The first scenario involved finding four minimum cost paths from point  $A = (1035, 1215, 298)$  to locations  $L1 = (1545, 1305, 280)$ ,  $L2 = (1545, 795, 271)$ ,  $L3 = (1305, 945, 292)$  and  $L4 = (1185, 765, 289)$  independently, and the second scenario involved finding minimum cost paths between the same points but avoiding the grey zone in Figure 3.5. Since we are asked to find the paths from  $A$  to  $L1 - L4$  independently, the shortest path approach gives optimal results. However, if the task was to find the shortest network that connects all terminals  $A$ ,  $L1$ ,  $L2$ ,  $L3$  and  $L4$ , then a Steiner tree in graph algorithm should be used, as mentioned in Section 3.1. An initial North-South incoming direction at point  $A$  was specified to represent the ending of an in-pit ramp.

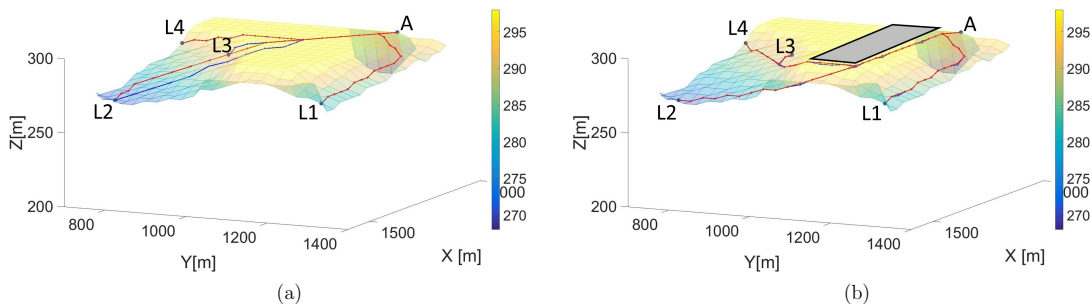


Figure 3.5: Illustration of Scenario 1 (a) and Scenario 2 (b). The grey zone in Figure (b) represents a forbidden area for the roads. Roads in red represent paths obtained with the high stripping model, while roads in blue show paths obtained with the low stripping model.

In the second case, a grid of 5,600 regular blocks of  $30\text{m} \times 30\text{m} \times 3\text{m}$  and the same economic parameters as in the first case were used. Also, a mining cost of \$1 per ton was added to account for the stripping cost, and an overall maximum wall slope of  $45^\circ$  was stipulated. Figure 3.5 illustrates the graphic results of both cases and scenarios, and Tables 3.1 and 3.2 summarize the results for the first and the second scenario respectively. The red path represents the high stripping solution while blue is used to represent the low stripping path (in most cases both solutions overlap and only one colour is visible).

Small differences in length are explained by the different representation of the topography of the formulations, a 3D set of blocks in the case of the high stripping model and a surface in the case of the low stripping formulation. The cost function is composed of the construction cost associated with the length of the road plus the changes of direction cost, where the latter depends on the number of slight (S.T.), right angle (R.T) and pronounced turns (P.T.). The high stripping formulation also includes the stripping cost associated with the modification of the topography.

| Case/Path | Length[m] | Cost[US\$] | S.T. | R.T. | Solv t[s] |
|-----------|-----------|------------|------|------|-----------|
| LS/A-L1   | 573.10    | 231,421    | 1    | 1    | 0.05      |
| LS/A-L2   | 685.15    | 262,250    | 2    | 0    | 0.05      |
| LS/A-L3   | 399.82    | 157,890    | 0    | 1    | 0.07      |
| LS/A-L4   | 512.58    | 188,960    | 1    | 0    | 0.06      |
| HS/A-L1   | 572.94    | 231,365    | 1    | 1    | 200.81    |
| HS/A-L2   | 685.01    | 254,910    | 1    | 0    | 175.69    |
| HS/A-L3   | 382.04    | 137,263    | 0    | 0    | 89.51     |
| HS/A-L4   | 512.58    | 188,960    | 1    | 0    | 147.64    |

Table 3.1: Minimal paths for scenario 1.

| Case/Path | Length[m] | Cost[US\$] | S.T. | R.T. | Solv t[s] |
|-----------|-----------|------------|------|------|-----------|
| LS/A-L1   | 573.10    | 231,421    | 1    | 1    | 0.05      |
| LS/A-L2   | 878.67    | 342,663    | 3    | 0    | 0.06      |
| LS/A-L3   | 770.49    | 327,732    | 3    | 1    | 0.05      |
| LS/A-L4   | 964.99    | 379,321    | 2    | 1    | 0.07      |
| HS/A-L1   | 572.94    | 231,365    | 1    | 1    | 102.57    |
| HS/A-L2   | 878.53    | 342,618    | 3    | 0    | 126.28    |
| HS/A-L3   | 770.21    | 327,637    | 3    | 1    | 132.98    |
| HS/A-L4   | 964.83    | 379,267    | 2    | 1    | 146.09    |

Table 3.2: Minimal paths for scenario 2.

For scenario 1, the high stripping model (HS) shows better results for routes from point A to Location 2 (A-L2) and from point A to location L3 (A-L3). In both cases, one block was extracted (7,290 tonnes), reducing the length of the path and saving a change of direction. The short-cut (in red) shown in route A-L3 in Figure 3.5 is not an option in the low stripping formulation due to the gradient constraint and the impossibility of modifying the topography. An improvement of 2.7% and 13% were obtained in path A-L2 and A-L3 using the HS model (see Table 3.1).

For paths A-L1 and A-L4 of scenario 1 and all paths of scenario 2, the HS and the LS formulations gave the same results. The average computation time for the LS model is 0.06 seconds versus 140.19 seconds for the HS model. Finally, the pronounced turns (P.T.) column is not shown in Tables 3.1 nor 3.2 since none of the paths contained a pronounced turn.

### 3.4.2 In-pit ramp

As mentioned in the previous section, the low stripping formulation is not suitable for finding the minimum cost ramp in steep terrain. Therefore, we compare the performance of the high stripping formulation to a ramp previously designed by a mining engineer. We used the same initial geological and topographical data that was used by the engineer for his design.

The pushback contains a total of 22.2 million tonnes of ore and 42.4 million tonnes of waste that need to be hauled to the top level of the pit. An average haulage cost of  $8.046 \cdot 10^{-4}$  [US\$/ton/m] was used for the optimisation model. The binary linear programming model (BLP) was solved using the commercial software Gurobi. The resulting BLP has 218,826 constraints, 65,568 binary variables and 702,991 non-zero coefficients. The solver default optimisation parameters remained unchanged except for the time limit, which was set to 24 hours (86,500 [s]).

The best solution of the BLP found by the solver within the time limit was US\$4,628,154 with an optimality gap, measured against linear relaxation, of 31.7%. However, despite the significant gap this solution is 43% more economical than the engineer's design which has a cost of US\$8,101,162.

We ran the local search procedure described in Section 3.3.2 to find a good solution in a more reasonable time (a few hours). The tolerance function  $tol(i)$  was defined as  $tol(i) = 0.2 \cdot e^{-0.4 \cdot i}$ . The resulting ramp found by our procedure has a value of US\$5,430,055, which is 33% better than the engineer's solution. Our solution method took only 1,145 [s] (approximately 19 minutes).

Figure 3.6 illustrates the evolution in time of the solutions found using the BLP model and the one found with our local search method. Note that the Local search procedure was entirely coded in native Python which is, in general, much slower than lower level programming languages such as C or C++. Therefore, a considerable speed up in our results can be expected by implementing our proposed algorithms in lower level programming languages.

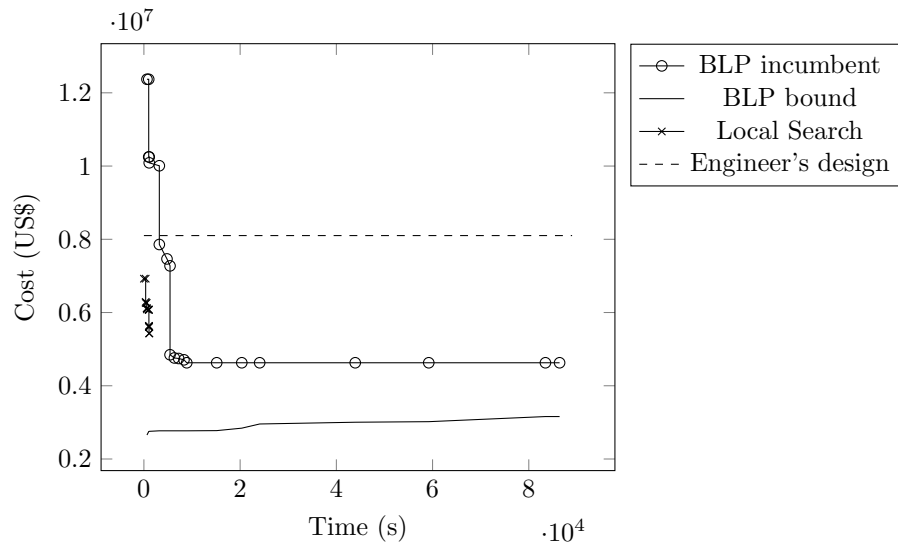


Figure 3.6: Comparison of the performance (cost and time) of the binary linear programming (BLP) model and the proposed local search method (Local Search).

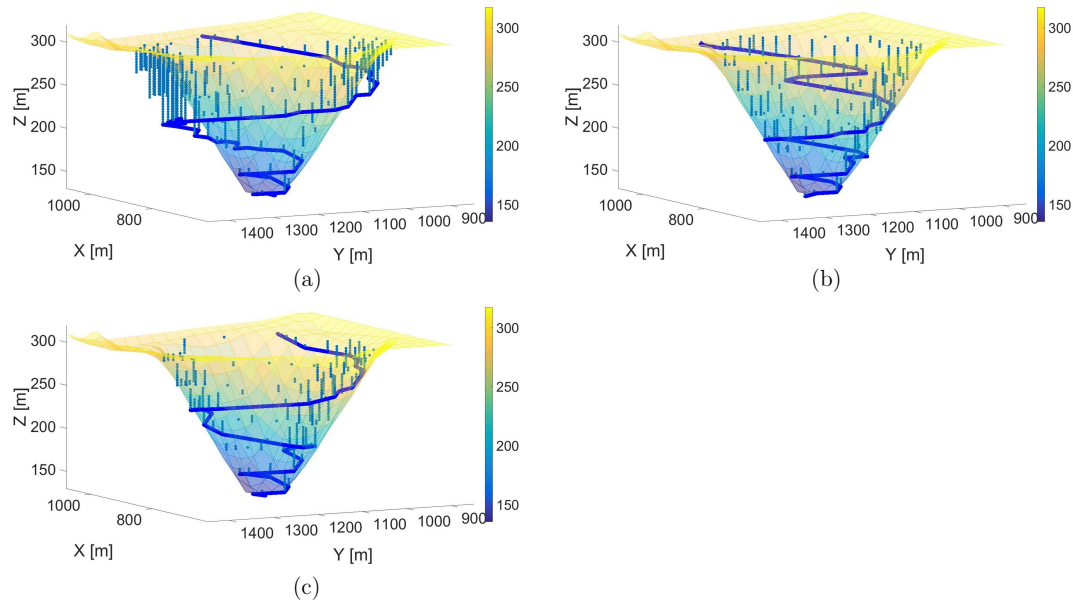


Figure 3.7: Illustration of the in-pit ramps. Figures (a), (b) and (c) represent the engineer design, the ramp obtained with the local search method and the ramp obtained solving the BLP model, respectively. Ramps are represented in blue lines, while the stripping associated with the ramps is represented with blue dots.

Figure 3.7 shows the engineer's design (a), the ramp obtained with the local search method (b) and the ramp found using the high stripping formulation. Ramps are drawn with blue lines, and the stripping associated with the construction of the ramp is repre-

sented with blue dots.

The cost of the high stripping ramp is 43% less than the cost of the engineer's design ramp (see Table 3.3). Although this result shows a great improvement in cost, it is not entirely surprising because the traditional ramp design process is done manually, so it relies on the engineer's experience. It is worth noting that some of the value gained in the optimisation might be lost in the smoothing process of generating the engineering CAD designs. However, the loss in value due to the smoothing process can be controlled by choosing a suitable block discretisation as well as a proper modelling of the curvature constraints.

|      | Length [m] | Total cost [US\$] | Stripping [US\$] | Hauling [US\$] | Direction [US\$] | Solv t. [sec] |
|------|------------|-------------------|------------------|----------------|------------------|---------------|
| BLP  | 2,047.4    | 4,628,154         | 3,455,460        | 1,085,215      | 87,480           | 86,500        |
| LS   | 2,146.5    | 5,430,055         | 4,196,870        | 1,087,385      | 145,800          | 1,145         |
| Eng. | 2,289.3    | 8,101,162         | 6,714,090        | 1,124,632      | 262,440          | N/A           |

Table 3.3: In-pit ramp cost comparison between the BLP model, the local search model and the engineer's design.

### 3.5 Conclusion

This chapter has presented a solution for the open pit ramp design problem. Two situations were studied: the low stripping and the high stripping cases. For the first case, an adaptation of the shortest path method was undertaken. In the second case, a binary linear programming approach was developed. In both cases, our methods give an exact solution to the problem within the model constraints. Additionally, both models allow gradient and curvature constraints to be handled within the given discretisation. They can incorporate construction and haulage costs and the formulations are flexible enough to find an optimal ramp that avoids any given forbidden region.

The low stripping road model can be solved in polynomial time ( $O(|A| + |V| \cdot \log(|V|))$  for a graph  $G(V, A)$ ). However, this formulation assumes that the road respects the existing topography, without the need for excavation. On the other hand, the high stripping ramp formulation can handle complex and steep terrains where extensive excavation may be required to satisfy the gradient constraint on the road. However, the high stripping model is much less efficient than the low stripping one, and does not scale well for large problems. To account for this we introduced a new algorithm that produces feasible solutions to the high stripping model (MEGAP algorithm) in linear time. We then developed a local search heuristic based on the MEGAP algorithm that rapidly finds good solutions for the high stripping model (typically within 20 minutes).

Finally, computational experiments were performed for ex-pit roads and in-pit ramps. When no stripping was required, the low stripping model shows the same results as the high stripping model, but running, on average, 2000 times faster. For the in-pit ramp case, the high stripping model gave a ramp with 43% less cost than the cost of the ramp

previously designed by an experienced mining engineer.

Some limitations are worth noting. Our in-pit model assumes that the optimal ramp occurs on the boundary of a precomputed pushback, which might be suboptimal compared to the problem of finding a ramp and a pushback simultaneously. Currently, we are working on a new mathematical model that incorporates the ramp design into the pushback optimisation problem. Also, a combined in-pit ramp design and scheduling optimisation problem, that takes into account the ways in which the properties of the haulage network interact with the mining schedule, would be another worthwhile avenue for future work.

### 3.6 Summary

The problem studied in this chapter is that of designing the optimal open pit haulage ramp that, for a given ramp width and a maximum ramp gradient, connects two points of the mine, minimising construction and operational costs. Because in-pit ramps require the removal of a considerable amount of non-valuable material (stripping), we discuss two different problems: high stripping (or in-pit) ramp design and low stripping (or ex-pit) road design. For the first situation, we present a mixed integer linear programming model; in the second case, a shortest path approach is undertaken. In both cases, the models can be solved exactly, and include gradient and curvature constraints. The proposed formulations have been tested on real mine data, showing a significant reduction in cost compared to the previous mine design.



## Chapter 4

# A mathematical model for the practical pushback design problem

*Pushbacks are the foundation upon which all open-pit mine sequence optimisation and budget should be built.*

— Anita Eccles [29].

**T**O spread the financial cost of mining waste material and to speed up the cash-flow associated with ore extraction, the ultimate pit limit (UPL) is subdivided into smaller volumes (pushbacks) that can be sequenced and mined. In this chapter we introduce the practical pushback design problem and present a mathematical model that approximates the practical constraints presented in Section 1.1.

The *practical pushback design problem* consists of finding a partition of the blocks inside the UPL that maximises the economic benefit of the mining extraction. Each subset of the partition (hereafter called practical pushback) must satisfy a number of practical constraints imposed by the mining technology in use. In the case of a truck-shovel system, practical constraints are: 1) there should be a minimum mining width to ensure the safe operation of the shovels and the supporting equipment required at every bench, 2) every bench of each pushback must be accessible via a connected haulage ramp that allows the trucks to reach the surface, and 3) each pushback must be a connected set, in terms of horizontal and vertical adjacencies between blocks. From now, we use this as a formal definition for practical pushbacks.

The contributions of this work are first to model the practical constraints in a mixed integer linear programming formulation to generate optimal practical pushbacks. This model has been designed to incorporate these constraints in a way that allows realistic sized problems (with millions of blocks) to be solved after suitable aggregation. Second, introduce a *closeness factor* to measure the similarity between a computer generated solution and a corresponding manual design that includes the practical constraints. Finally, a computational result using real mine data, comparing the use of our formulation to a traditional methodology is presented.

This chapter is organised as follows. In Section 4.1, a mathematical formulation to this problem is presented. In Section 4.2, introduce the *closeness factor* to measure the

## 4.1 The mathematical model

The ramp is modelled as a set of adjacent blocks that creates a continuous path from the bottom to the top of the pushback. At each level, the ramp must have a minimum number of blocks to satisfy the ramp gradient constraint. We define a *ramp segment* as a set of adjacent blocks that connect two consecutive levels and satisfy the gradient constraint. The bottom-most block from which a ramp segment starts is defined as a *ramp entrance*.

Figure 4.1: Example of a *ramp segments template* composed of 16 *ramp segments* with a 1:4 gradient constraint, plan view, considering block 122 as a *ramp entrance*.

The mining width and the connectivity constraint are modelled by a *compactness factor*  $\lambda$  which biases the objective function to encourage the selection of blocks that are closer to the ramp entrance. For values of  $\lambda$  large enough, wide and connected shapes can be obtained. Extremely large values of  $\lambda$  will force the algorithm to select all blocks within a given distance from the ramp entrance which do not violate the model constraints.

The following describe the sets, indices and parameters used in our formulation. Guidelines presented in [94] were followed to facilitate the readability of our formulation.

### Indices and sets

$p \in \mathcal{P}$  : pushbacks;  $1, \dots, P$

$b \in \mathcal{B}$  : blocks;  $1, \dots, B$ .

$l \in \mathcal{L}$  : levels in the block model;  $1, \dots, L$ .

$\check{b} \in \check{\mathcal{B}}_l$  : blocks that belong to a level  $l$ .

$\check{b} \in \check{\mathcal{B}}_b$  : blocks that share the same level with block  $b$ .

$\hat{b} \in \hat{\mathcal{B}}_b$  : blocks constituting vertical precedences (upwards) for block  $b$  (see example in Figure 4.2).

$b_s^+ \in \mathcal{B}_s^+$  : set of blocks immediately above blocks in ramp segment  $s$ .

$b_s^- \in \mathcal{B}_s^-$  : set of blocks immediately below blocks in ramp segment  $s$ .

$s' \in \mathcal{S}'_b$  : ramp segments that start at block  $b$ . The block  $b$  act as a *ramp entrance* of the segments  $s' \in \mathcal{S}'_b$

$s'' \in \mathcal{S}''_b$  : ramp segments that end at block  $b$ .

$s \in \mathcal{S}$  : all ramp segments ( $\mathcal{S} = \bigcup_{b \in \mathcal{B}} \mathcal{S}'_b$ ).

$D_{b\check{b}}$  : Euclidean distance from block  $b$  to block  $\check{b}$  (metres).

$D_p^*$  : maximum distance from the ramp entrance to every block at the same level as the ramp entrance that belongs to pushback  $p$  (metres).

$V_b$  : net profit of block  $b$ , this will be negative for any block that is waste (dollars).

$W_b$  : tonnage of block  $b$  (tons).

$\underline{W}_p$  : minimum tonnage of pushback  $p$  (tons).

$\overline{W}_p$  : maximum tonnage of pushback  $p$  (tons).

$\lambda_b$  : compactness factor of pushback  $p$ .

$\delta_p$  : value discount factor for pushback  $p$ . The value of  $\delta_p$  implicitly assumes the timing of each pushback is known beforehand.  $\delta_p$  can be estimated as  $\frac{1}{(1+r)^t}$  where  $t$  is the average period when the pushback  $p$  is mined and  $r$  represents the financial discount rate.

The blocks that belong to the set  $\hat{\mathcal{B}}_b$  are those which need to be removed before the extraction of block  $b$  in order to keep the maximum wall slope constraint. This idea of using vertical precedences to guarantee geotechnically feasible wall slopes was established by [60]. For example, Figure 4.2 illustrates a five block  $45^\circ$  wall slope constraint, assuming cubic blocks. Here, blocks 1 to 5 need to be removed before the extraction of block 6, thus,  $\hat{\mathcal{B}}_6 = \{1, 2, 3, 4, 5\}$ .

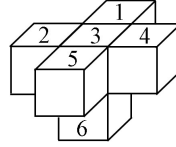


Figure 4.2: Example of a  $45^\circ$  geotechnical slope upward precedence.

To build the set  $\mathcal{S}$  it is necessary to define a template of ramp segments that connect two consecutive levels satisfying the maximum gradient constraint. In the example in Figure 4.1, we consider a template of 16 possible connections with origin in the block 122. Note that there may be more than one path between the same start-and end-pair of blocks. Therefore, to define the set  $\mathcal{S}$  it is necessary to compute the template for all blocks for levels  $l > 1$ , where the top level is  $l = 1$ . For each  $s \in \mathcal{S}$ , the set  $\mathcal{B}_s$  is simply generated by adding the blocks that belong to each template. In the example of Figure 4.1, if the top-left ramp segment is ramp segment 1 then the set  $\mathcal{B}_1$  is  $\mathcal{B}_1 = \{122, 113, 104, 95, 5\}$ .

### Decision variables

Three types of variables are used in the model, all of them binary. The first variable type is associated with the extraction of the block:

$$x_{bp} = \begin{cases} 1, & \text{if the block } b \text{ belongs to pushback } p \\ 0, & \text{otherwise} \end{cases}$$

The second variable type represents the ramp segment which determines the ramp, and acts as a relative coordinate system to control the shape of the pushback:

$$z_{sp} = \begin{cases} 1, & \text{if the ramp segment } s \text{ is used in pushback } p \\ 0, & \text{otherwise} \end{cases}$$

Finally, the third type of variable accounts for the simultaneous extraction of a ramp entrance and the blocks on the same level. The variable  $u_{b,\check{b},p}$  is set to be zero if  $D_{b,\check{b}} > D_p^*$

$$u_{b\check{b}p} = \begin{cases} 1, & \text{if } \sum_{s' \in \mathcal{S}'_b} z_{s'p} = 1 \text{ and } x_{bp} = 1 \\ 0, & \text{otherwise} \end{cases}$$

### Objective function

The objective function is defined as follows subject to a number of constraints:

$$\text{Max } \sum_{p \in \mathcal{P}} \delta_p \left( \sum_{b \in \mathcal{B}} V_b \cdot x_{bp} + \lambda_p \sum_{b \in \mathcal{B}} \sum_{\check{b} \in \mathcal{B}_b} \frac{u_{b\check{b}p}}{D_{b\check{b}}} \right)$$

The objective function maximises the discounted profit of the mining extraction associated with the pushbacks biased by the spatial proximity horizontally from the blocks to the ramp entrance of each pushback (controlled by the parameter  $\lambda_p$ ).

$$\sum_{p \in \mathcal{P}} x_{bp} \leq 1, \quad b \in \mathcal{B} \quad (4.1)$$

$$\sum_{p \in \mathcal{P}} z_{sp} \leq 1, \quad s \in \mathcal{S} \quad (4.2)$$

$$\underline{W}_p \leq \sum_{b \in \mathcal{B}} W_b \cdot x_{bp} \leq \overline{W}_p, \quad p \in \mathcal{P} \quad (4.3)$$

Constraint (4.1) forces each block to be assigned to at most one pushback. Similarly, Constraint (4.2) ensures that each ramp segment can be assigned to at most one pushback. Constraint (4.3) states the minimum and maximum tonnage for each pushback. This type of constraint can be extended to any content requirement. However, when Constraint (4.3) accounts for the total tonnage (as in this case), the maximum tonnage is also bounded by the chosen maximum distance  $D_p^*$ .

### Geotechnical constraints

$$\sum_{\rho \leq p} x_{b\rho} - \sum_{\rho \leq p} x_{\hat{b}\rho} \leq 0, \quad b \in \mathcal{B} \quad \hat{b} \in \hat{\mathcal{B}}_b \quad p \in \mathcal{P} \quad (4.4)$$

$$\sum_{\rho \leq p} z_{s\rho} + \sum_{\rho \leq p} x_{b_s^-\rho} \leq 1, \quad s \in \mathcal{S} \quad b_s^- \in \mathcal{B}_s^- \quad p \in \mathcal{P} \quad (4.5)$$

Constraint (4.4) ensures the vertical precedences are honoured in the extraction of any block. Note that Constraint (4.4) is equivalent to  $x_{bp} \leq \sum_{\rho \leq p} x_{\hat{b}\rho}$ , but Constraint (4.4) gives a tighter bound in the linear relaxation of the IP model (see [13]). Constraint (4.5) avoid the extraction of blocks below the ramp.

### Ramp continuity constraints

$$\sum_{s'' \in \mathcal{S}_b''} z_{s''p} - \sum_{s' \in \mathcal{S}_b'} z_{s'p} = 0, \quad b \in \mathcal{B} \quad p \in \mathcal{P} \quad (4.6)$$

$$x_{\check{b}p} - \sum_{\check{b} \in \check{\mathcal{B}}_b} \sum_{s' \in \mathcal{S}_b'} z_{s'p} \leq 0, \quad l \in \mathcal{L} - \{1\} \quad \check{b} \in \check{\mathcal{B}}_l \quad p \in \mathcal{P} \quad (4.7)$$

$$\sum_{\check{b} \in \check{\mathcal{B}}_l} \sum_{s' \in \mathcal{S}_b'} z_{s'p} \leq 1, \quad l \in \mathcal{L} \quad p \in \mathcal{P} \quad (4.8)$$

Constraint (4.6) ensures the continuity of the ramp segments. Constraint (4.7) ensures that there exists a ramp segment of pushback  $p$  at each level of any block of pushback  $p$ . Constraint (4.8) imposes a maximum of one ramp segment per level per pushback. Note that Constraints (4.1),(4.2), (4.5) and (4.8) could give rise to clique cuts that can be generated a priori or internally in some solvers.

### Ramp accessibility constraints

$$\sum_{\rho \leq p} z_{s\rho} - \sum_{\rho \leq p} x_{b_s^+ \rho} \leq 0, \quad s \in \mathcal{S} \quad b_s^+ \in \mathcal{B}_s^+ \quad p \in \mathcal{P} \quad (4.9)$$

Constraint (4.9) forces the extraction of blocks above ramp segments. The previous discussion of Constraint (4.4) also applies here.

### Ramp and blocks extraction constraints

$$u_{b\check{b}p} - x_{\check{b}p} \leq 0, \quad b \in \mathcal{B} \quad \check{b} \in \check{\mathcal{B}}_b \quad p \in \mathcal{P} \quad (4.10)$$

$$u_{b\check{b}p} - \sum_{s' \in \mathcal{S}_b'} z_{s'p} \leq 0, \quad b \in \mathcal{B} \quad \check{b} \in \check{\mathcal{B}}_b \quad p \in \mathcal{P} \quad (4.11)$$

Constraints (4.10)-(4.11) implies that if  $u_{b\check{b}p} = 1$  then both, a ramp segment starting at block  $b$  and a block  $\check{b}$  belong to pushback  $p$ .

## 4.2 Closeness factor

One of the major issues of models used as a guidance in the pushback design process is the great amount of manual intervention needed to make the solution practical. A significant disparity between the guidance design produced by the mathematical model

and the subsequent manual design of the engineer may imply a substantial loss of value. Therefore, we define a closeness factor ( $c_f$ ) to measure the similarity between the mathematical solution and the engineer's design (see Figure 4.3).

$g_b$  = index of the pushback assigned to block  $b$  that is used as a guidance.  $g_b$  is set to 0 for unassigned blocks.

$d_b$  = index of the pushback that is assigned to block  $b$  after the engineering design.  $d_b$  is set to 0 for unassigned blocks.

$N^g$  = total number of blocks  $b \in \mathcal{B}$  such that  $g_b > 0$ .

$N^d$  = total number of blocks  $b \in \mathcal{B}$  such that  $d_b > 0$ .

$N$  = maximum of  $N^g$  and  $N^d$  ( $\max(N^g, N^d)$ ).

$$m_b = \begin{cases} 1, & \text{if } g_b = d_b \text{ and } g_b > 0, d_b > 0. \\ 1/2, & \text{if } |g_b - d_b| = 1 \text{ and } g_b > 0, d_b > 0. \\ 0, & \text{otherwise.} \end{cases}$$

The matching factor  $m_b$  takes a value of 1/2 when the difference between  $g_b$  and  $d_b$  differs in at most one unit to take into account that consecutive pushbacks may be extracted at the same period. Because of this,  $c_f$  is only meaningful for mines with more than three pushbacks.

$$c_f = \frac{\sum_{b \in \mathcal{B}} m_b}{N} \% \quad (4.12)$$

We use Figure 4.3 to illustrate a simple example of the closeness factor. In this 2D example the number of blocks is  $N = 109$  ( $N^g = N^d = 109$ ), and the total matching factor is  $\sum_{b \in \mathcal{B}} m_b = 67.5$ . Therefore, inserting the values in Equation 4.12 we get a  $c_f = 61.93\%$ .

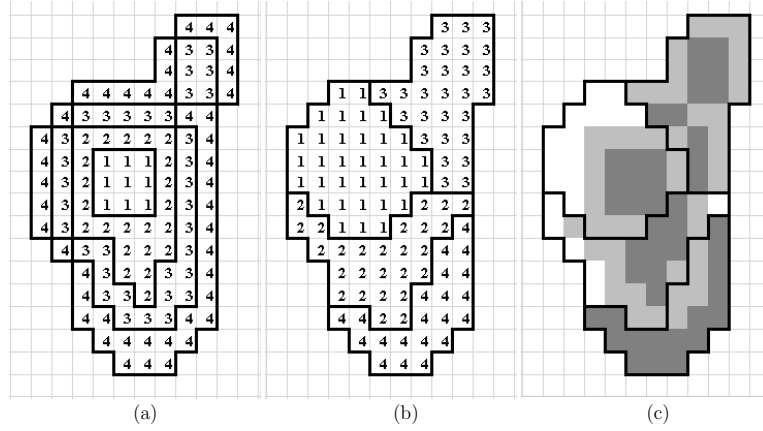


Figure 4.3: Illustration of the matching factor (plan view). (a) Guidance shells. (b) Manual design. (c) Matching blocks,  $g_b = d_b$  and  $|g_b - d_b| = 1$  are represented in dark and light grey, respectively.

### 4.3 Case study

The model introduced in Section 4.1 was implemented in Python 2.7 and solved using Gurobi 7.5 MILP-solver for a real mineral deposit. All the pre-processing routines, such as the definition of the sets and variables was done in Python. Execution of the code was done on an Intel Xeon machine with 32 gigabytes running Windows 10. This machine has eight processors that run at 2.40 gigahertz.

An early test of our formulation showed that only instances with fewer than 10,000 blocks could be solved in reasonable computational time (typically considered as a day in mine planning literature, for example see [13], [63] and [21]), for a single pushback. Consequently, two techniques were considered to solve realistic problem instances. First, the original block model was reblocked to reduce the size of the problem. Second, we created the set of pushbacks sequentially, from first to last. When generating the next pushback the previous pushbacks are fixed, so only unassigned blocks are considered for future pushbacks.

Since the results from our approach still need refinement to be implemented, the guidance pushbacks obtained were manually smoothed and scheduled to compute a detailed NPV for the life of the mine. Additionally, for comparison purposes, a senior planning engineer manually designed and scheduled the same mineral deposit following the traditional methodology (nested pits) by using commercial mine planning software. The similarity between both outputs (our model and the traditional methodology) and the respective manual design was measured using the *closeness factor*.

In practice, the number of the pushbacks is obtained only after incorporating the haulage ramps, ensuring the connectivity and the mining width of the guidance pits. Therefore, no predefined number of pushbacks was considered either by our methodol-

ogy or by the traditional methodology. However, our model requires that a maximum number of possible pushbacks is given beforehand, as well as an estimation of the  $\delta_p$  values for each pushback. A simple way of calculating the maximum number of pushbacks is to divide the minimum tonnage desired inside the pushbacks by the total tonnage inside the UPL. Similarly, the values of  $\delta_p$  can be estimated by simply assuming that the time taken to extract a pushback is proportional to its tonnage (bounded by the maximum tonnage constraint), and by assuming that the pushbacks are mined sequentially one after the other.

The test instance consists of a copper deposit located in the Copiapó region, Chile. The original block model consists of 7,776,000 blocks of 5x5x5 metres each, which were aggregated in 6 by 6 by 6 blocks to obtain a block model of 36,000 blocks of dimensions of 30x30x30 metres each. The mine already has a unique processing plant for copper sulphides, also known as a concentrator plant. A set of realistic technical and economic parameters was chosen to compute the profit of the blocks, to generate the pushbacks design and to produce the mine schedule. Table 4.1 summarises the economic, design and scheduling parameters. Since we are now using cubic blocks of 30x30x30 metres each, the ramp width is treated as 30 metres in our model. Note that the aggregation of the original block model into 20x20x20 metres blocks results in 121,500 blocks, which cannot be solved in a reasonable time according to our early test experiments. However, the ramp width of 20 metres was considered in the practical design after the optimisation (manual smoothing).

| Parameter                   | Unit      | Value  |
|-----------------------------|-----------|--------|
| Mining cost                 | \$/ton    | 2.2    |
| Processing cost             | \$/ton    | 5.3    |
| Processing recovery         | %         | 88     |
| Copper price                | \$/pound  | 2.0    |
| Treatment and refining cost | \$/pound  | 0.5    |
| Minimum mining width        | metres    | 60     |
| Ramp gradient               | %         | 20     |
| Ramp width                  | metres    | 20     |
| Mining rate                 | kton/year | 22,000 |
| Processing plan capacity    | kton/year | 12,000 |
| Discount rate               | % /year   | 12.5   |

Table 4.1: Economic, design and scheduling parameters.

#### 4.3.1 Optimal practical pushbacks

In this section, we show how to generate a set of pushbacks using the mathematical model introduced in Section 4.1, which can be implemented and solved directly in any MILP-solver. The commercial solver Gurobi 7.5 was used with default parameters except for the selection of the aggressive generation of clique cuts.

### Model simplifications

We noticed three major computational bottlenecks in our model. First, the linear relaxation of Constraint (4.2) is likely to produce loose upper bounds due to the highly fractional values of variables  $x$  in the branch and bound process. Second, our model requires a considerable number of binary variables  $u$  (proportional to the value of  $D^*$  to the power of two) when large values of  $D^*$  are chosen. Third, the number of binary variables  $z$  grows exponentially with the inverse of the ramp's gradient. Therefore, we developed some model simplifications and inspection techniques to overcome these issues.

Constraint (4.2), that accounts for the minimum and maximum tonnage inside each pushback, can be removed by instead searching for a proper set of values of the parameter  $D_p^*$ . By imposing a maximum distance of blocks from the ramp entrance as a constraint, the volume bound for a pushback can be well controlled.

According to our previous experiments, our model is generally able to produce connected and wide pushbacks even when the value of  $\lambda_p$  is set to zero. Geological deposits where the amount of waste is comparable to the amount of ore (a ratio between the tonnes of waste over the ore inside the UPL not larger than 4) and where the ore is not highly disseminated (scattered) seem to be most favourable for setting the value of  $\lambda_p$  to zero. Hence, in those cases, as in the geological model used in our case study, the set of variables  $u_{b,\check{b},p}$  is no longer needed. Thus, the objective function can be simplified to:  $\text{Max } \sum_{p \in \mathcal{P}} \delta_p \sum_{b \in \mathcal{B}} V_b \cdot x_{b,p}$ , and Constraints (4.10)-(4.11) can be replaced by Constraint (4.13):

$$D_{b,\check{b}} \cdot \left( \sum_{s' \in \mathcal{S}'_b} z_{s'p} + x_{\check{b},p} - 1 \right) \leq D_p^*, \quad b \in \mathcal{B}, \quad \check{b} \in \check{\mathcal{B}}_b, \quad p \in \mathcal{P} \quad (4.13)$$

Ideally, the set  $\mathcal{S}$  should incorporate all possible ramp segments. However, the number of possible ramp segments from a block to the level above grows exponentially with the inverse of the gradient of the ramp segment. For example, if the gradient constraint is 1:1, only eight possible ramp segments per block exist to connect that block to the level above. However, that number rapidly grows to more than 6.5 million when a gradient of 1:8 is considered. Our model assumes that the ramp width is equivalent to the width of the block. If that is not the case, the resultant ramp after optimisation may not be optimal.

Therefore, we tested two options: 1) choosing a subset of possible connections for a ramp gradient of 1:8 (around 50 ramp segments per block), and 2) relaxing the ramp gradient constraint to 1:2 and considering the 56 possible ramp connections per block. Computational experiments showed that smoother pushbacks (similar to an engineer's design) were obtained as a consequence of the relaxation of the ramp gradient constraint. Thus, the second option was taken for the case study. Note that relaxing the ramp gradient constraint precludes the use of the resultant ramp as guidance for the manual ramp design, but it can be used to control the shape of the pushback. However, a practical ramp can be included in the semi-practical pushbacks with limited effect if the slope constraint is modified to take into account the impact of a haulage ramp system (see Figure 4.4). In our case, we assumed that a 20 metres wide ramp intersected a wall of 450 metres

height four times, on average, modifying the slope constraint from 45 to 40 degrees.

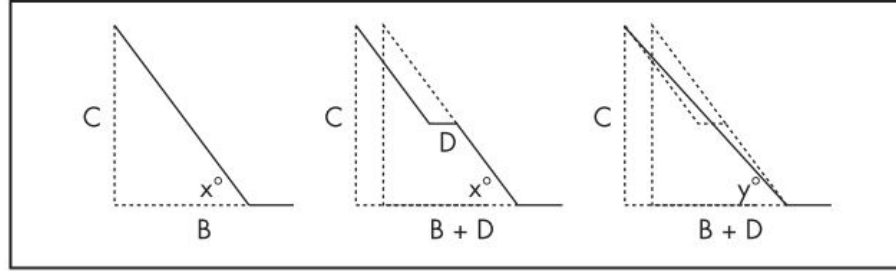


Figure 4.4: Modified slope  $y^\circ$  after a ramp of width  $D$  has been included ([100]).

## Results

To begin, an ultimate pit was computed to discard non-profitable blocks that would not need to be extracted. Since there is no guarantee that the optimal solution of our formulation is contained within the ultimate pit limit, an increment of 20% of the original profit was considered to generate a larger ultimate pit. At the end of this stage, the problem instance was reduced from 36,000 blocks to a total of 3,916 blocks. For simplicity, the discount factor  $\delta_p$  was estimated by assuming that pushbacks are sequentially mined and it takes one year to mine each pushback ( $\frac{1}{(1+r)^p}$ , where  $r$  is the discount rate).

By varying the value of  $D^*$  (the maximum permitted distance of a block from the ramp entrance) the number and volume of the pushbacks can be controlled. In our case, we evaluated distances between 350 and 550 metres to generate useful semi-practical pushbacks that can be used in practice. The value of  $D^*$  was modified in increments of 50 metres for each pushback until a total of 12 pushbacks were obtained (which we will refer to as guidance pushbacks). The total running time for the 12 guidance pushbacks was 13.27 hours. Table 4.2 summarises the set of distances  $D^*$  for each pushback. Table 4.3 summarises the material content and value of the guidance pushbacks.

| Pushback       | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| $D^*$ (metres) | 350 | 350 | 350 | 350 | 350 | 350 | 350 | 350 | 450 | 450 | 550 | 550 |

Table 4.2: Maximum distance parameter  $D^*$  for each pushback used in the optimisation. Values showed here are the result of a trial and error process.

Subsequently, a total of 10 optimised practical pushbacks were manually designed based on the 12 guidance pushbacks obtained from our optimisation model. Only guidance pushbacks 10 to 12 required significant modification. The main reasons for the changes in guidance pushbacks 10-12 were: the disconnection of some of their blocks and the lack of space to satisfy the minimum width due to the greedy type procedure

used to solve the problem. The time spent to design the optimised practical pushback from the guidance pushback was about 3 hours.

After including a ramp of gradient 1:8 per pushback (relaxed to 1:2 in the optimisation model), ensuring the pushback connectivity and the minimum mining width, a large closeness factor was obtained. The value of the closeness factor between the guidance pushbacks and the optimised practical pushback was 93.07%. Figure 4.5 shows the guidance pushbacks and the optimised pushbacks, where each pushback is represented as a different colour. The small differences between the guidance pushbacks and the optimised practical pushbacks can be appreciated by considering the images in Figure 4.5, and comparing Tables 4.3 and 4.4 (note that profits are un-discounted).

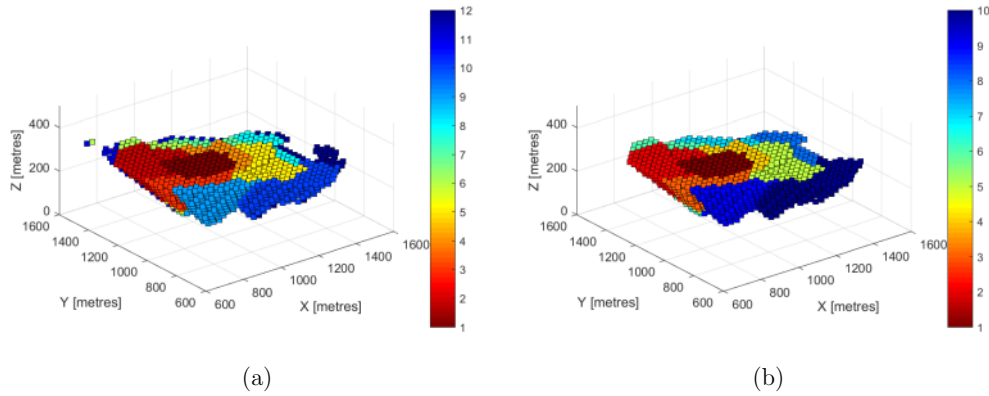


Figure 4.5: Isometric view of the guidance and optimised practical pushbacks, colours represent different pushbacks (a) Guidance pushbacks. (b) Optimised practical pushbacks.

| Pushback | Ore (ton)  | Waste (ton) | Total (ton) | Profit (usd) |
|----------|------------|-------------|-------------|--------------|
| 1        | 7,513,800  | 775,246     | 8,289,046   | 71,298,992   |
| 2        | 9,628,022  | 6,506,128   | 16,134,150  | 72,759,124   |
| 3        | 5,529,310  | 5,236,801   | 10,766,111  | 32,333,601   |
| 4        | 6,760,774  | 3,157,441   | 9,918,215   | 36,693,063   |
| 5        | 8,249,160  | 8,786,724   | 17,035,885  | 27,138,592   |
| 6        | 2,508,376  | 3,542,088   | 6,050,463   | 11,122,914   |
| 7        | 5,418,826  | 5,417,041   | 10,835,867  | 15,166,455   |
| 8        | 4,527,967  | 9,745,389   | 14,273,356  | 9,738,948    |
| 9        | 5,170,225  | 9,516,064   | 14,686,289  | 14,332,008   |
| 10       | 5,440,093  | 7,740,873   | 13,180,966  | 23,008,130   |
| 11       | 1,638,058  | 2,442,228   | 4,080,286   | 6,852,064    |
| 12       | 2,472,548  | 4,518,875   | 6,991,424   | 5,802,386    |
| Total    | 64,857,158 | 67,384,899  | 132,242,057 | 326,246,276  |

Table 4.3: Output for the guidance pushbacks for the case study illustrated in Figure 4.5(a).

| Pushback | Ore (ton)  | Waste (ton) | Total (ton) | Profit (usd) |
|----------|------------|-------------|-------------|--------------|
| 1        | 7,513,800  | 775,246     | 8,289,046   | 71,297,820   |
| 2        | 9,703,558  | 7,077,838   | 16,781,396  | 71,023,539   |
| 3        | 5,846,181  | 5,236,801   | 11,082,982  | 35,444,713   |
| 4        | 6,682,204  | 3,157,441   | 9,839,645   | 36,490,832   |
| 5        | 7,612,939  | 9,122,411   | 16,735,349  | 20,888,005   |
| 6        | 3,079,056  | 4,163,800   | 7,242,856   | 13,952,326   |
| 7        | 6,212,199  | 6,288,881   | 12,501,080  | 21,412,152   |
| 8        | 5,074,258  | 12,551,850  | 17,626,108  | 6,570,015    |
| 9        | 5,644,414  | 9,673,204   | 15,317,618  | 17,579,501   |
| 10       | 7,063,028  | 9,269,711   | 16,332,739  | 30,424,800   |
| Total    | 64,431,637 | 67,317,184  | 131,748,820 | 325,083,702  |

Table 4.4: Material content of optimised practical pushbacks illustrated in Figure 4.5(b).

Finally, we manually generated a simple bench/pushback schedule for the optimised practical pushbacks described in Table 4.4, using the scheduling parameters shown in Table 4.1. This schedule assumes that no stockpiles or cut-off grade optimisation have been used; the mining production of each period is limited by either the maximum mining rate or the maximum processing plant capacity. An NPV of \$232,921,532 was obtained. Details are presented in Tables 4.5 and 4.6.

| Year\Pushback | P1    | P2     | P3     | P4    | P5     | P6    | P7     | P8     | P9     | P10    | Total (kton) |
|---------------|-------|--------|--------|-------|--------|-------|--------|--------|--------|--------|--------------|
| 1             | 8,289 | 11,065 | 2,270  | -     | -      | -     | -      | -      | -      | -      | 21,623       |
| 2             | -     | 5,717  | 7,392  | 4,835 | 3,846  | -     | -      | -      | -      | -      | 21,790       |
| 3             | -     | -      | 1,421  | 5,005 | 11,191 | 3,938 | -      | -      | -      | -      | 21,554       |
| 4             | -     | -      | -      | -     | 1,698  | 3,305 | 12,501 | 4,538  | -      | -      | 22,043       |
| 5             | -     | -      | -      | -     | -      | -     | -      | 13,088 | 8,876  | -      | 21,964       |
| 6             | -     | -      | -      | -     | -      | -     | -      | -      | 6,442  | 9,079  | 15,520       |
| 7             | -     | -      | -      | -     | -      | -     | -      | -      | -      | 7,254  | 7,254        |
| Total (kton)  | 8,289 | 16,781 | 11,083 | 9,840 | 16,735 | 7,243 | 12,501 | 17,626 | 15,318 | 16,333 | 131,749      |

Table 4.5: Material mined each year from each optimised practical pushback.

| Year       | Ore to process (ton) | Waste (ton) | Total (ton) | Profit (usd) |
|------------|----------------------|-------------|-------------|--------------|
| 1          | 11,972,781           | 9,650,644   | 21,623,425  | 86,410,099   |
| 2          | 12,053,538           | 9,736,887   | 21,790,425  | 76,535,404   |
| 3          | 12,007,384           | 9,546,947   | 21,554,330  | 55,061,119   |
| 4          | 10,843,978           | 11,198,596  | 22,042,574  | 44,078,908   |
| 5          | 6,237,434            | 15,726,238  | 21,963,673  | 6,626,572    |
| 6          | 6,212,382            | 9,307,822   | 15,520,204  | 19,278,384   |
| 7          | 5,104,139            | 2,150,051   | 7,254,190   | 37,093,217   |
| Total      | 64,431,637           | 67,317,184  | 131,748,820 | 325,083,702  |
| NPV(12.5%) |                      |             |             | 232,921,532  |

Table 4.6: Output of the schedule for optimised practical pushbacks in terms of ore, waste and profit (based on a 12.5% discount rate).

### 4.3.2 Traditional methodology

For comparison, we invited an independent mining consultant to design pushbacks for the same case study, using traditional methods. The traditional methodology starts by running a set of nested pits for a set of different metal selling prices.

| Pit shell | Selling price | Ore (ton)  | Waste (ton) | Total (ton) | Profit (usd) |
|-----------|---------------|------------|-------------|-------------|--------------|
| 32        | 0.64          | 78,412     | 20,004      | 98,417      | 1,345,085    |
| 35        | 0.70          | 76,050     | 14          | 76,064      | 1,086,687    |
| 37        | 0.74          | 71,206     | -           | 71,206      | 939,941      |
| 38        | 0.76          | 77,403     | 2,383       | 79,786      | 983,735      |
| 40        | 0.80          | 312,266    | 118,129     | 430,395     | 3,948,475    |
| 41        | 0.82          | 2,190,476  | 198,169     | 2,388,644   | 24,519,262   |
| 42        | 0.84          | 1,004,343  | 80,354      | 1,084,698   | 10,811,601   |
| 43        | 0.86          | 78,570     | -           | 78,570      | 785,977      |
| 44        | 0.88          | 1,924,977  | 425,208     | 2,350,185   | 19,723,585   |
| 45        | 0.90          | 774,125    | 22,271      | 796,396     | 7,271,383    |
| 46        | 0.92          | 1,896,462  | 277,503     | 2,173,964   | 17,085,204   |
| 47        | 0.94          | 454,646    | -           | 454,646     | 3,815,853    |
| 48        | 0.96          | 2,533,743  | 282,333     | 2,816,076   | 21,262,467   |
| 50        | 1.00          | 1,938,238  | 488,126     | 2,426,364   | 15,765,493   |
| 51        | 1.02          | 1,939,043  | 710,799     | 2,649,842   | 15,701,100   |
| 53        | 1.06          | 231,758    | 156,710     | 388,468     | 1,876,101    |
| 54        | 1.08          | 3,561,371  | 1,437,294   | 4,998,665   | 25,946,243   |
| 55        | 1.10          | 851,355    | 412,408     | 1,263,763   | 6,071,526    |
| 56        | 1.12          | 1,020,919  | 673,484     | 1,694,403   | 7,685,569    |
| 58        | 1.16          | 393,183    | 219,554     | 612,737     | 2,588,534    |
| 60        | 1.20          | 3,043,138  | 2,876,414   | 5,919,551   | 20,768,110   |
| 61        | 1.22          | 467,604    | 132,265     | 599,870     | 2,469,330    |
| 62        | 1.24          | 937,262    | 457,473     | 1,394,736   | 5,090,805    |
| 65        | 1.30          | 2,586,013  | 2,500,782   | 5,086,795   | 14,414,231   |
| 67        | 1.34          | 623,075    | 444,641     | 1,067,716   | 2,963,692    |
| 68        | 1.36          | 734,965    | 409,051     | 1,144,017   | 3,082,468    |
| 69        | 1.38          | 1,333,797  | 1,141,635   | 2,475,432   | 6,015,612    |
| 70        | 1.40          | 1,178,457  | 948,902     | 2,127,359   | 5,147,882    |
| 71        | 1.42          | 235,074    | 146,988     | 382,062     | 948,448      |
| 72        | 1.44          | 312,800    | 154,113     | 466,913     | 1,062,024    |
| 73        | 1.46          | 2,906,788  | 3,379,802   | 6,286,590   | 11,562,159   |
| 74        | 1.48          | 2,553,606  | 3,780,072   | 6,333,677   | 10,825,643   |
| 75        | 1.50          | 384,262    | 160,491     | 544,753     | 1,241,766    |
| 76        | 1.52          | 695,455    | 356,601     | 1,052,056   | 1,975,634    |
| 77        | 1.54          | 369,615    | 484,289     | 853,904     | 1,304,777    |
| 78        | 1.56          | 390,679    | 288,801     | 679,481     | 1,070,590    |
| 79        | 1.58          | 153,360    | 109,520     | 262,880     | 453,311      |
| 80        | 1.60          | 313,314    | 394,503     | 707,817     | 833,224      |
| 81        | 1.62          | 11,280,947 | 18,170,459  | 29,451,406  | 35,013,260   |
| 82        | 1.64          | 534,315    | 1,088,310   | 1,622,625   | 1,617,611    |
| 83        | 1.66          | 3,531,205  | 4,825,700   | 8,356,905   | 8,383,472    |
| 84        | 1.68          | 77,760     | -           | 77,760      | 111,191      |
| 85        | 1.70          | 78,570     | 156,247     | 234,817     | 254,157      |
| 86        | 1.72          | 543,064    | 1,094,565   | 1,637,628   | 1,271,545    |
| 87        | 1.74          | 705,982    | 518,885     | 1,224,866   | 1,060,902    |
| 88        | 1.76          | 158,437    | 149,640     | 308,077     | 269,078      |
| 89        | 1.78          | 623,365    | 549,142     | 1,172,508   | 834,369      |
| 90        | 1.80          | 1,930,912  | 4,274,510   | 6,205,422   | 3,176,980    |
| 91        | 1.82          | 545,417    | 1,029,542   | 1,574,959   | 791,831      |
| 92        | 1.84          | 1,698,569  | 3,263,645   | 4,962,214   | 2,517,491    |
| 94        | 1.88          | 3,295,293  | 8,762,110   | 12,057,402  | 3,368,765    |
| 95        | 1.90          | 459,304    | 604,616     | 1,063,920   | 301,671      |
| 96        | 1.92          | 391,445    | 673,665     | 1,065,111   | 321,029      |
| 98        | 1.96          | 235,711    | 197,683     | 433,395     | 68,106       |
| 100       | 2.00          | 315,628    | 753,394     | 1,069,022   | 13,364       |
| Total     | -             | 67,033,735 | 69,803,200  | 136,836,935 | 339,818,352  |

Table 4.7: Material content inside each pit shell.

In this case, a total of 100 copper prices from 0.0 to 2.0 (\$/pound) was selected. Table 4.7 summarises the set of nested pits obtained. It is important to notice that an increment in price does not necessarily result in a new shell because of the economic trade-off between the ore and the waste blocks. For example, pit shell 32 is the first profitable pit. Pit

shells 80 and 81 in Table 4.7 serves as an example of the big difference in volume between two adjacent pit shells, this problem is known in the literature as the *gap problem* and is one of the major shortcomings of the traditional methodology.

Only four guidance shells were created by the aggregation of the 55 pit shells, as is shown in Figure 4.6. The aggregation criteria were based on the material content and the distance between shells. Pits 32-54, 55- 80, 81-83 and 84-100 were selected as guidance shells 1, 2, 3 and 4 respectively. Table 4.8 summarises the material content and the profit of each guidance shell. Note that the aggregation process relies on the given criteria and the experience of the engineer who designs the pushbacks. In this case, no constraints were imposed regarding the number of pushbacks, leaving the engineer free to decide on the optimum number.

| Guidance    | Selling price interval | Ore (ton)  | Waste (ton) | Total (ton) | Profit (usd) |
|-------------|------------------------|------------|-------------|-------------|--------------|
| 1           | [0.00, 1.08]           | 19,143,089 | 4,219,298   | 23,362,387  | 172,868,195  |
| 2           | (1.08, 1.60]           | 21,484,722 | 19,471,789  | 40,956,512  | 107,575,335  |
| 3           | (1.60, 1.66]           | 15,346,467 | 24,084,468  | 39,430,936  | 45,014,343   |
| 4           | (1.66, 2.00]           | 11,059,456 | 22,027,645  | 33,087,101  | 14,360,479   |
| Grand total | [0.00, 2.00]           | 67,033,735 | 69,803,200  | 136,836,935 | 339,818,352  |

Table 4.8: Material and profit of each pushback used as pit guidance for the traditional methodology.

Next, a set of practical pushbacks were manually designed by the engineer (hereafter called manual practical pushbacks) based on the guidance shells. The time taken to design the set of four manual practical pushbacks was approximately 2 days. Table 4.9 summarises the material content and the un-discounted profit from each manual practical pushback. A closeness factor of 73.12% was obtained between the pit guidance and the manual practical pushbacks.

| Pushback | Ore (ton)  | Waste (ton) | Total (ton) | Profit (usd) |
|----------|------------|-------------|-------------|--------------|
| 1        | 16,444,690 | 6,083,000   | 22,527,690  | 141,987,215  |
| 2        | 21,081,099 | 27,441,657  | 48,522,757  | 95,546,168   |
| 3        | 15,347,445 | 22,462,835  | 37,810,280  | 36,395,239   |
| 4        | 14,160,500 | 13,815,708  | 27,976,208  | 65,889,730   |
| Total    | 67,033,735 | 69,803,200  | 136,836,935 | 339,818,352  |

Table 4.9: Material inside each manual practical pushback.

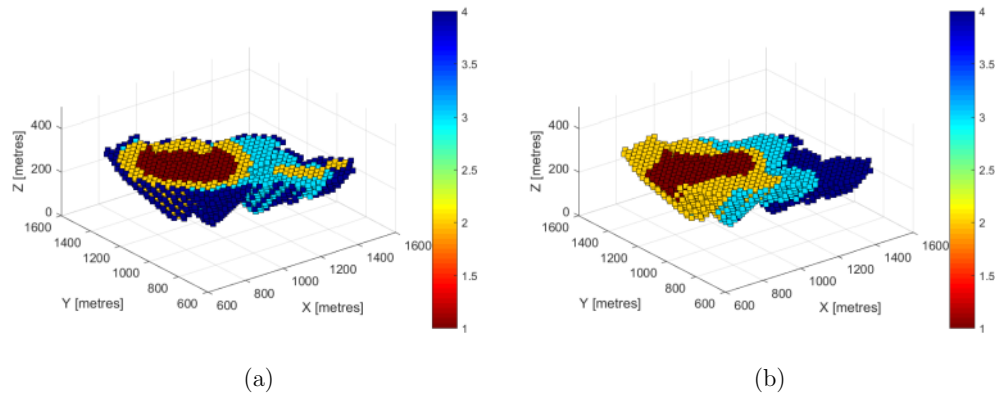


Figure 4.6: Isometric view of the guidance and optimiser practical pushbacks, colours represent different pushbacks (a) Guidance shells. (b) Manual practical pushbacks.

Finally, the mining engineer used the software *Datamine NPV Scheduler* to generate a bench/pushback schedule with the manual practical pushbacks described in Table 4.9 and scheduling parameters shown in Table 4.1. An NPV of \$221,034,886 was obtained for this case, \$11.89 million (5.4%) less than the NPV obtained using our optimised practical pushbacks. Details of the schedule are presented in Tables 4.10 and 4.11. However, the optimised practical pushbacks were able to produce ten pushbacks which give more flexibility to the scheduler to obtain a higher NPV. Therefore, to estimate which portion of the increment in the NPV came from increasing the number of pushbacks and which portion came from the better designs we produced a quick optimised practical pushback design by aggregating the ten pushbacks into four pushbacks, which gave an NPV of \$228,380,580 (see Table 4.13). In this case, we can say that \$4.54 million of the total increment in the NPV was produced by increasing the number of pushbacks and \$7.34 million came from the designs itself. Details of the aggregation and the four-pushbacks schedule are presented in Section 4.3.3.

| Year\Pushback | P1     | P2     | P3     | P4     | Total (kton) |
|---------------|--------|--------|--------|--------|--------------|
| 1             | 10,165 | 11,780 | -      | -      | 21,945       |
| 2             | 11,112 | 10,805 | -      | -      | 21,917       |
| 3             | 855    | 17,816 | 3,426  | -      | 22,097       |
| 4             | 396    | 8,122  | 13,562 | -      | 22,080       |
| 5             | -      | -      | 20,822 | 686    | 21,508       |
| 6             | -      | -      | -      | 22,057 | 22,057       |
| 7             | -      | -      | -      | 5,233  | 5,233        |
| Total (kton)  | 22,528 | 48,523 | 37,810 | 27,976 | 136,837      |

Table 4.10: Material mined each year from each pushback for the manual practical pushbacks.

| Year       | Ore to process (ton) | Waste (ton) | Total (ton) | Profit (usd) |
|------------|----------------------|-------------|-------------|--------------|
| 1          | 5,818,277            | 16,126,567  | 21,944,845  | 7,097,264    |
| 2          | 12,048,017           | 9,868,831   | 21,916,848  | 93,971,315   |
| 3          | 12,020,096           | 10,076,862  | 22,096,958  | 67,873,335   |
| 4          | 11,977,461           | 10,102,717  | 22,080,178  | 59,247,228   |
| 5          | 11,472,630           | 10,035,706  | 21,508,336  | 48,431,877   |
| 6          | 9,719,730            | 12,337,539  | 22,057,269  | 22,514,983   |
| 7          | 3,977,523            | 1,254,978   | 5,232,501   | 40,682,351   |
| Total      | 67,033,735           | 69,803,200  | 136,836,935 | 339,818,352  |
| NPV(12.5%) |                      |             |             | 221,034,886  |

Table 4.11: Output of the manual practical pushback schedule in terms of ore, waste and profit (based on a 12.5% discount rate).

#### 4.3.3 Schedule of aggregated optimised pushbacks

Old pushbacks (from Table 4.4) 1-3, 4-5, 6-8, 9-10 were selected as aggregated pushbacks 1, 2, 3, 4, respectively.

| Year\Pushback | P1     | P2     | P3     | P4     | Total (kton) |
|---------------|--------|--------|--------|--------|--------------|
| 1             | 22,025 | -      | -      | -      | 22,025       |
| 2             | 11,365 | 9,673  | -      | 871    | 21,908       |
| 3             | 2,764  | 13,404 | 4,938  | 981    | 22,086       |
| 4             | -      | 3,499  | 18,536 | -      | 22,035       |
| 5             | -      | -      | 13,897 | 7,971  | 21,867       |
| 6             | -      | -      | -      | 21,828 | 21,828       |
| Total (kton)  | 36,153 | 26,575 | 37,370 | 31,650 | 131,749      |

Table 4.12: Material mined each year from each optimised practical pushback.

| Year        | Ore to process (ton) | Waste (ton) | Total (ton) | Profit (usd) |
|-------------|----------------------|-------------|-------------|--------------|
| 1           | 10,398,033           | 11,627,091  | 22,025,124  | 60,167,534   |
| 2           | 11,944,607           | 9,963,329   | 21,907,936  | 82,810,723   |
| 3           | 11,797,739           | 10,288,058  | 22,085,797  | 56,885,646   |
| 4           | 10,643,988           | 11,390,589  | 22,034,577  | 31,941,503   |
| 5           | 11,070,932           | 10,796,155  | 21,867,088  | 57,780,910   |
| 6           | 8,576,338            | 13,251,962  | 21,828,300  | 35,497,386   |
| Total       | 64,431,637           | 67,317,184  | 131,748,820 | 325,083,702  |
| NPV (12.5%) |                      |             |             | 228,380,580  |

Table 4.13: Output of the schedule for optimised practical pushbacks in terms of ore, waste and profit (based on a 12.5% annual discount rate).

## 4.4 Conclusions

Although the pushback design problem has previously been considered in the literature, these studies have either used an oversimplified model or not included practical constraints in the problem formulation. In this study we introduced a mixed integer linear programming model for the practical pushback design problem, which allows the generation of practical or semi-practical pushbacks. Also, we established the closeness factor that quantifies the practicality of the program outputs. The closeness factor, as a metric, is independent of the optimisation model used.

Compared to the traditional methodology, computational results showed an improvement of \$11.89 million (5.4%) for the NPV of the project and a reduction in time from 2 days to 16.27 hours. Note that different scheduler has been used to create our mine schedule (a manual simple bench/pushback schedule), but one would expect that by using *Datamine NPV scheduler* in our designs a better NPV can be achieved. It is also worth noting that the time spent to design pushbacks using the traditional methodology is proportional to the number of pushbacks, so it can be expected that the engineer would take considerably more time to design the same number of pushbacks as our method produced (10). Furthermore, our model was shown to be capable of capturing the practical constraints and generating appropriate guidelines for practical pushbacks. These guidelines produced exceptional results for the closeness factor (93.07%) in contrast to the 73.12% obtained from the traditional methodology.

Our model provides practical constraints that can extend previous mixed integer linear programming formulations for the pushback design problem such as [7], [64] or [85]. The model proposed can automate the pushback design process capturing the engineer's experience. Most notably, the idea of using the ramp as a moving coordinate system to control the number and shape of the pushbacks can be extended to different types of formulations.

However, some limitations are worth noting. For general deposits, our formulation enforces the connectivity and mining width constraints by biasing the objective function, requiring an extra parameter  $\lambda$  (compactness factor) to be calibrated. Also, greedy designs may result in sub-optimality. A different approach, that never requires a compactness factor might be worth looking for. Also, other heuristics, like the Sliding Time Window Heuristic (see [24]), might improve the results obtained from the greedy approach. Finally, a direct implementation of our model only allows solving instances with a relatively small number of blocks (necessitating aggressive reblocking on large problem). Future work should therefore study methods to directly impose pushback width and connectivity constraints and scale the presented formulation, enabling the solution of large problem instances.

## 4.5 Summary

In open pit mines, the region within the excavation that maximises the mine's value is subdivided into manageable mining units called pushbacks which are mined to feed the processing plants. Practical pushbacks are considered to be workable volumes that: are connected; satisfy a minimum width to allow safe operation of the mining equipment; and have a haulage ramp to access the pushback from top to bottom. Existing models to design pushbacks tend to avoid the complexity of modelling these practical conditions. Consequently, most of the output from those models requires significant intervention by mining engineers. This work presents a new formulation to generate maximum value practical pushbacks. A closeness factor is introduced as a metric to quantify the practicality of the program output. Finally, a case study based on a real mine shows that our model produces more practical and better value pushbacks in less time than the traditional approach.

## Chapter 5

# Two mathematical models for the semi-practical pushback problem

*Pushbacks with width less than the operating width, no matter how optimal, would not be acceptable.*

— Mike Robb [86].

**O** PEN-PIT mining is a high-risk economic activity due to substantial upfront capital costs and fluctuations in input costs and commodity prices. Early production of high-grade ore is crucial to maximise NPV and minimise risk. Much value can be gained or lost based on the planning and design of mines and mining companies incorporate substantial discount factors into mine planning to focus investment on designs producing higher returns.

Open-pit mines are sequentially excavated to reach the ultimate pit limit. The sequencing is done by excavating smaller pits or *pushbacks*. Each pushback is comprised of an area with sufficient size and infrastructure to support open-pit mining operations (drilling, blasting, loading and hauling). An open-pit mine typically has between four and twenty pushbacks. In order to keep a balance between operational resources, capital expenditure, and speed of progress, the number of pushbacks that are active in mining operation at any point in time is generally between two and six.

For efficient mining production, each pushback should be a connected region, i.e. not have separated parts, and the bench width is set by the requirements of mining equipment for ore production and waste stripping. Additionally, each bench of the pushback should be accessible by a haulage ramp, which connects the working faces with the processing facilities and dumps outside the mine. Most mining engineers consider these to be the three most important practical conditions in their designs. However, the ramp access condition is often relaxed for the purpose of a quick economic evaluation of the pushback design. Thus a pushback that has a minimum mining width and that is connected is called a *semi-practical pushback* as has been defined in Chapter 1.

As shown in Section 2.2, the majority of the models used to consider the pushback design problem lack consideration of practical constraints. Consequently, the output from these models generally requires significant intervention by mining engineers in order to

find a set of pushbacks that are feasible, let alone close to optimal.

This chapter describes the semi-practical pushback design problem and presents and compares two mathematical formulations that produce optimal designs aimed at automating the process and reducing the need for engineering intervention. Additionally, we present computational results using a real mine instance and some modified instances from the publicly available library Minelib [34] (<http://mansci-web.uai.cl/minelib/>).

This chapter is organised as follows. In Section 5.1, a formal description of the semi-practical pushback problem is introduced. In Section 5.2 two mathematical formulations for this problem are presented. In Section 5.3, we compare the performance of both models using a small test instance. Section 5.4 introduces a couple of heuristic methods to solve medium size instance problems to near optimality. Finally, Section 5.5 presents two case studies based on real mines datasets.

## 5.1 The problem

The *semi-practical pushback design problem* consists of finding a partition of the blocks from the geological block model that maximises the economic benefit of the mining extraction. Each subset of the partition (hereafter called semi-practical pushbacks) must satisfy two practical constraints imposed by the mining technology being used. In the case of a truck-shovel system (the most common loading and hauling system in use), practical constraints are: 1) there should be a minimum mining width to ensure the safe operation of the shovels and the supporting equipment required at every bench, and 2) each pushback must be a connected set, in terms of horizontal and vertical adjacencies between blocks; and as such the slope conditions must be honoured. From now on, we use this as a formal definition for semi-practical pushbacks.

## 5.2 The mathematical models

In this section, two mathematical models that solve the semi-practical pushback design problem are introduced. The models aim to maximise an approximated NPV associated with the extraction of the pusbacks. Since the models do not account for a detailed production schedule, every pushback should yield a quantity of each mineral resource within a specified range. Unlike the approach taken in Section 4.1, these new mathematical formulations explicitly account for the mining width and connectivity constraints. The main difference between the models is the way the connectivity constraint is accounted for. In the first model we use an auxiliary reference point, that takes the role of a vertical ramp, for purposes of connectivity and shape, to which all blocks in the same pushback must be connected. Alternatively, the second model uses a set of horizontal networks for a given set of directions at each level of the block model to ensure the connectivity of the blocks within the same pushback.

### 5.2.1 The mathematical model 1: Connectivity to a ramp

The first mathematical formulation consists of a binary linear programming model (BLP) that finds a set of semi-practical pushbacks that maximises an approximate net present value (NPV) of the mining extraction. The inputs for the model are geological, which will include the geological block model and the dependencies between blocks based on the required wall slope conditions; operational inputs, which include the minimum operational width and bounds over the material content within the pushbacks; and economic such as, time-discount factors and profits associated with the blocks.

The minimum mining width constraint is modelled by considering a set of horizontal rectangular templates of given dimension that satisfies the mining width condition. The idea of using a rectangular template is then to impose that if a block is selected as part of a pushback, then all blocks from a rectangular template that contains that block must belong to that pushback. The set of rectangular templates  $\mathcal{W}$  is the collection of templates resulting from placing the horizontal template over each block considered for the model. For example, in Figure 5.1 if the block in red is selected for pushback  $p$  then either all blocks marked in black or all blocks marked in grey (or both) must be selected for pushback  $p$  as well.

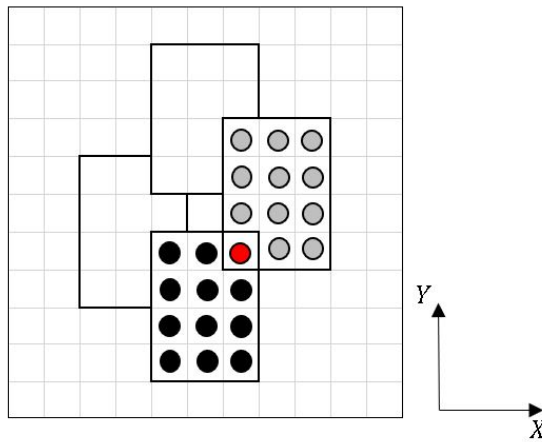


Figure 5.1: Example of horizontal rectangular templates, each template is represented as a black rectangle. If the block filled in red is chosen to belong to a pushback then either all blocks marked in black or all blocks marked in grey (or both) must belong to the same pushback.

The connectivity is modelled by forcing all blocks that belong to a pushback to be connected to a reference point (variable) which is determined during the optimisation. The wall slope condition is modelled as a precedence relationship between the blocks, which is standard practice in mine planning, as discussed in Chapters 3 and 4. Figure 5.2 shows a simple precedence relationship between cubic blocks to model a  $45^\circ$  slope angle. In order to mine block 6 it is required to have mined blocks 1 to 5. This model will be referred to as the Semi-Practical Pushback Model 1 ( $SPPM1_{BLP}$ ).

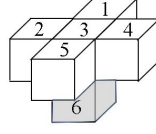


Figure 5.2: Illustration of a block precedence relationship to model a wall slope of  $45^\circ$ , assuming cubic blocks. Blocks 1-5 must be mined in order to mine block 6.

### Indices, sets and parameters

$b \in \mathcal{B}$  : blocks in the block model  $1, \dots, B$ . A LG-UPL pit boundary (as described in Chapter 2) could be used to reduce the size of the problem. However, there is no guarantee that the set of optimal pushbacks will be contained within the LG-UPL.

$p \in \mathcal{P}$  : pushbacks  $1, \dots, P$ , where  $P$  is the maximum number of pushbacks. In case the user does not want to impose a particular value of  $P$ , this can be calculated as follows  $P = \lceil \frac{\sum_{b \in \mathcal{B}} W_b}{\min_{p \in \mathcal{P}} \{\underline{W}_p\}} \rceil$ , parameters  $W_b$  and  $\underline{W}_p$  are explained below.

$d \in \mathcal{D}$  : set of distances from the reference point associated to pushback  $p$ .  $0, \dots, D$ . The distance is measured as the length of the path (measured in blocks) formed by a series of adjacent blocks that start at a reference point  $c$  which is determined for each pushback during the optimisation. The value of  $D$  also help to control the maximum spread of the pushback from the reference point.

$\hat{b} \in \hat{\mathcal{B}}_b$  : vertical precedences of block  $b$ .

$\bar{b} \in \bar{\mathcal{B}}_b$  : set of horizontal adjacent blocks of block  $b$  (this includes diagonally adjacent blocks).

$w \in \mathcal{W}$  : set of mining width rectangular templates (as shown in Figure 5.1).

$w' \in \mathcal{W}'_b$  : set of mining width rectangular templates that contain block  $b$ .

$b' \in \mathcal{B}'_w$  : set of blocks contained in the mining width rectangular template  $w$ .

$c \in \mathcal{C}$  : set of XY-coordinates that act as a reference point to ensure the connectivity of the blocks within a pushback. A priori the set  $\mathcal{C}$  is formed by the XY-projection of the  $(x, y, z)$  coordinates associated with all blocks  $b \in \mathcal{B}$ .

$V_b$  : economic value of of block  $b$  [US\$/ton] .

$W_b$  : amount of material contained in block  $b$  [tonnes].

$\underline{W}_p$  : minimum material content for pushback  $p$  [tonnes].

$\overline{W}_p$  : maximum material content for pushback  $p$  [tonnes].

$\delta_p$  : value discount factor for pushback  $p$ . The value of  $\delta_p$  implicitly assumes the timing of each pushback is known beforehand.  $\delta_p$  can be estimated as  $\frac{1}{(1+r)^t}$  where  $t$  is an estimate of the average period (in years) during which the pushback  $p$  is mined and  $r$  represents the financial discount rate.

$\mathcal{F}$  : a mapping function  $\mathcal{F} : \mathcal{B} \rightarrow \mathcal{C}$  such that  $\mathcal{F}(b) = c$  is the 2D projection over the XY plane of the spatial location of block  $b$ .

A simple way of defining the values  $\delta_p$  is to construct a *quick & dirty* pushback schedule based on the mining capacity and the number of simultaneous pushbacks that can be under operation at each period. For example, consider a mine with 140 million tonnes of material within the pit boundary, a mining capacity of 20 million tonnes per year and a discount rate of 8% per year. Assume that we want pushbacks that account for 2-3 years of production and no more than 2 pushbacks should be under operation at each period. In this case  $\underline{W}_p$  and  $\overline{W}_p$  will be 40 and 60 million tonnes, respectively. The maximum number of pushbacks will be 4 ( $P = \lceil \frac{140}{40} \rceil$ ) and the maximum number of periods for the schedule will be 7 ( $\lceil \frac{140}{20} \rceil$ ). Figure 5.3 shows a simple pushback schedule that satisfies the maximum number of pushbacks (2) under operation at each period. Then, the average mining period for pushbacks 1-4 is years 2, 3, 5, and 6, respectively. Therefore, the values of  $\delta_p$  are:  $\delta_1 = \frac{1}{(1+0.08)^2}$ ,  $\delta_2 = \frac{1}{(1+0.08)^3}$ ,  $\delta_3 = \frac{1}{(1+0.08)^5}$  and  $\delta_4 = \frac{1}{(1+0.08)^6}$ .

| Pushback | Period [years] |   |   |   |   |   |   |
|----------|----------------|---|---|---|---|---|---|
|          | 1              | 2 | 3 | 4 | 5 | 6 | 7 |
| P1       |                |   |   |   |   |   |   |
| P2       |                |   |   |   |   |   |   |
| P3       |                |   |   |   |   |   |   |
| P4       |                |   |   |   |   |   |   |

Figure 5.3: Example of a *quick & dirty* pushback schedule to estimate the value of parameters  $\delta_p$ .

### Decision variables

The first type of variable,  $x$ , is associated with the extraction of a block. The second type of variable,  $z$ , represents a selected reference point to control the shape of the pushback. The third type of variable,  $v$ , is an auxiliary variable that keeps record of the distance between a block and the reference point for the pushback containing that block. Finally, the fourth type of variable,  $u$  is used to track the rectangular templates that are used to satisfy the mining width constraints.

$$x_{bp} = \begin{cases} 1, & \text{if the block } b \text{ belongs to pushback } p \\ 0, & \text{otherwise} \end{cases}$$

$$z_{cp} = \begin{cases} 1, & \text{if the coordinate } c \text{ is selected as a reference point for pushback } p \\ 0, & \text{otherwise} \end{cases}$$

$$v_{bdp} = \begin{cases} 1, & \text{if the block } b \text{ is at distance } d \text{ from a coordinate } c \text{ } (d = f(c)), \text{ both of pushback } p. \\ 0, & \text{otherwise} \end{cases}$$

$$u_{wp} = \begin{cases} 1, & \text{if mining width template } w \text{ is assigned to pushback } p \\ 0, & \text{otherwise} \end{cases}$$

Figure 5.4 illustrates the relationship between the variables  $x$  and  $u$ ; and  $x$ ,  $v$  and  $z$ .

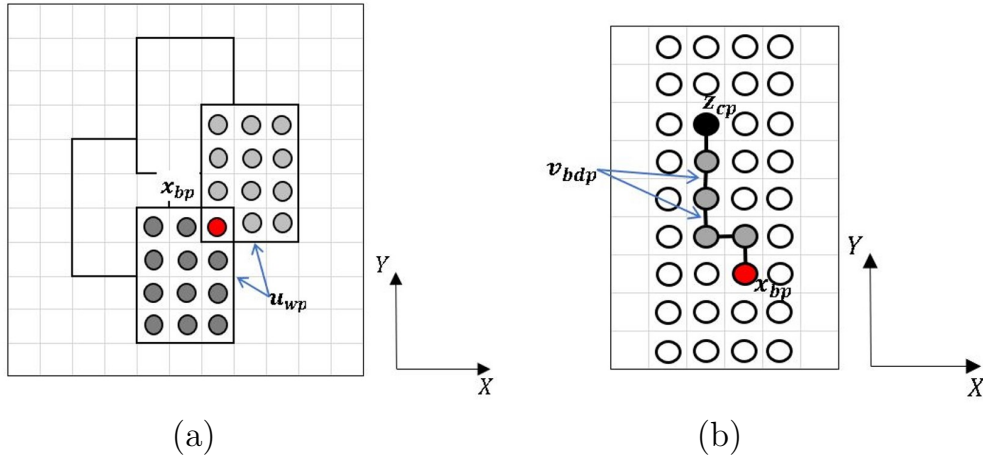


Figure 5.4: (a) Example of the relationship between variables  $x$  and  $u$ . If the block  $b$  (in red) belongs to pushback  $p$ , then at least one of the rectangular templates  $w$  which contains  $b$  must belong to the same pushback; forcing all the blocks within that (those) template(s) (blocks marked in grey and black) to belong to  $p$  as well. (b) Illustration of the relationship between variables  $v$ ,  $x$  and  $z$ . The variable  $v_{bdp}$  is one if there is a path of blocks (marked in grey) that belong to pushback  $p$  of length less than or equal to  $d$  from the selected coordinate  $c$  (marked as black) and the block  $b$  (marked as red).

### Formulation of the $SPPM1_{BLP}$

The objective function is defined as follows:

$$\text{Max} \sum_{p \in \mathcal{P}} \sum_{b \in \mathcal{B}} \delta_p \cdot V_b \cdot W_b \cdot x_{bp}$$

The objective function maximises the time-weighted total profit of the mining extraction. This gives an approximation of the net present value (NPV) of the mine extraction for an appropriate choice of  $\delta_p$ . Blocks are weighted by a factor that is a decreasing func-

tion of the index of the pushback to which they belong. Thus, we are taking into account the time expended by the pushback extraction.

$$\sum_{p \in \mathcal{P}} x_{bp} \leq 1 \quad b \in \mathcal{B} \quad (5.1)$$

$$\underline{W}_p \leq \sum_{b \in \mathcal{B}} W_b \cdot x_{bp} \leq \overline{W}_p \quad p \in \mathcal{P} \quad (5.2)$$

Constraint (5.1) implies that mined blocks can only belong to one pushback. Constraint (5.2) gives bounds on the amount of material contained in each pushback.

### Geotechnical constraints

$$\sum_{\rho \leq p} x_{b\rho} \leq \sum_{\rho \leq p} x_{\hat{b}\rho} \quad b \in \mathcal{B}, \quad \hat{b} \in \hat{\mathcal{B}}_b, \quad p \in \mathcal{P} \quad (5.3)$$

Constraint (5.3) ensures the vertical precedences are honoured in the extraction of any block. Note that Constraint (5.3) is equivalent to  $x_{bp} \leq \sum_{\rho \leq p} x_{\hat{b}\rho}$ , but Constraint (5.3) gives a tighter bound in the linear relaxation of the BLP model (see [13]).

### Connectivity constraints

$$x_{bp} \leq \sum_{d=\{0,\dots,D\}} v_{bdp} \quad b \in \mathcal{B}, \quad p \in \mathcal{P} \quad (5.4)$$

$$v_{b(d+1)p} \leq \sum_{\bar{b} \in \bar{\mathcal{B}}_b} v_{\bar{b}dp} \quad b \in \mathcal{B}, \quad d = \{0, \dots, D-1\}, \quad p \in \mathcal{P} \quad (5.5)$$

$$v_{b0p} \leq z_{cp} \quad b \in \mathcal{B}, \quad c = \mathcal{F}(b), \quad p \in \mathcal{P} \quad (5.6)$$

$$v_{bdp} \leq x_{bp} \quad b \in \mathcal{B}, \quad d \in \mathcal{D}, \quad p \in \mathcal{P} \quad (5.7)$$

$$\sum_{c \in \mathcal{C}} z_{cp} \leq 1 \quad p \in \mathcal{P} \quad (5.8)$$

Constraint (5.4) implies that block  $b$  of pushback  $p$  must be within distance  $D$  from the reference point  $c$ . Constraint (5.5) means that a block  $b$  can only be classified as having distance  $d+1$  from a reference point if one of its neighbouring blocks has distance  $d$  from the reference point. Constraint (5.6) sets as zero the distance  $d$  between block  $b$  to a reference point  $c$ . Constraint (5.7), together with the previous two constraints, ensures the connectivity between the blocks within a pushback from the chosen reference point. Constraint (5.8) forces only one reference point to be chosen per pushback.

### Mining width constraints

$$x_{bp} \leq \sum_{w' \in \mathcal{W}'_b} u_{w'p} \quad \forall b \text{ such that } |\mathcal{W}'_b| > 0, \quad p \in \mathcal{P} \quad (5.9)$$

$$u_{wp} \leq x_{b'p} \quad p \in \mathcal{P}, \quad w \in \mathcal{W}, \quad b' \in \mathcal{B}'_w \quad (5.10)$$

$$\sum_{p \in \mathcal{P}} u_{wp} \leq 1 \quad w \in \mathcal{W} \quad (5.11)$$

Constraint (5.9) states that if a block is assigned to a pushback then at least one mining width template that contains that block must be assigned to that pushback. Constraint (5.10) forces all blocks within a mining width template  $w$  that belongs to a pushback  $p$  to belong to the same pushback. Finally, Constraint (5.11) ensures that any mining width template is assigned to at most one pushback.

The main limitation of the  $SPPM1_{BLP}$  is that the number of variables  $v_{bdp}$  is proportional to the maximum distance  $D$  imposed. The distance  $D$  is not a requirement for the semi practical pushback problem and for each increment of the value of  $d$  the number of binary variables  $v_{bdp}$  increases by  $B \times P$ . In practice the maximum distance  $D$  represents the maximum distance of any pushback at the upper level of the block model, since the horizontal connectivity is ensured level by level.

### 5.2.2 The mathematical model 2: Horizontal network connectivity

Like the model introduced in Section 5.2.1, the second mathematical formulation consists of a binary linear programming model (BLP) that finds a set of semi-practical pushbacks that maximises an approximate net present value (NPV) of the mining extraction. However, in this case, the connectivity is modelled by using a set of graphs  $\mathcal{G}_l$ , such that each graph represents a horizontal line  $l$  at each level of the block model. The lines  $l$  follow a set of given directions  $\vec{d}$ . For example, Figure 5.5 illustrates a set directions  $\vec{d}$  following coordinates  $\vec{X}$  and  $\vec{Y}$ .

The idea of using a networks to model connectivity constraints has been previously introduced in Chapter 3, Sections 3.1 and 3.2. A natural approach would be to use a single network that connects all horizontal neighbours of a block at each level. However, the single network per level approach might produce disconnected pushbacks if no sub-tour elimination is also imposed. To overcome the sub-tour elimination problem, a new approach subdividing the single horizontal network into a number of unidirectional networks is taken. Because of the unidirectionality of the networks no cycles are possible, and therefore, no sub-tour elimination is required.

The construction of a graph  $\mathcal{G}_l$  is as follows. First, add two virtual vertices  $S_l$  and  $T_l$  to  $\mathcal{V}_l$ ; second, for each block in the line  $l$  add a corresponding vertex  $i$  to  $\mathcal{V}_l$ . Then, for each vertex  $i$  in  $\mathcal{V}_l$  such that  $i \neq S_l$  and  $i \neq T_l$  add the arcs  $(S_l, i)$  and  $(i, T_l)$  to  $\mathcal{A}_l$ . Finally, for each pair of vertices  $i, j$  ( $\neq S_l$  and  $\neq T_l$ ) such that  $i, j \in \mathcal{V}_l$  and  $j$  is the immediate

neighbour of  $i$  for a given direction  $\vec{d}$ , add the arc  $(i, j)$  to  $\mathcal{A}_l$ . Figure 5.5 illustrates a simple case of graph  $\mathcal{G}_l$ .

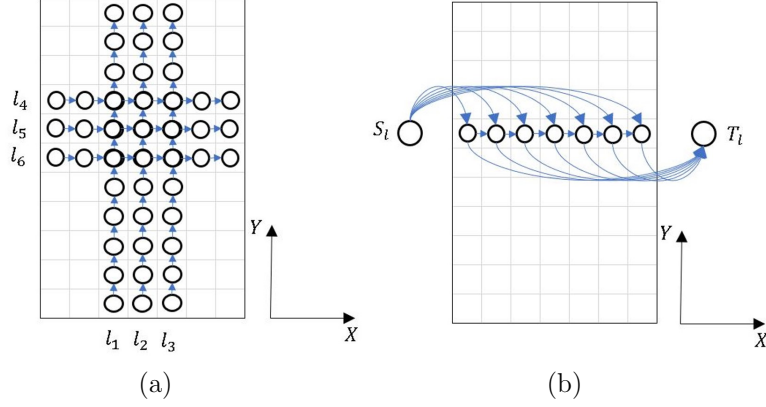


Figure 5.5: (a) Example of a set of lines  $l$  following directions  $\vec{X}$  and  $\vec{Y}$  for a level. (b) Example of a simple graph  $\mathcal{G}_l$ . Arcs from  $S_l$  to the blocks and from the blocks to  $T_l$  are represented as bowed arrows, while arcs that connect blocks with their immediate neighbours are drawn as straight arrows.

The connectivity is ensured only over the directions  $\vec{d}$  chosen, so unless many directions are used this approach does not guarantee connectivity (but makes it extremely likely). In practice, we have seen that four directions ( $\vec{X}$ ,  $\vec{Y}$ ,  $\vec{X}\vec{Y}$  and  $\vec{Y}\vec{X}$ ) are usually enough to get connected shapes in massive (not scattered) ore deposit. However, considering a large number of directions poses some limitations such as the increase of the number of binary variables and limiting the shape of the pushbacks. A more detailed explanation of these limitations is given at the end of this section.

As in Section 5.2.1, the minimum mining width constraint is modelled by considering horizontal rectangular templates for which the dimensions are given beforehand. This model will be referred as the Semi-Practical Pushback Model 2 ( $SPPM2_{BLP}$ ). The description for the indices, sets and parameters given in Section 5.2.1 applies to almost all the items described except for the set  $\mathcal{L}$ .

### Indices, sets and parameters

In addition to the graphs  $\mathcal{G}_l(\mathcal{V}_l, \mathcal{A}_l)$ , the following sets and parameters are required to formulate the  $SPPM2_{BLP}$ .

$b \in \mathcal{B}$  : blocks  $1, \dots, B$ .

$p \in \mathcal{P}$  : pushbacks  $1, \dots, P$ .

$l \in \mathcal{L}$  : lines  $1, \dots, L$ , in a given set of horizontal directions for which the connectivity will be imposed.

$\hat{b} \in \hat{\mathcal{B}}_b$  : vertical precedences of block  $b$ .

$\bar{b} \in \bar{\mathcal{B}}_b$  : set of horizontal adjacent blocks of block  $b$ .

$w \in \mathcal{W}$  : set of mining width rectangular templates.

$w' \in \mathcal{W}'_b$  : set of mining width rectangular templates that contains block  $b$ .

$b' \in \mathcal{B}'_w$  : set of blocks contained in the mining width rectangular template  $w$ .

$V_b$  : economic value [US\$/ton] of block  $b$ .

$W_b$  : material contained in block  $b$  [tonnes].

$\underline{W}_p$  : minimum material content for pushback  $p$  [tonnes].

$\overline{W}_p$  : maximum material content for pushback  $p$  [tonnes].

$\delta_p$  : discount factor for pushback  $p$ , which is a function of when pushback  $p$  is extracted.

### Decision variables

The first variable,  $x$ , is associated with the extraction of the block. The second variable,  $a$ , keeps track of the arcs in  $\mathcal{G}_l$  that have been assigned to a pushback. Finally, the third variable,  $u$ , is associated to the mining width templates. Note that the variable  $x$  is introduced as an auxiliary variable for illustration purposes, and it can be replaced by an appropriate set of variables  $a$  as is shown in Constraint (5.18).

$$x_{bp} = \begin{cases} 1, & \text{if the block } b \text{ belongs to pushback } p \\ 0, & \text{otherwise} \end{cases}$$

$$a_{ijp} = \begin{cases} 1, & \text{if the arc } (i, j) \text{ is selected for pushback } p \\ 0, & \text{otherwise} \end{cases}$$

$$u_{wp} = \begin{cases} 1, & \text{if mining width template } w \text{ is assigned to pushback } p \\ 0, & \text{otherwise} \end{cases}$$

**Formulation of the  $SPPM2_{BLP}$** 

The objective function is defined as follows:

$$\text{Max} \sum_{p \in \mathcal{P}} \sum_{b \in \mathcal{B}} \delta_p \cdot V_b \cdot W_b \cdot x_{bp}$$

As in the  $SPPM1_{BLP}$ , the objective function maximises the time-weighted total profit of the mining extraction, or the net present value (NPV) of the mine extraction for an appropriate choice of  $\delta_p$ . Blocks are weighted by a factor that is a decreasing function of the index of the pushback to which they belong. Thus, we are taking into account the time expended by the pushback extraction.

$$\sum_{p \in \mathcal{P}} x_{bp} \leq 1 \quad b \in \mathcal{B} \quad (5.12)$$

$$\underline{W}_p \leq \sum_{b \in \mathcal{B}} W_b \cdot x_{bp} \leq \overline{W}_p \quad p \in \mathcal{P} \quad (5.13)$$

Constraint (5.12) implies that mined blocks can belong to only one pushback. Constraint (5.13) gives bounds to the material within each pushback.

**Geotechnical constraints**

$$\sum_{\rho \leq p} x_{b\rho} \leq \sum_{\rho \leq p} x_{\hat{b}\rho} \quad b \in \mathcal{B}, \hat{b} \in \hat{\mathcal{B}}_b, p \in \mathcal{P} \quad (5.14)$$

Constraint (5.14) ensures the vertical precedences are honoured in the extraction of any block. Note that Constraint (5.14) is equivalent to  $x_{bp} \leq \sum_{\rho \leq p} x_{\hat{b}\rho}$ , but Constraint (5.14) gives a tighter bound in the linear relaxation of the BLP model.

**Connectivity constraints**

$$\sum_{(i,j) \in \mathcal{A}_l | i \neq S_l, j \neq T_l} a_{ijp} - \sum_{(j,k) \in \mathcal{A}_l | j \neq S_l, k \neq T_l} a_{jkp} = 0, \quad l \in \mathcal{L}, \quad p \in \mathcal{P}, j \in \mathcal{B} \quad (5.15)$$

$$\sum_{(i,j) \in \mathcal{A}_l} a_{ijp} - \sum_{(j,k) \in \mathcal{A}_l} a_{jkp} \leq \begin{cases} -1, & \text{if } j = S_l \\ 1, & \text{if } j = T_l \end{cases} \quad l \in \mathcal{L}, \quad p \in \mathcal{P}, j \in \mathcal{B} \cup \{S_l, T_l\} \quad (5.16)$$

$$\sum_{p \in \mathcal{P}} a_{ijp} \leq 1 \quad (i,j) \in \mathcal{A}_l, \quad l \in \mathcal{L} \quad (5.17)$$

$$\sum_{(i,j) \in \mathcal{A}_l, j \neq T_l} a_{ijp} = x_{jp} \quad \forall j \neq T_l | (i,j) \in \mathcal{A}_l, l \in \mathcal{L}, \quad p \in \mathcal{P} \quad (5.18)$$

Constraints (5.15) and (5.16) ensure the flow conservation between the arcs in the graph  $\mathcal{G}_l$  for all the lines that have at least one block belonging to a pushback  $p$ . Constraint (5.17) ensures that each arc  $(i, j)$  is assigned to at most one pushback. Constraint (5.18) establishes the equivalence between the set of arcs that end in a block and the extraction of the same block.

### Mining width constraints

$$x_{bp} \leq \sum_{w' \in \mathcal{W}'_b} u_{w'p} \quad \forall b \text{ such that } |\mathcal{W}'_b| > 0, p \in \mathcal{P} \quad (5.19)$$

$$u_{wp} \leq x_{b'p} \quad p \in \mathcal{P}, w \in \mathcal{W}, b' \in \mathcal{B}'_w \quad (5.20)$$

$$\sum_{p \in \mathcal{P}} u_{wp} \leq 1 \quad w \in \mathcal{W} \quad (5.21)$$

Constraint (5.19) states that if a block is assigned to a pushback then at least one mining width template that contains that block must be assigned to that pushback. Constraint (5.20) forces all blocks within a mining width template  $w$  that belong to a pushback  $p$  to belong to the same pushback. Finally, Constraint (5.21) ensures that any mining width template is assigned to at most one pushback.

There are three main limitations of the  $SPPM2_{BLP}$  formulation. First, the use of pre-defined directions to model the connectivity limits the possible shapes for the pushbacks. For example, shapes that are connected but have a separation along a direction  $\vec{d}$  cannot be produced by using this approach, since the method imposes connectivity in all the directions simultaneously. In other words, the  $SPPM2_{BLP}$  prevents the formation of non-convex shapes if only rectilinear directions are chosen, a small example is illustrated in Figure 5.6a. Second, the connectivity is ensured only over the directions  $\vec{d}$  chosen; for instance, if only directions  $\vec{X}$  and  $\vec{Y}$  are considered for the problem, shapes that are disconnected in direction  $\vec{X}\vec{Y}$  may occur as show in Figure 5.6b. Third, the number of the variables  $a_{ijp}$  will proportionally increase with the number of directions considered. In practice, we noticed that 2 to 4 directions ( $\vec{X}$ ,  $\vec{Y}$ ,  $\vec{X}\vec{Y}$ ,  $\vec{Y}\vec{X}$ ) were enough to ensure connectivity over the instances studied, but the number and directions sufficient to ensure connectivity will depend on each case.

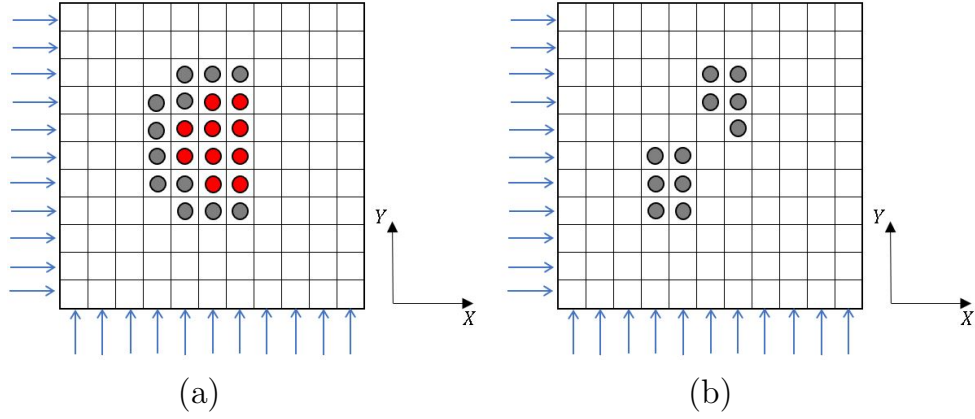


Figure 5.6: (a) Example of a non-convex shape (black) that cannot be obtained by imposing directions  $\vec{X}$  and  $\vec{Y}$  simultaneously. The resultant shape when using directions  $\vec{X}$  and  $\vec{Y}$  adds the section marked in red. (b) Example of disconnected shapes in direction  $\vec{X}\vec{Y}$  when only directions  $\vec{X}$  and  $\vec{Y}$  are considered.

### 5.3 A computational comparison between models $SPPM1_{BLP}$ and $SPPM2_{BLP}$

In this section a small test instance ( $I_1$ ), based on the *Newman* instance available in Minelib [34], is used to compare the computational performance of models  $SPPM1_{BLP}$  and  $SPPM2_{BLP}$ . The instance  $I_1$  consists of a block model with 273 blocks, with a total tonnage of 1,546,272 tons within the ultimate pit limit and a total value of \$14,292,360. The optimisation parameters chosen to test the model are: mining width templates of 2 by 2 blocks, a discount factor of 20% and a maximum of four pushbacks. The minimum tonnage for each pushback is set to 100,000 tonnes, while the maximum tonnage is constrained to 500,000 tonnes. Four horizontal neighbours were considered to create the set  $\vec{B}$ , and a distance  $D = 10$  was selected for the formulation of the  $SPPM1_{BLP}$ . In the case of the  $SPPM2_{BLP}$  formulation, directions  $\vec{X}$  and  $\vec{Y}$  were considered to create the set of lines  $\mathcal{L}$ .

Both models  $SPPM1_{BLP}$  and  $SPPM2_{BLP}$  were implemented in Python 2.7 and solved using Gurobi 7.5 MILP-solver. All the pre-processing routines, such as the definition of the sets and variables and the early and late start algorithm were done in Python. Execution of the code was done on an Intel Xeon machine with 32 gigabytes running Windows 10. This machine has eight processors that run at 2.40 gigahertz. The results summarised in Table 5.1 show that the formulation  $SPPM2_{BLP}$  uses around half the number of binary variables and constraints used by  $SPPM1_{BLP}$  and it takes around a quarter of the time to solve the instance  $I_1$ .

|               | No of binary variables | No of constraints | Obj. Function [US\$] | Time [sec] |
|---------------|------------------------|-------------------|----------------------|------------|
| $SPPM1_{BLP}$ | 13,952                 | 30,825            | 7,180,116            | 31.9       |
| $SPPM2_{BLP}$ | 7,748                  | 15,270            | 7,180,569            | 8.4        |

Table 5.1: Computational comparison of models  $SPPM1_{BLP}$  and  $SPPM2_{BLP}$  for instance  $I_1$ .

## 5.4 Solution method

Although the models introduced in Sections 5.2.1 and 5.2.2 can be directly implemented and solved using commercial MILP-solvers, it can take several hours to solve small to medium sized instances to optimality. For that reason a set of preprocessing routines (early and late start) and a sliding window heuristic are proposed here.

### 5.4.1 Early and late start

The early and late start algorithms are preprocessing routines that allow the reduction of variables required for the problem by imposing bounds on the pushbacks to which the blocks can belong. This algorithms work by combining the upper and lower bounds of the tonnage constraints for each pushback with the precedence constraints to calculate these bounds. In particular, these algorithms compute the earlier and latest pushbacks to which any given block can belong.

The *early start* of a block  $b$  is defined as the division between the tonnage supported by block  $b$  (the sum of the total tonnage of the blocks above block  $b$  according to the precedence relationship), and the upper bound of the tonnage constraint ( $\bar{W}_p$ ). Similarly, the late start of a block  $b$  is computed by calculating the rate between the difference of the total material inside the pit minus the total material below the block  $b$  and the lower bound of the tonnage constraint. A detailed explanation of the early and late start algorithms can be found in [39] pages 43-52. The computational complexity of these algorithms are bounded by  $O(nm)$ , where  $n$  correspond to the number of blocks and  $m$  is the number of precedence relationships between blocks (wall slope condition).

### 5.4.2 Sliding window heuristic

The sliding window heuristic (SWH) introduced here is based on the sliding time window heuristic described in [24]. The main idea of the SWH is to repeatedly solve a partly relaxed sub-problem of a mixed integer linear programming model. At each iteration of the algorithm a sub-problem is set-up and solved, then a fraction of the variables of that sub-problem is fixed, and this is repeated until all variables of the mixed integer linear programming model have been fixed.

The first step of the SWH is to define three sequential subsets of  $\mathcal{P}$ :  $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ , such that  $|\mathcal{P}_1| + |\mathcal{P}_2| + |\mathcal{P}_3| \leq |\mathcal{P}|$ . The variables in the subset  $\mathcal{P}_1$  correspond to the variables that will be fixed with the solution values obtained at a previous iteration of the algorithm (variables marked with the star symbol, \*). The variables in the subset  $\mathcal{P}_2$  correspond to the integer window, which means that the variables in  $\mathcal{P}_2$  remain as integer. Finally, the integrality constraint for the variables in the subset  $\mathcal{P}_3$  is relaxed. Figure 5.7 illustrates the SWH for  $n$  iterations, the columns represent the sequential subsets of  $\mathcal{P}$  (black, blue and red colours are used to represent  $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ , respectively).

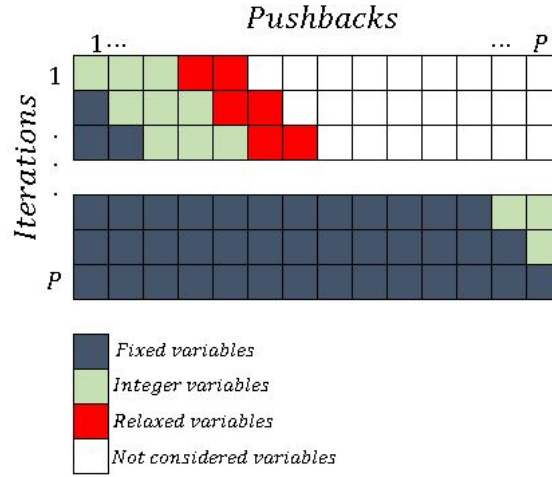


Figure 5.7: Illustration of the SWH with  $Iwin = 3$  (the number of consecutive pushbacks in each integer window) and  $Cwin = 2$  (the number of pushbacks in each continuous window, with the integrality of the variables relaxed), black, green and red colours represent  $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ , respectively.

The following is a description of the SWH applied to the  $SPPM2_{BLP}$  formulation, since the  $SPPM2_{BLP}$  performed better than the  $SPPM1_{BLP}$  in the experiment shown in Section 5.3. However, the framework of the SWH for the  $SPPM1_{BLP}$  is the same. Lines 1–2 initialise the set  $\mathcal{P}_1$  as empty. Then, the loop from iteration 1 to  $P$  begins at Line 3. In Lines 4–7 the solutions of the variables  $x, a$  and  $u$  are initialised as empty. In Lines 8–11, the sets  $\mathcal{P}_2$  and  $\mathcal{P}_3$  are defined with respect to the iteration  $i$ . Line 12 formulates the  $SPPM1_{sub}$  (described by the Equations 5.22 to 5.41) for the sets  $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, x_{bp}^*, a_{ijp}^*, u_{wp}^*$ . Then, Line 13 solves the  $SPPM2_{sub}$  and finds the value of the variables  $x_{bp}, a_{ijp}$  and  $u_{wp}$  for  $p \in \mathcal{P}_1$  and stores them in the sets  $x_{bp}^*, a_{ijp}^*, u_{wp}^*$ . Finally,

Lines 15–16 update the set  $\mathcal{P}_1$  and the value of  $P_1$ .

---

**Algorithm 1:** Sliding window heuristic algorithm
 

---

**Data:** integer window:  $Iwin$ , continuous window:  $Cwin$

```

1 begin
2    $\mathcal{P}_1 \leftarrow \emptyset$ 
3    $P_1 \leftarrow 0$ 
4   foreach  $i \in \{1, \dots, P\}$  do
5      $x_{bp}^* \leftarrow \emptyset$ 
6      $a_{ijp}^* \leftarrow \emptyset$ 
7      $u_{wp}^* \leftarrow \emptyset$ 
8      $P_2 \leftarrow \min(P, i + Iwin - 1)$ 
9      $P_3 \leftarrow \min(P, i + Iwin + Cwin)$ 
10     $\mathcal{P}_2 \leftarrow \{P_1 + 1, \dots, P_2\}$ 
11     $\mathcal{P}_3 \leftarrow \{P_2 + 1, \dots, P_3\}$ 
12    formulate  $SPPM2_{sub}(\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, x_{bp}^*, a_{ijp}^*, u_{wp}^*)$ 
13    solve  $SPPM2_{sub}$ 
14     $x_{bp}^*, a_{ijp}^*, u_{wp}^* \leftarrow$  get solution from  $SPPM2_{sub}$  such that  $p \in \mathcal{P}_1$ 
15     $\mathcal{P}_1 \leftarrow \mathcal{P}_1 \cup \{i\}$ 
16     $P_1 \leftarrow P_1 + 1$ 

```

---

The parameter  $Iwin$  can take any value in the set  $\{1, \dots, P\}$  and the parameter  $Cwin$  can take any value in the set  $\{0, \dots, P - 1\}$  such that  $Iwin + Cwin \leq P$ . Two extreme situations are worth noting, 1) when  $Iwin = P$  the sliding window is equivalent to solving the BLP formulation, and 2) when  $Iwin = 1$  and  $Cwin = 0$  the sliding window is equivalent to a greedy approach. It is worth noting that large size instances might not be solved for values of  $Iwin$  close to  $P$ . Also, values of  $Iwin$  and  $Cwin$  such that  $Iwin + Cwin \ll P$  are unlikely to lead to a feasible solution for the BLP model.

**Formulation for the  $SPPM2_{sub}(\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, x_{bp}^*, z_{cp}^*, u_{bdp}^*)$**

$$\text{Max} \sum_{p \in \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3} \sum_{b \in \mathcal{B}} \delta_p \cdot V_b \cdot W_b \cdot x_{bp} \quad (5.22)$$

$$\sum_{p \in \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3} x_{bp} \leq 1 \quad b \in \mathcal{B} \quad (5.23)$$

$$\underline{W}_p \leq \sum_{b \in \mathcal{B}} W_b \cdot x_{bp} \leq \overline{W}_p \quad p \in \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3 \quad (5.24)$$

$$\sum_{\rho \leq p} x_{b\rho} \leq \sum_{\rho \leq p} x_{\hat{b}\rho} \quad b \in \mathcal{B}, \hat{b} \in \hat{\mathcal{B}}_b, p \in \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3 \quad (5.25)$$

$$\sum_{(i,j) \in \mathcal{A}_l | i \neq S_l, j \neq T_l} a_{ijp} - \sum_{(j,k) \in \mathcal{A}_l | j \neq S_l, k \neq T_l} a_{jkp} = 0, \quad l \in \mathcal{L}, p \in \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3, j \in \mathcal{B} \quad (5.26)$$

$$\sum_{(i,j) \in \mathcal{A}_l} a_{ijp} - \sum_{(j,k) \in \mathcal{A}_l} a_{jkp} \leq \begin{cases} -1, & \text{if } j = S_l \\ 1, & \text{if } j = T_l \end{cases} \quad l \in \mathcal{L}, p \in \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3, j \in \mathcal{B} \cup \{S_l, T_l\} \quad (5.27)$$

$$\sum_{p \in \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3} a_{ijp} \leq 1 \quad (i, j) \in \mathcal{A}_l, l \in \mathcal{L} \quad (5.28)$$

$$\sum_{(i,j) \in \mathcal{A}_l, j \neq T_l} a_{ijp} = x_{jp} \quad l \in \mathcal{L}, p \in \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3 \quad (5.29)$$

$$x_{bp} \leq \sum_{w' \in \mathcal{W}'_b} u_{w'p} \quad \forall b \text{ such that } |\mathcal{W}'_b| > 0, p \in \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3 \quad (5.30)$$

$$u_{wp} \leq x_{b'p} \quad p \in \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3, w \in \mathcal{W}, b' \in \mathcal{B}'_w \quad (5.31)$$

$$\sum_{p \in \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3} u_{wp} \leq 1 \quad w \in \mathcal{W} \quad (5.32)$$

$$x_{bp} \equiv x_{bp}^* \quad p \in \mathcal{P}_1, b \in \mathcal{B}, \quad (5.33)$$

$$a_{ijp} \equiv a_{ijp}^* \quad p \in \mathcal{P}_1, (i, j) \in \mathcal{A}_l, l \in \mathcal{L} \quad (5.34)$$

$$u_{wp} \equiv u_{wp}^* \quad p \in \mathcal{P}_1, w \in \mathcal{W} \quad (5.35)$$

$$x_{bp} \in \{0, 1\} \quad p \in \mathcal{P}_2, b \in \mathcal{B}, \quad (5.36)$$

$$a_{ijp} \in \{0, 1\} \quad p \in \mathcal{P}_2, (i, j) \in \mathcal{A}_l, l \in \mathcal{L} \quad (5.37)$$

$$u_{wp} \in \{0, 1\} \quad p \in \mathcal{P}_2, w \in \mathcal{W} \quad (5.38)$$

$$0 \leq x_{bp} \leq 1 \quad p \in \mathcal{P}_3, b \in \mathcal{B}, \quad (5.39)$$

$$0 \leq a_{ijp} \leq 1 \quad p \in \mathcal{P}_3, (i, j) \in \mathcal{A}_l, l \in \mathcal{L} \quad (5.40)$$

$$0 \leq u_{wp} \leq 1 \quad p \in \mathcal{P}_3, w \in \mathcal{W} \quad (5.41)$$

## 5.5 Computational experiments

In this section the sliding window heuristic is used to solve larger sized instances (from a few hundred to a couple of thousands blocks). Parameters  $Iwin$  and  $Cwin$  control the performance of the quality and time of the sliding window heuristic. Motivated by the results obtained in Section 5.3, we are only considering the  $SPPM2_{BLP}$  formulation for the experiments.

Section 5.5.1 presents a sensitivity analysis for parameters  $Iwin$  and  $Cwin$  that give a better understanding of the influence of these parameters over the quality of the solution. A small instance is considered because the optimal solution can be computed. In Section 5.5.2 two larger instances (up to 4600 blocks) are considered to evaluate the performance in time of the sliding window heuristic. Additionally, the result of our method is compared with the set of pushbacks obtained without considering connectivity and mining width constraints.

### 5.5.1 Sensitivity analysis of parameters $Iwin$ and $Cwin$ for instance $I_1$

The mining instance  $I_1$  (based on the *Newman* instance from Minelib [34]) is considered in order to study the impact of the parameters  $Iwin$  and  $Cwin$  on the quality of the solution. A combination of eight pairs of values ( $Iwin$ ,  $Cwin$ ) such that  $Iwin + Cwin \leq P$  ( $P=4$ ) is selected for the experiments. The set of experiments are notated as  $I_1(1,0)$ ,  $I_1(1,1)$ ,  $I_1(1,2)$ ,  $I_1(1,3)$ ,  $I_1(2,1)$ ,  $I_1(2,2)$ ,  $I_1(3,1)$ , where the label  $I_1(Iwin, Cwin)$  stands for instance  $I_1$  with parameters  $Iwin$  and  $Cwin$ . Also,  $I_1 Opt$  represents the optimal solution of  $SPPM2_{BLP}$  for the instance  $I_1$ . Similarly,  $I_1 LP$  represents the solution of the linear relaxation of  $SPPM2_{BLP}$  for the instance  $I_1$ .

Table 5.2 summarises the value of the objective function, running time and linear relaxation gap of the experiments. The configurations  $I_1(1,0)$ ,  $I_1(1,1)$ ,  $I_1(2,1)$  did not provide a feasible solution for the  $SPPM2_{BLP}$ .

| Experiment | Obj. Function Value [US\$] | Running Time [sec] | linear relax. gap [%] |
|------------|----------------------------|--------------------|-----------------------|
| $I_1(1,2)$ | 7,128,174                  | 6.02               | 4.70                  |
| $I_1(1,3)$ | 7,126,804                  | 6.95               | 4.72                  |
| $I_1(2,2)$ | 7,049,871                  | 9.25               | 5.75                  |
| $I_1(3,1)$ | 7,144,819                  | 13.10              | 4.48                  |
| $I_1 Opt$  | 7,180,568                  | 8.94               | 4.00                  |
| $I_1 LP$   | 7,479,954                  | 0.70               | -                     |

Table 5.2: Experiment over parameters  $Iwin$  and  $Cwin$  for instance  $I_1$ .

One thing worth noting is that increasing the size of the integer window ( $Iwin$ ) does not always lead to a better quality solution. For example, computational experiments  $I_1(1,2)$  and  $I_1(1,3)$  in Table 5.2 show a better result in time and value than  $I_1(2,2)$ . One would expect that by increasing the value of  $Iwin$  the SWH should increase the accuracy in the prediction of the value of the  $SPPM2_{BLP}$  (as in experiment  $I_1(3,1)$ ), but that is not always the case.

### 5.5.2 Larger computational instances

Two mining instances ( $I_2$  and  $I_3$ ) have been considered to measure the performance of the SWH when larger datasets are used. The instance  $I_2$  consists of a block model of 675 blocks whereas the instance  $I_3$  is composed of 4,682 blocks. Figure 5.8 illustrates the mining input data for instances  $I_2$  and  $I_3$ , where colour scale represents the economic value of the blocks.

Instance  $I_2$  originated from a real dataset from a Chilean mine. To reduce the number of variables, the original block model was aggregated from  $5 \times 5 \times 5$  metres blocks to macro blocks of  $60 \times 60 \times 30$ . After the aggregation, blocks outside the UPL are discarded from the block model, ending with 675 macro-blocks. The total tonnage within the UPL

is 171,249,775 tons. The optimisation parameters chosen to test this model are: mining width template of 2 by 2 blocks, a discount factor of 20% and a maximum of seven pushbacks. The minimum tonnage for each pushback is set at 10,000,000 tonnes, while the maximum tonnage is constrained to 30,000,000 tonnes.

Instance  $I_3$  is a modification of the KD block model publicly available in the *Minelib* library [34]. The size of each of the blocks is 20x20x15 metres, and the total tonnage within the modified UPL is 74,631,720 tonnes. The optimisation parameters chosen to test the instance  $I_3$  are: mining width template of 3 by 2 blocks, a discount factor of 20% and a maximum number of pushbacks of five. The minimum tonnage for the pushbacks is set to 10,000,000 tonnes, while the maximum tonnage is constrained to 15,000,000 tonnes.

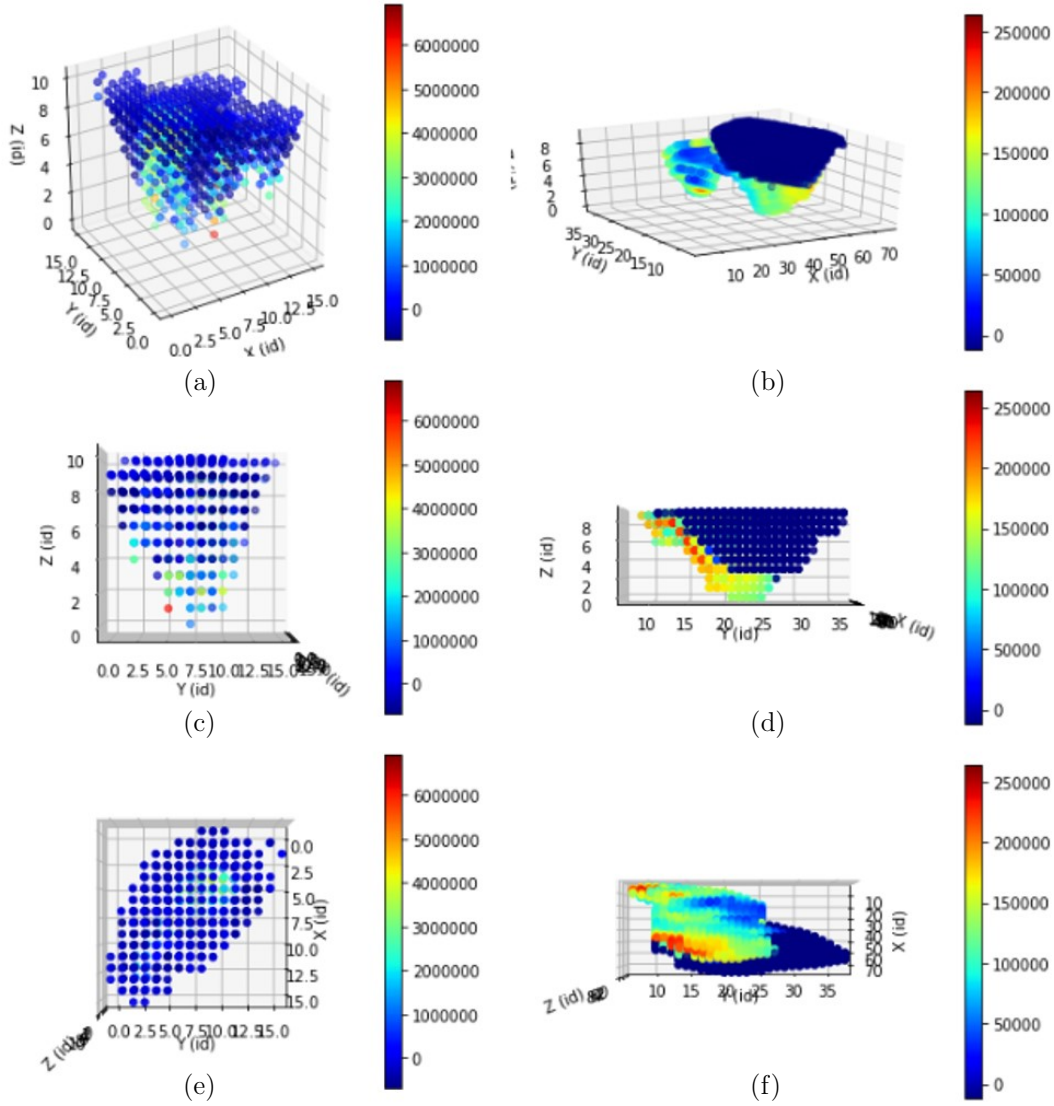


Figure 5.8: Mining input data for instances  $I_2$  and  $I_3$ , where colour scale represents the economic value of the blocks. Plots (a), (c) and (e) are isometric, YZ and XY views of the input data for instance  $I_2$ , respectively. Similarly, plots (b), (d) and (f) are isometric, YZ and XY views of the input data for instance  $I_3$ , respectively.

The combination of parameters  $I_{win}$  and  $C_{win}$  for instances  $I_2$  and  $I_3$  summarised in Table 5.3 are the ones that result in a feasible solution for the  $SPPM2_{BLP}$  using the SWH in less than two hours (7,200 [s]). Also, a maximum running time of 24 hours (86,400 [s]) was imposed for  $I_2 Opt$  and  $I_3 Opt$ . Table 5.3 summarises the result of the computational experiments for instances  $I_2$  and  $I_3$ . Note that it was not possible to find a solution for the instances  $I_2 Opt$  and  $I_3 Opt$  in less than 24 hours, so the values reported in Table 5.3 correspond to the best solution found by the solver within the time limit.

| Experiment  | Obj. Function Value [US\$] | Running Time [sec] | linear relax. gap [%] |
|-------------|----------------------------|--------------------|-----------------------|
| $I_2(1, 3)$ | 192,111,800                | 139                | 20.8                  |
| $I_2(2, 3)$ | 215,880,863                | 1092               | 11.0                  |
| $I_2(2, 4)$ | 203,215,852                | 2,828              | 16.2                  |
| $I_2 Opt^*$ | 215,597,421                | 86,400             | 11.1                  |
| $I_2 LP$    | 242,476,937                | 74                 | -                     |
| $I_3(1, 1)$ | 215,452,113                | 2,414              | 6.6                   |
| $I_3(1, 2)$ | 218,204,764                | 4,971              | 5.4                   |
| $I_3 Opt^*$ | 214,356,412                | 86,400             | 7.1                   |
| $I_3 LP$    | 230,735,261                | 11,900             | -                     |

Table 5.3: Computational experiments for problem instances  $I_2$  and  $I_3$ . The values presented in  $I_2 Opt^*$  and  $I_3 Opt^*$  are the best solutions found by the solver within the time limit of 24 hours.

One interesting result is that experiments  $I_2(2, 3)$ ,  $I_3(1, 1)$  and  $I_3(1, 2)$  outperformed their respective  $Opt^*$  (restricted to running for 24 hours). Also, it can be seen that the size of the window for the SWH does not always lead to a better quality solution. For example computational experiment  $I_2(2, 4)$  in Table 5.3 shows a worse result in time and value than  $I_2(2, 3)$ . The increment of the size of the window in the SWH should increase the accuracy in the prediction of the value of the  $SPPM2_{BLP}$ , but that is not always the case.

In the majority of cases tested, the infeasibility of a solution is due to the violation of the minimum tonnage constraint. This phenomenon tends to happen more frequently when the size of the window ( $Iwin + Cwin$ ) is small compared to the number of pushbacks ( $P$ ). In practice the combination of values  $Iwin$  and  $Cwin$  that gave better results were the ones such that  $P/2 \leq Iwin + Cwin \leq P$ . Also, we observed that increasing the value of  $Iwin$  helped to find feasible solutions for complex problem instances. However, for large instances (block models larger than 15,000 blocks) values of  $Iwin > 1$  were computationally too expensive.

The largest instance tested consisted of 11,076 blocks, 5 pushbacks, a mining width templates of  $3 \times 2$ , a sliding window of  $Iwin = 1$  and  $Cwin = 1$ , two directions  $\vec{d}$ :  $\vec{X}$  and  $\vec{Y}$ . The instance took 6,269 [sec] to solve. However, we noticed that the solution time increased significantly when more than 20,000 blocks were considered.

Additionally, a relaxation of the mining width and connectivity constraints was computed for the mining instances  $I_2$  and  $I_3$  (named  $I_2RC$  and  $I_3RC$ , respectively), similar to the resource-constrained pushback models introduced in [90] and [66]. The running times to find the global optimum solution for  $I_2RC$  and  $I_3RC$  (without using any heuristic method) were 1 and 492 seconds, respectively. However, despite the significant reduction in running time, the solutions obtained for  $I_2RC$  and  $I_3RC$  were highly scattered and disconnected, requiring a significant intervention to impose mining width and connectivity conditions afterwards. Figures 5.9 and 5.10 show a visual comparison between instances  $I_2$  and  $I_3$  without and with mining width and connectivity constraints, respectively. Also,

we observed that relaxing the mining width and connectivity conditions may result in a significant overestimation of the approximated NPV. For example, the value obtained in  $I_2RC$  and  $I_3RC$ , were 461,364,742 and 232,633,595, respectively; which in the case of  $I_2RC$  results in being 90% higher than the upper bound (see  $I_2LP$  from Table 5.3).

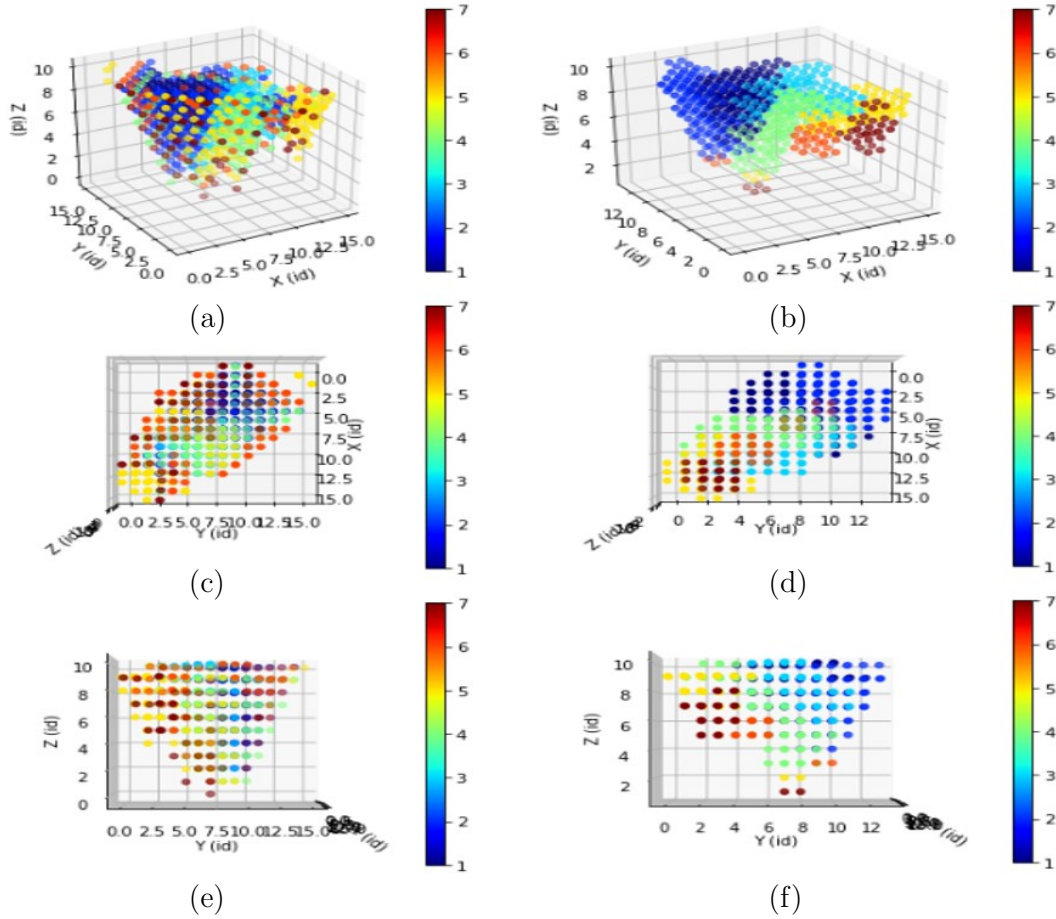


Figure 5.9: Visualisation of the  $I_2(2,3)$  plots ((b),(d),(f)) compared to the instance  $I_2$  without considering the mining width and connectivity constraints (plots (a),(c),(e)). Plots (a), (c) and (e) are isometric, YZ and XY views of the relaxation of the practical constraints for instance  $I_2$ , respectively. Similarly, plots (b), (d) and (f) are isometric, YZ and XY views of the output of instance  $I_2(2,3)$ , respectively.

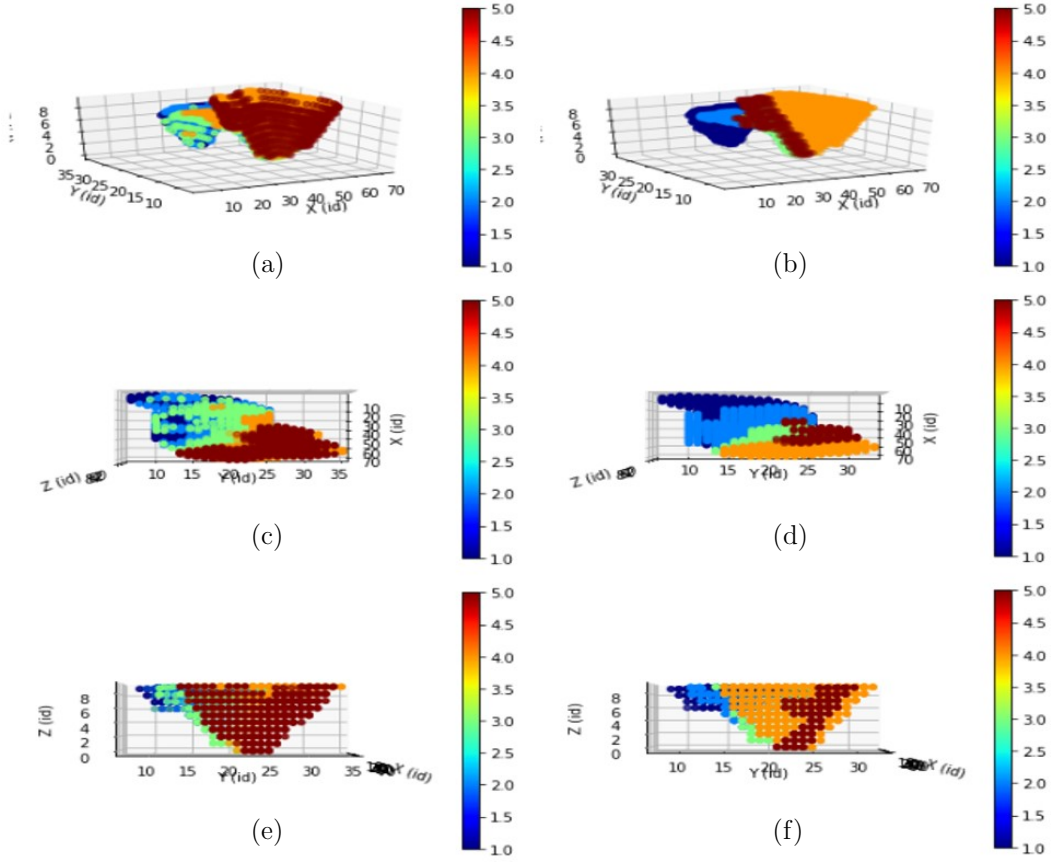


Figure 5.10: Visualisation of the  $I_3(1,2)$  (plots (b),(d),(f)) compared to the instance  $I_3$  with a relaxation of the mining width and connectivity constraints (plots (a),(c),(e)). Plots (a), (c) and (e) are isometric, YZ and XY views of the relaxation of the practical constraints for instance  $I_3$ , respectively. Similarly, plots (b), (d) and (f) are isometric, YZ and XY views of the output of instance  $I_3(1,2)$ , respectively.

## 5.6 Conclusion

This chapter has presented two mathematical models for the semi-practical pushback problem. Both formulations address connectivity and ensure a minimum operational width for the pushbacks. The first formulation ( $SPPM1_{BLP}$ ) uses a reference point to model the connectivity, ensuring that all blocks selected in a pushback are connected to that point. The second formulation ( $SPPM2_{BLP}$ ) uses a set of horizontal unidirectional graphs  $\mathcal{G}_l$  to enforce the connectivity. A small instance was considered to solve the two formulations to optimality, showing that the model  $SPPM2_{BLP}$  required only half as many variables and constraints and took a quarter of the time to solve.

An early and late start variable reduction preprocess [39] and an adaptation of the

Sliding Time Window Heuristic [24] named Sliding Window Heuristic (SWH) were considered to solve larger size instances of the  $SPPM2_{BLP}$ . A sensitivity analysis of the SWH parameters  $IWin$  (integer window) and  $CWin$  (continuous window) was conducted considering three mining instances ( $I_1$ ,  $I_2$  and  $I_3$ ). As a result, we found that in some cases, smaller windows ( $window = IWin + CWin$ ) led to better solutions than larger size windows, for the same problem instance. Also, the use of small windows ( $window < P/2$ ) ended in non-feasible solutions for the majority of the cases studied. Additionally, we noted that considering large values for parameter  $IWin$  helped to find feasible solutions, but we also noted that values of  $IWin > 1$  were computationally too expensive for instances with more than 15,000 blocks.

In addition, a couple of experiments that relaxed pushback workability conditions (minimum mining width and connectivity) while keeping the material content conditions (similar to pushback models introduced in [90] and [66], see Chapter 2) showed that highly scattered and disconnected pushbacks (and so impractical) were obtained. Also, significant overestimation in value may be obtained through the use of those relaxed models. This result greatly contrasts with the pushbacks obtained with our new method, highlighting the importance of adding workability constraints in order to produce more accurate and realistic designs.

The use of the SWH allowed us to find good quality solutions in considerably less time than the direct implementation and solution of the  $SPPM2_{BLP}$  formulation into MILP solvers. In the case of the problem instance  $I_2$ , the best solution found had a gap of 11% and took 18.2 minutes to compute, while the direct implementation of  $SPPM2_{BLP}$  found a solution with 11.1% gap in 24 hours. For the problem instance  $I_3$ , the solution method allowed us to find a solution with a gap of 6.6 % in 40.2 minutes ( $I_3(1,1)$ ) and a solution with a gap of 5.4% in 82.8 minutes ( $I_3(1,2)$ ). Whereas the solution from the direct implementation of the  $SPPM2_{BLP}$  into the solver (without heuristics) found a solution with 7.1% gap in 24 hours.

Some limitations are worth mentioning. First, the SWH proposed here is limited by the size and computational complexity of the sub-problems solved in the iterations of the heuristic. We noted that the SWH is capable of finding solutions in less than 24 hours when problem instances are limited to a few tens of thousands of blocks. For larger size instances other heuristic methods might be worth exploring. Second, although the  $SPPM2_{BLP}$  formulation required fewer variables and constraints than the  $SPPM1_{BLP}$  for the instances studied, the connectivity is only guaranteed along lines in the directions considered, which in some problem instances may lead to disconnected regions where not enough directions are included. Also, the shapes obtained are constrained by 1) the direction considered for the graphs  $\mathcal{G}_l$  as shown in Figure 5.6 (convex shapes when more than two non-parallel directions are considered), and 2) the shape of the rectangular templates considered. Third, the models introduced do not account for a detailed production schedule, so pushbacks obtained are optimal for an approximated NPV based on the estimated values for parameter  $\delta_p$ .

## 5.7 Summary

Open-pit mines are surface excavations created to extract valuable material which, in most of the cases, is located near the surface. Pushbacks are connected regions of a mine with enough area to support the mining operation over a defined period. Pushbacks designed without taking into account haulage ramps are defined here as semi-practical pushbacks. Two new optimisation models for semi-practical pushbacks are introduced, that account for workability conditions: minimum operational area and connectivity within the blocks that compose the pushbacks. Additionally, the adaptation of the Sliding Time Window Heuristic [24] and other preprocessing routines allowed us to significantly reduce the solution time and obtain close to optimal solutions for the instances studied.



## Chapter 6

# A mathematical formulation for the open-pit long-range mine planning problem

*Determining the ultimate pit limit is a complex process bound by many constraints covering both economic and physical aspects of the full life of the mine . . . physical aspects -such as access road, safety batters, etc.- are not considered in current methods and must be added later by hand. . . meaning that the final solution will now lose any claims to optimality in real terms.*

—Tim Gill, Mike Robey and Terry Caelli [43].

ONE of the key problems in open-pit long-range mine planning consists of finding the set of pushbacks, their production schedule and the allocation of the material to an appropriate destination that maximises the discounted cash-flows (NPV) and satisfies the operational conditions of the pushback design, as discussed in Chapter 1 this problem is called the open-pit long-range mine planning problem (OPLRMPP).

The OPLRMPP implies that two different problems -the pushback design and the production scheduling problems- must be addressed simultaneously. The pushback design problem aims to find a mining sequence that maximises the economic value given by the sequential (one after the other) extraction of the pushbacks. The key constraints imposed on the practical pushbacks are: minimum mining width, block connectivity within the same pushback, ramp access at each level of any pushback (this condition is relaxed in the case of semi-practical pushbacks), geotechnical constraint (also known as the precedence constraint), and resources content constraints or tonnage constraints.

On the other hand, the production scheduling problem -as discussed in Chapter 2- consists of finding the period of time when the blocks should be mined and processed in order to maximise the NPV, subject to satisfying the mining and processing capacity constraints, as well as the geotechnical constraints. Some versions of the production scheduling problem also aim to find the best destination for the blocks according to the trade-off between the profit generated at each destination and the available processing capacity at each period.

Therefore, the OPLRMPP must consider the practical constraints related to the pushback design, as well as account for the scheduling of the blocks within those pushbacks in a way that the processing and mining capacities are honoured at any period. In addition, the OPLRMPP must also find the best destination for the mined blocks at each period. Since the OPLRMPP considers the pushback and the production scheduling problems simultaneously, the necessity of imposing tonnage constraints on the pushbacks is no longer required. These are taken into account in the processing constraints.

The approaches taken to produce practical and semi-practical pushback designs in Chapter 4 and Chapter 5 indirectly account for the extraction time of the blocks by using the tonnage constraint to pre-compute an estimate of when each pushback takes place and then applying a suitable weighting factor. Also, the destination of the blocks is pre-defined before the optimisation, by simply choosing the highest profit destination for each block.

To sum up, the OPLRMPP can be defined as the problem of finding the sequence and the schedule of the blocks that maximises the NPV of the extraction. The mining sequence is given by a set of practical pushbacks that must satisfy the following three conditions: 1) there should be a minimum mining width at every bench, 2) every bench of each pushback must be accessible via a connected haulage ramp, and 3) each pushback must be a connected set, in terms of horizontal and vertical adjacencies between blocks. The mining scheduling involves defining the period when the blocks within the pushbacks are mined and, at the same time, determining how much of the mined material is sent to the processing plant and how much is treated as waste. The mine schedule must satisfy the minimum and maximum mining and processing capacity constraints at any period. Additionally, the blocks that belong to the set of the precedences of any block  $b$  are those which need to be removed before the extraction of block  $b$  in order to keep the maximum wall slope constraint. Also, because the cost of the infrastructure related to a pushback can be considered expensive and there is limited mobility of some large mining equipment (like electric shovels), the number of pushbacks that are active in a mining operation at any point in time is constrained by the maximum number of pushbacks that can be mined simultaneously.

This chapter is organised as follows. In Section 6.1, a mathematical formulation for this problem is presented. In Section 6.2, we introduce the *sample and refine* algorithm to solve medium size problem instances. Finally, in Section 6.3, we present a case study for the open pit planning and design formulation.

## 6.1 The mathematical model

In the following we propose a mathematical formulation of the OPLRMPP (*OPLRMPP – Mod*). Our model produces a set of pushbacks which maximises the time discounted cash-flows (NPV). The cut-off grade (classifying each block as either waste or ore) is determined dynamically in the model, so there is no need for a pre-determined cut-off grade like the approaches taken in Chapters 4 and Chapter 5. In our model, each pushback must

have a ramp that connects the bottom of the pushback to the surface. For simplicity, this model assumes that all material either gets sent directly to the mill or treated as waste (as previously stated in Chapter 1).

The ramp is modelled as a set of adjacent blocks that creates a continuous path from the bottom to the top of the pushback. To achieve vertical distance between adjacent levels and the gradient constraint, the ramp must have a minimum number of blocks at each level. The ramp is modelled using the concepts of *ramp segments*, *ramp entrance* and *ramp segment template* introduced in Chapter 4.

The mining width is modelled using the rectangular templates introduced in Chapter 5, and the connectivity constraint is modelled by a combination of the approaches introduced in Chapter 4 and Chapter 5. The idea is to use the connectivity path idea introduced in the model  $SPPM1_{BLP}$  of Chapter 5, but using the *ramp entrance* (as in Chapter 4) as the reference point.

The geotechnical slope wall constraint is modelled using a set of vertical precedences, as explained in Chapter 5. The mine scheduling and the definition of the best processing destination for the mined material is modelled using a combination of integer and fractional variables, following the approach introduced in [14], [31] and [70]. A time window (initial and final period) for the extraction of the pushbacks is imposed to model the pushback extraction and to keep the problem linear. The maximum number of active pushbacks per period can be imposed through a proper definition of the time window for the pushbacks.

The following notation describes the sets, indices and parameters used in our formulation. Guidelines presented in [94] were followed to facilitate the readability of our formulation.

### Indices, sets and parameters

$b \in \mathcal{B}$  : blocks in the block model  $1, \dots, B$ . A LG-UPL pit boundary (as described in Chapter 2) could be used to reduce the size of the problem. However, there is no guarantee that the set of optimal pushbacks will be contained within the LG-UPL.

$t \in \mathcal{T}$  : periods  $1, \dots, T$ , where  $T$  is the maximum period considered for the schedule.  $T$  can be estimated as  $T = \lceil \frac{\sum_{b \in \mathcal{B}} W_b}{\min_{t \in \mathcal{T}} \{W_t^{ML}\}} \rceil$ , parameters  $W_b$  and  $W_t^{ML}$  are explained below.

$p \in \mathcal{P}$  : pushbacks  $1, \dots, P$ .  $P$  corresponds to the maximum number of pushbacks to be considered, as no minimum resource condition is imposed on the pushbacks, some of the pushbacks can be empty. However, in case a resultant pushback has a material content that is unsuitable for the mining operation, it might be worth adding a minimum material content constraint for that pushback.

$d \in \mathcal{D}_p$  : set of distances,  $0, \dots, D$ , from the reference point associated with pushback  $p$ . The distance is measured as the length (in blocks) of the path formed by

a series of adjacent blocks that start at a reference point  $c$  which is determined for each pushback during the optimisation.

$\hat{b} \in \hat{\mathcal{B}}_b$  : vertical precedences of block  $b$ .

$l \in \mathcal{L}$  : levels in the block model;  $1, \dots, L$ .

$\ddot{b} \in \ddot{\mathcal{B}}_l$  : blocks that belong to a level  $l$ .

$\breve{b} \in \breve{\mathcal{B}}_b$  : blocks that share the same level with block  $b$ .

$b_s^+ \in \mathcal{B}_s^+$  : set of blocks immediately above blocks in ramp segment  $s$  (as in Chapter 4).

$b_s^- \in \mathcal{B}_s^-$  : set of blocks immediately below blocks in ramp segment  $s$  (as in Chapter 4).

$s' \in \mathcal{S}'_b$  : ramp segments that start at block  $b$ . The block  $b$  acts as a *ramp entrance* of the segments  $s' \in \mathcal{S}'_b$  (as in Chapter 4).

$s'' \in \mathcal{S}''_b$  : ramp segments that end at block  $b$  (as in Chapter 4).

$s \in \mathcal{S}$  : all ramp segments ( $\mathcal{S} = \bigcup_{b \in \mathcal{B}} \mathcal{S}'_b$ ), as in Chapter 4.

$\bar{b} \in \bar{\mathcal{B}}_b$  : set of horizontal adjacent blocks of block  $b$  (this includes diagonally adjacent blocks).

$w \in \mathcal{W}$  : set of mining width rectangular templates (as in Chapter 5).

$w' \in \mathcal{W}'_b$  : set of mining width rectangular templates that contain block  $b$  (as in Chapter 5).

$b' \in \mathcal{B}'_w$  : set of blocks contained in the mining width rectangular template  $w$  (as in Chapter 5).

$V'_b$  : economic value per ton [US\$/ton] of block  $b$  sent to process (it can be negative).

$V''_b$  : economic value per ton [US\$/ton] of block  $b$  sent to waste dump. Associated with the extraction cost per tonne (drilling, blasting, loading, hauling and waste dump maintenance cost).

$W_b$  : tonnage of block  $b$  [tonnes].

$\overline{W}_t^{ML}$  : maximum mining limit [tonnes] for period  $t$ .

$\overline{W}_t^{PL}$  : maximum processing limit [tonnes] for period  $t$ .

$\underline{W}_t^{ML}$  : minimum mining limit [tonnes] for period  $t$ .

$\underline{W}_t^{PL}$  : minimum processing limit [tonnes] for period  $t$ .

$T_p^{ini}$  : minimum extracting period for pushback  $p$ .

$T_p^{fin}$  : maximum mining period for pushback  $p$ .

$\delta_t$  : discount factor for period  $t$ .  $\delta_t = (1 + \zeta)^{-t}$ , where  $\zeta$  is the discount rate.

Four types of binary variables are used in the model. The first variable type indicates to which pushback a block belongs and when that block, and subsequently, that pushback is mined. Variables  $x_{bpt}$  that are outside the bounds ( $T_p^{ini} \leq t \leq T_p^{fin}$ ) are set to zero in a preprocessing step:

$$x_{bpt} = \begin{cases} 1, & \text{if the block } b \text{ belongs to pushback } p \text{ and is mined in period } t, \text{ where } T_p^{ini} \leq t \leq T_p^{fin}. \\ 0, & \text{otherwise.} \end{cases}$$

The second variable type represents the ramp segment which determines the ramp of the pushback:

$$z_{sp} = \begin{cases} 1, & \text{if the ramp segment } s \text{ is selected for pushback } p \\ 0, & \text{otherwise} \end{cases}$$

The third type of variable accounts for the horizontal distance (in number of blocks) between the blocks and the ramp entrance on the same level.

$$v_{bdp} = \begin{cases} 1, & \text{if the block } b \text{ is exactly at distance } d \text{ from a ramp segment of pushback } p \\ 0, & \text{otherwise} \end{cases}$$

Finally, the fourth type of variable represents the rectangular templates that are used to force the minimum mining width for the pushbacks.

$$u_{wp} = \begin{cases} 1, & \text{if mining width template } w \text{ is assigned to pushback } p \\ 0, & \text{otherwise} \end{cases}$$

Finally, there are two types of continuous variables, both restricted to lying between 0 and 1

$y'_{bt}$  = fraction of block  $b$  that is sent to the processing plant at period  $t$ .

$y''_{bt}$  = fraction of block  $b$  that is sent to the waste dump at period  $t$ .

### 6.1.1 Objective function

The objective function is defined as follows:

$$\text{Max} \sum_{t \in \mathcal{T}} \delta_t \sum_{b \in \mathcal{B}} W_b (V'_b \cdot y'_{bt} + V''_b \cdot y''_{bt})$$

The objective function maximises the time-weighted total profit of the mining extraction, or the net present value (NPV) of the mine extraction.

$$\sum_{p \in \mathcal{P}} x_{bpt} = y'_{bt} + y''_{bt} \quad b \in \mathcal{B}, \quad t \in \mathcal{T} \quad (6.1)$$

$$\underline{W}_t^{ML} \leq \sum_{b \in \mathcal{B}} W_b (y'_{bt} + y''_{bt}) \leq \overline{W}_t^{ML} \quad t \in \mathcal{T} \quad (6.2)$$

$$\underline{W}_t^{PL} \leq \sum_{b \in \mathcal{B}} W_b \cdot y'_{bt} \leq \overline{W}_t^{PL} \quad t \in \mathcal{T} \quad (6.3)$$

$$\sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} x_{bpt} \leq 1 \quad b \in \mathcal{B} \quad (6.4)$$

Constraint (6.1) states that the material sent to the waste dump or processing plant is equal to the material mined. Constraint (6.2) ensures that the total material mined per period is within the minimum and maximum limits. Similarly, Constraint (6.3) ensures that the total material sent to the processing plant at each period is within the processing capacity limits. Constraint (6.4) forces each block to be assigned to at most one pushback and that it will be mined in one period only.

### 6.1.2 Geotechnical constraints

$$\sum_{\rho \leq p} \sum_{\tau \leq t} x_{b\rho\tau} \leq \sum_{\rho \leq p} \sum_{\tau \leq t} x_{\hat{b}\rho\tau} \quad \hat{b} \in \hat{\mathcal{B}}_b, \quad b \in \mathcal{B}, \quad p \in \mathcal{P}, \quad t \in \mathcal{T} \quad (6.5)$$

As in Chapter 5, Constraint (6.5) ensures the vertical precedences are honoured in the extraction of any block.

### 6.1.3 Mining width constraints

$$\sum_{t \in \mathcal{T}} x_{bpt} \leq \sum_{w' \in \mathcal{W}'_b} u_{w'p} \quad \forall b \text{ such that } |\mathcal{W}'_b| > 0, \quad p \in \mathcal{P} \quad (6.6)$$

$$u_{wp} \leq \sum_{t \in \mathcal{T}} x_{b'pt} \quad p \in \mathcal{P}, \quad w \in \mathcal{W}, \quad b' \in \mathcal{B}'_w \quad (6.7)$$

$$\sum_{p \in \mathcal{P}} u_{wp} \leq 1 \quad w \in \mathcal{W} \quad (6.8)$$

As in Chapter 5, Constraint (6.6) states that if a block is assigned to a pushback then at least one mining width template that contains that block must be assigned to that pushback. Constraint (6.7) guarantees that if a mining width template belongs to a pushback  $p$  then all blocks within that mining width template must belong to the same pushback. Constraint (6.8) ensures that each mining width template is assigned to at most one pushback.

#### 6.1.4 Connectivity constraints

$$\sum_{t \in \mathcal{T}} x_{bpt} \leq \sum_{d \in \mathcal{D}} v_{bdp} \quad b \in \mathcal{B}, \quad p \in \mathcal{P} \quad (6.9)$$

$$v_{b(d+1)p} \leq \sum_{\bar{b} \in \bar{\mathcal{B}}_b} v_{\bar{b}dp} \quad b \in \mathcal{B}, \quad d \in \mathcal{D} - \{D\}, \quad p \in \mathcal{P} \quad (6.10)$$

$$v_{b0p} \leq \sum_{s' \in \mathcal{S}'_b} z_{s'p} \quad b \in \mathcal{B}, \quad p \in \mathcal{P} \quad (6.11)$$

$$v_{bdp} \leq \sum_{t \in \mathcal{T}} x_{bpt} \quad b \in \mathcal{B}, \quad d \in \mathcal{D}, \quad p \in \mathcal{P} \quad (6.12)$$

$$\sum_{\bar{b} \in \bar{\mathcal{B}}_l} \sum_{s' \in \mathcal{S}'_{\bar{b}}} z_{s'p} \leq 1 \quad l \in \mathcal{L}, \quad p \in \mathcal{P} \quad (6.13)$$

Constraint (6.9) implies that block  $b$  of pushback  $p$  must be within distance  $D$  from a ramp segment. Constraint (6.10) means that a block  $b$  can only be classified as having distance  $d + 1$  from a reference point if one of its neighbouring blocks has distance  $d$  from the reference point. Constraint (6.11) sets as zero the distance  $d$  between block  $b$  and the ramp segment that starts at the same block. Constraint (6.12) ensures the connectivity between the blocks within a pushback from the chosen ramp segment. Constraint (6.13) means that at most one ramp segment can be chosen at each level per pushback.

#### 6.1.5 Haulage ramp constraints

$$\sum_{s'' \in \mathcal{S}''_b} z_{s''p} - \sum_{s' \in \mathcal{S}'_b} z_{s'p} = 0, \quad b \text{ such that } |\mathcal{S}'_b| + |\mathcal{S}''_b| > 0, \quad p \in \mathcal{P} \quad (6.14)$$

$$\sum_{t \in \mathcal{T}} x_{\bar{b}pt} \leq \sum_{\bar{b} \in \bar{\mathcal{B}}_l} \sum_{s' \in \mathcal{S}'_{\bar{b}}} z_{s'p} \quad \bar{b} \in \bar{\mathcal{B}}_l, \quad l \in \mathcal{L}, \quad p \in \mathcal{P} \quad (6.15)$$

$$z_{sp} + \sum_{t \in \mathcal{T}} \sum_{\rho \leq p} x_{b_s^- \rho t} \leq 1 \quad s \in \mathcal{S}, \quad b_s^- \in \mathcal{B}_s^-, \quad p \in \mathcal{P} \quad (6.16)$$

$$z_{sp} - \sum_{t \in \mathcal{T}} x_{b_s^+ pt} \leq 0 \quad s \in \mathcal{S}, \quad b_s^+ \in \mathcal{B}_s^+, \quad p \in \mathcal{P} \quad (6.17)$$

$$z_{sp} - \sum_{t \in \mathcal{T}} x_{b_s pt} \leq 0 \quad s \in \mathcal{S}, \quad b_s \in \mathcal{B}_s, \quad p \in \mathcal{P} \quad (6.18)$$

Constraint (6.14) ensures the continuity of the ramp segments. Constraint (6.15) ensures that there exists a ramp segment for pushback  $p$  at each level of any block of pushback  $p$ . Constraint (6.16) forces the blocks directly below a ramp segment to be either not mined or to belong to a pushback with a larger index  $p$ . Finally, Constraints (6.17) and (6.18) ensure that blocks within and immediately above a ramp segment must belong to the same pushback as the ramp segment in order to ensure the ramp is located at the boundary of the pushback.

## 6.2 Solution method

Although the *OPLRMPP – Mod* introduced in Section 6.1 can be implemented directly and solved using commercial MILP-solvers, it can take several hours to solve small to medium size instances to optimality. For that reason, a set of preprocessing routines (early and late start), a *sliding time window heuristic*, a relaxation of the haulage ramp constraint, and a *sampling and refining* heuristic are proposed here. The haulage ramp is replaced by a single point (as in Chapter 5) to reduce the number of variables in the *OPLRMPP – Mod*. Then, the *sampling and refining* algorithm is designed to help find such reference points  $c \in \mathcal{C}$  ( $\mathcal{C}$ : set of xy-coordinates of all blocks in  $\mathcal{B}$ ) for each pushback.

As in Chapter 5 Section 5.4.1, the early and late start algorithms work by combining the upper and lower bounds of the tonnage constraints at each period with the precedence constraints. As was mentioned, a detailed explanation of the early and late start algorithms can be found in [39] pages 43-52. The difference between the sliding window heuristic presented in this section and the one in Chapter 5, is that the current approach divides the set  $\mathcal{T}$  instead of  $\mathcal{P}$ , since most of the key border conditions of the *OPLRMPP – Mod* are time-dependent (like processing and mining capacity constraints).

### 6.2.1 Sliding time window heuristic

The sliding time window heuristic (STWH) described here is an adaptation of the heuristic described in [24]. The main idea of the STWH is to repeatedly solve a partly relaxed sub-problem of the *OPLRMPP-Mod*. At each iteration of the algorithm a sub-problem (*OPLRMPP – Mod<sub>sub</sub>*) is set-up and solved, then a fraction of the variables of that sub-problem is fixed, and this is repeated until all the variables of the mixed integer linear programming model have been fixed. The first step of the STWH is to define three sequential subsets of  $\mathcal{T}$ :  $\mathcal{T}_1$ ,  $\mathcal{T}_2$ ,  $\mathcal{T}_3$ , such that  $|\mathcal{T}_1| + |\mathcal{T}_2| + |\mathcal{T}_3| \leq |\mathcal{T}|$ . The variables in the subset  $\mathcal{T}_1$  correspond to the variables that will be fixed with the solution values obtained at a previous iteration of the algorithm (variables marked with the star symbol, \*). The variables in the subset  $\mathcal{T}_2$  correspond to the integer window (*IWin*), which means that the variables in  $\mathcal{T}_2$  remain as integer. Finally, the integrality constraint for the variables in the subset  $\mathcal{T}_3$  is relaxed (continuous window or *CWin*).

### 6.2.2 Sampling and refining algorithm

The key idea of the *sampling and refining* algorithm is to iteratively solve a series of simplified problems that have a reduced number of coordinates  $\hat{\mathcal{C}} \subseteq \mathcal{C}$ . At each iteration of the algorithm the elements in the set  $\hat{\mathcal{C}}$  change according to the solution obtained in the previous iteration. Initially, the elements in  $\hat{\mathcal{C}}$  are a sample of the set  $\mathcal{C}$ , while in the following iterations the elements in  $\hat{\mathcal{C}}$  correspond to the neighbours of the coordinates obtained in the solution of the previous iteration. The algorithm stops when there has been no improvement in NPV after a certain number of iterations. Figure 6.1 illustrates an example of how the *sampling and refining* algorithm works for a simple case.

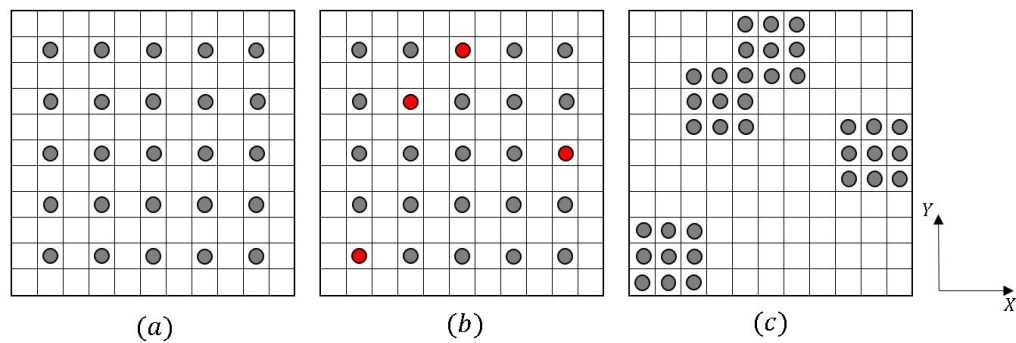


Figure 6.1: Example of the *sampling and refining* algorithm. (a) blocks marked in grey represent the reduced number of coordinates  $\hat{\mathcal{C}}$  at the first iteration of the algorithm. (b) the blocks in red represent the coordinates selected as reference points for the pushbacks. (c) the blocks in grey represent the new set of reduced coordinates which correspond to the neighbours of the coordinates selected as a solution in the previous iteration.

**Algorithm 2:** Sampling and refining**Data:**  $Iwin, Cwin, \mathcal{C}, maxSteps, dx, dy$ .**Result:** A solution for the OPLRMPP – Mod.

```

1 begin
2    $steps \leftarrow 0; solVal \leftarrow 0; valOpt \leftarrow 0; solZ \leftarrow \emptyset; solSet \leftarrow \emptyset; solOpt \leftarrow \emptyset;$ 
3    $\hat{\mathcal{C}} \leftarrow$  a sample of elements in set  $\mathcal{C}$ 
4    $(solZ, solSet, solVal) \leftarrow \text{STWH}(Iwin, Cwin, \hat{\mathcal{C}})$ 
5    $valOpt \leftarrow solVal; solOpt \leftarrow solSet;$ 
6   while  $steps \leq maxSteps$  do
7      $\hat{\mathcal{C}} \leftarrow$  set of neighbours within distances  $dx$  and  $dy$  from coordinates in  $solZ$ 
8      $(solZ, solSet, solVal) \leftarrow \text{STWH}(Iwin, Cwin, \hat{\mathcal{C}})$ 
9     if  $solVal > valOpt$  then
10       $valOpt \leftarrow solVal; steps \leftarrow 0; solOpt \leftarrow solSet$ 
11    else
12       $steps \leftarrow steps + 1$ 
13  return  $solOpt$ 
14  Function  $\text{STWH}(Iwin, Cwin, \mathcal{C})$ :
15     $\mathcal{T}_1 \leftarrow \emptyset; T_1 \leftarrow 0; solSet \leftarrow \emptyset; solVal \leftarrow 0; solZ \leftarrow \emptyset$ 
16    foreach  $i \in \{1, \dots, T\}$  do
17       $x_{bpt}^* \leftarrow \emptyset; y_{bt}^* \leftarrow \emptyset; y_{bt}''^* \leftarrow \emptyset$ 
18       $T_2 \leftarrow \min(T, i + Iwin - 1)$ 
19       $T_3 \leftarrow \min(T, i + Iwin + Cwin)$ 
20       $\mathcal{T}_2 \leftarrow \{T_1 + 1, \dots, T_2\}$ 
21       $\mathcal{T}_3 \leftarrow \{T_2 + 1, \dots, T_3\}$ 
22      formulate OPLRMPP – Modsub( $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{C}, x_{bpt}^*, y_{bt}^*, y_{bt}''^*$ )
23      solve OPLRMPP – Modsub
24       $x_{bpt}^*, y_{bt}^*, y_{bt}''^* \leftarrow$  get solution from OPLRMPP – Modsub such that  $t \in \mathcal{T}_1$ 
25       $\mathcal{T}_1 \leftarrow \mathcal{T}_1 \cup \{i\}$ 
26       $T_1 \leftarrow T_1 + 1$ 
27     $solZ \leftarrow z_{cp}^*$ 
28     $solSet \leftarrow x_{bpt}^*, y_{bt}^*, y_{bt}''^*$ 
29     $solVal \leftarrow$  value of the objective function of OPLRMPPMsub
30    return ( $solZ, solSet, solVal$ )

```

The following is the description of Algorithm 2: *sampling and refining*. Line 2 initialises the problem variables. Line 3 defines a subset of coordinates from a regularly spaced sample of  $\mathcal{C}$  which is stored in  $\hat{\mathcal{C}}$ . Lines 4–5 call the STWH function using parameters  $Iwin$ ,  $Cwin$  and  $\hat{\mathcal{C}}$ , and store the solution. Then, the *refining* process starts with the while loop at Line 6. The loop runs until the no-improvement counter ( $steps$ ) reaches the limit ( $maxSteps$ ). In Line 7 the neighbours of the coordinates in  $solZ$  with respect to the distances  $dx$  and  $dy$  are stored as the new set of coordinates  $\hat{\mathcal{C}}$ . Lines 8–12 compute the STWH for the set of coordinates  $\hat{\mathcal{C}}$ . If the new solution is better than the current one, it is then stored as the new optimum, otherwise the no-improvement

counter is increased by 1 unit. Finally, Line 13 returns the best solution found  $solOpt$ .

Lines 14–30 of Algorithm 2 correspond to the STWH function, which works as follows. Line 15 initialises the instance variables. Then, the loop from iteration 1 to  $T$  begins at Line 16. In Line 17 the solution of the variables  $x$ ,  $y'$  and  $y''$  is initialised as empty. In Lines 18–21, the sets  $\mathcal{T}_2$  and  $\mathcal{T}_3$  are defined with respect to the iteration  $i$ . For the sets in Lines 20–21, if the first term is larger than the last then we have an empty set. Line 22 formulates the  $OPLRMPP - Mod_{sub}$  (described by the Equations 6.19 to 6.42 below) for the sets  $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, x_{bpt}^*, y_{bt}^*, y_{bt}^{''*}$ . Then, Line 23 solves the  $OPLRMPP - Mod_{sub}$  and gets the value of the variables  $x_{bpt}$ ,  $y_{bt}'$  and  $y_{bt}''$  for  $t \in \mathcal{T}_1$  saving them in the sets  $x_{bpt}^*, y_{bt}^*, y_{bt}^{''*}$ . Lines 25–26 update the set  $\mathcal{T}_1$  and the value of  $T_1$ . Finally, Lines 27–30 store and return the solution of the coordinates  $z_{cp}^*$  in  $solZ$ , the solution of the variables  $x_{bpt}^*, y_{bt}^*, y_{bt}^{''*}$  in  $solSet$  and the value of the solution in the objective function in  $solVal$ .

#### Formulation for the sub-problem $OPLRMPP - Mod_{sub}(\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{C}, x_{bp}^*, z_{cp}^*, u_{bdp}^*)$

To simplify the formulation introduced in Section 6.1, the haulage ramp condition has been relaxed and replaced with a single point. This is the same approach taken in Section 5.2.1: the variable  $z$  represents a moving reference point to control the shape of the pushback. Similarly, the set of coordinates  $\mathcal{C}$  and a mapping function  $\mathcal{F}$  are also introduced.

$c \in \mathcal{C}$  : set of coordinates that act as a reference point to ensure the connectivity of the blocks within a pushback.

$\mathcal{F}$  : a mapping function  $\mathcal{F} : \mathcal{B} \rightarrow \mathcal{C}$  such that  $\mathcal{F}(b) = c$  is the 2D projection over the XY plane of the spatial location of block  $b$ .

$$z_{cp} = \begin{cases} 1, & \text{if the coordinate } c \text{ is selected as a reference point for pushback } p \\ 0, & \text{otherwise} \end{cases}$$

$$\text{Max} \sum_{t \in \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3} \delta_t \sum_{b \in \mathcal{B}} W_b (V_b' \cdot y_{bt}' + V_b'' \cdot y_{bt}'') \quad (6.19)$$

$$\sum_{p \in \mathcal{P}} x_{bpt} = y_{bt}' + y_{bt}'' \quad b \in \mathcal{B}, \quad t \in \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3 \quad (6.20)$$

$$\underline{W}_t^{ML} \leq \sum_{b \in \mathcal{B}} W_b (y_{bt}' + y_{bt}'') \leq \overline{W}_t^{ML} \quad t \in \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3 \quad (6.21)$$

$$\underline{W}_t^{PL} \leq \sum_{b \in \mathcal{B}} W_b \cdot y_{bt}' \leq \overline{W}_t^{PL} \quad t \in \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3 \quad (6.22)$$

$$\sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3} x_{bpt} \leq 1 \quad b \in \mathcal{B} \quad (6.23)$$

$$\sum_{\rho \leq p} \sum_{\tau \leq t} x_{b\rho\tau} \leq \sum_{\rho \leq p} \sum_{\tau \leq t} x_{\hat{b}\rho\tau} \quad \hat{b} \in \hat{\mathcal{B}}_b, \quad b \in \mathcal{B}, \quad p \in \mathcal{P}, \quad t \in \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3 \quad (6.24)$$

$$\sum_{t \in \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3} x_{bpt} \leq \sum_{w' \in \mathcal{W}'_b} u_{w'p} \quad \forall b \text{ such that } |\mathcal{W}'_b| > 0, \quad p \in \mathcal{P} \quad (6.25)$$

$$u_{wp} \leq \sum_{t \in \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3} x_{b'pt} \quad p \in \mathcal{P}, \quad w \in \mathcal{W}, \quad b' \in \mathcal{B}'_w \quad (6.26)$$

$$\sum_{p \in \mathcal{P}} u_{wp} \leq 1 \quad w \in \mathcal{W} \quad (6.27)$$

$$\sum_{t \in \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3} x_{bpt} \leq \sum_{d \in \mathcal{D}} v_{bdp} \quad b \in \mathcal{B}, \quad p \in \mathcal{P} \quad (6.28)$$

$$v_{b(d+1)p} \leq \sum_{\bar{b} \in \bar{\mathcal{B}}_b} v_{\bar{b}dp} \quad b \in \mathcal{B}, \quad d \in \mathcal{D} - \{D\}, \quad p \in \mathcal{P} \quad (6.29)$$

$$v_{b0p} \leq z_{cp} \quad b \in \mathcal{B}, \quad c = \mathcal{F}(b), \quad p \in \mathcal{P} \quad (6.30)$$

$$v_{bdp} \leq \sum_{t \in \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3} x_{bpt} \quad b \in \mathcal{B}, \quad d \in \mathcal{D}, \quad p \in \mathcal{P} \quad (6.31)$$

$$\sum_{p \in \mathcal{P}} z_{cp} \leq 1 \quad c \in \mathcal{C} \quad (6.32)$$

$$x_{bpt} \equiv x_{bpt}^* \quad t \in \mathcal{T}_1, \quad p \in \mathcal{P}, \quad b \in \mathcal{B}, \quad (6.33)$$

$$y'_{bt} \equiv y'^{*}_{bt} \quad t \in \mathcal{T}_1, \quad b \in \mathcal{B} \quad (6.34)$$

$$y''_{bt} \equiv y''^*_{bt} \quad t \in \mathcal{T}_1, \quad b \in \mathcal{B} \quad (6.35)$$

$$x_{bpt} \in \{0, 1\} \quad t \in \mathcal{T}_2, \quad b \in \mathcal{B} \quad (6.36)$$

$$z_{cp} \in \{0, 1\} \quad p \in \mathcal{P}, \quad c \in \mathcal{C} \quad (6.37)$$

$$u_{wp} \in \{0, 1\} \quad p \in \mathcal{P}, \quad w \in \mathcal{W} \quad (6.38)$$

$$v_{bdp} \in \{0, 1\} \quad b \in \mathcal{B}, \quad d \in \mathcal{D}, \quad p \in \mathcal{P} \quad (6.39)$$

$$0 \leq x_{bpt} \leq 1 \quad t \in \mathcal{T}_3, \quad p \in \mathcal{P}, \quad b \in \mathcal{B}, \quad (6.40)$$

$$0 \leq y'_{bt} \leq 1 \quad t \in \mathcal{T}_2 \cup \mathcal{T}_3, \quad b \in \mathcal{B} \quad (6.41)$$

$$0 \leq y''_{wp} \leq 1 \quad t \in \mathcal{T}_2 \cup \mathcal{T}_3, \quad b \in \mathcal{B} \quad (6.42)$$

### 6.3 Computational experiments

Algorithm 2 was implemented in Python 2.7 and the instances of the mixed integer linear programming model *OPLRMPP – Mod<sub>sub</sub>* were solved using Gurobi 8.0 MILP-solver for a real mineral deposit. All the pre-processing routines, such as the definition of the sets and variables were done in Python. The execution of the code was done on an Intel Xeon machine with 32 gigabytes running Windows 10. This machine has eight processors that run at 2.40 gigahertz.

The mining instance used in this section corresponds to the same copper deposit considered in Chapter 4. However, the economic parameters have been modified to take into account the two different profits  $V'_b$  and  $V''_b$ . Nonetheless, the same aggregation process that is described in Chapter 4 is used here. At the end of the aggregation process, the problem instance was reduced to 3,445 blocks containing a total of 247,623,221 tonnes. Figure 6.2 shows different views of the resultant block model.

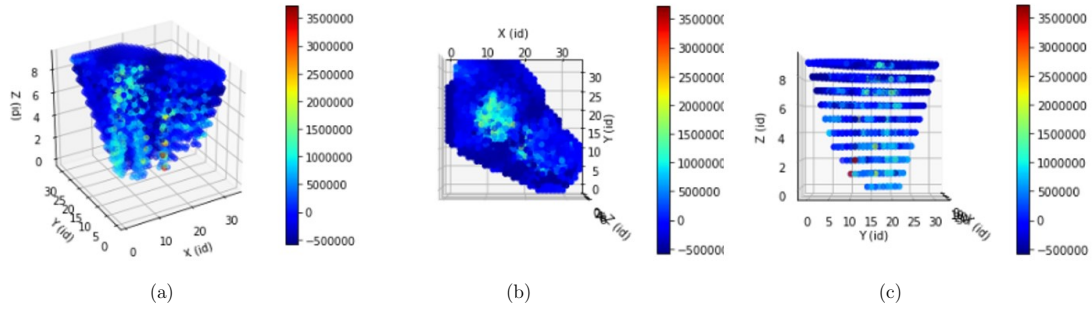


Figure 6.2: Blocks within the larger ultimate pit. The colour scale represents the value of the blocks in dollars. Plots (a), (b) and (c) represent isometric, XY and YZ views, respectively.

### 6.3.1 Optimisation parameters

The parameters chosen for the problem instance were as follows: there were six periods, each one representing a year; the maximum number of pushbacks was four, each with a maximum permitted distance  $D = 13$ ; a discount rate of  $\zeta = 12.5\%$  was chosen. The selected earliest and latest extraction periods for the pushbacks were (1,3), (2,4), (3,5) and (4,6) for pushbacks 1 to 4, respectively. The mining width template considered for this instance was 2 by 2 blocks in X and Y directions. Finally, the minimum processing plant capacity as well as the minimum mining capacity were set to zero.

The parameters  $Iwin$  and  $Cwin$  were set as 1 ( $Iwin = 1$  and  $Cwin = 1$ ) for the sliding time window heuristic (STWH) since this combination allowed finding a solution in less than a day. The parameters for the solver remained unchanged, except for the gap parameter which was set to 10%. The maximum number of steps for the no-improvement counter was set at two.

The initial set  $\hat{\mathcal{C}}$  consisted of a sample of the elements in  $\mathcal{C}$  where each pair was separated by a distance of 5 units in the X and Y directions. At the refining stages, the set  $\hat{\mathcal{C}}$  contained all the coordinates within a distance of 2 units in the X and Y directions from each coordinate  $c^*$  that formed the solution of the previous iteration of the algorithm.

### 6.3.2 Results

The *sampling and refining* algorithm was implemented in Python and the mixed integer linear programming models were solved using the commercial MILP solver Gurobi. The algorithm took slightly less than 8 hours to solve the problem instance described above. At the first iteration a feasible solution for the *OPLRMPP – Mod* with a value of \$341 million was found in 6.46 hours. Then, at the first refining stage, a second solution was found in about 21 minutes, but no improvement in NPV was made. Finally, a third solution was found in 53 minutes, but again no improvement was made, forcing the algorithm to stop due to the maximum number of no-improvement steps being reached. Table 6.1 summarises the stages of the algorithm, the value of the solutions and the running time.

| Iteration | Algorithm stage | NPV (USD)   | Runtime [sec] / [hours] |
|-----------|-----------------|-------------|-------------------------|
| 1         | sampling        | 341,892,274 | 23,279 / 6.46           |
| 2         | refining 1      | 335,169,155 | 24,541 / 6.82           |
| 3         | refining 2      | 336,410,984 | 27,742 / 7.71           |

Table 6.1: *Sampling and refining* algorithm results.

Figure 6.3 illustrates different views of the solution found by the *sampling and refining* algorithm at each stage. Each colour represent a different pushback. Sub-figures (a), (b) and (c) show the solutions for the first, second and third iteration of the algorithm, respectively.

Table 6.2 and Table 6.3 show the mine schedule and the pushback schedule for the best solution found (first iteration of the algorithm). Table 6.2 summarises the material sent to the processing plant (processed) and the material sent to the waste dump (wasted) per period. Since the program defines the cut-off grade dynamically as part of the optimisation, the total amount of profitable material (ore) is about 57.2 million tonnes and the total waste material is about 45.6 million tonnes.

| Period | Material processed [tonnes] | Material wasted [tonnes] | Total [tonnes] |
|--------|-----------------------------|--------------------------|----------------|
| 1      | 12,500,000                  | 7,599,159                | 20,099,159     |
| 2      | 12,500,000                  | 9,999,912                | 22,499,912     |
| 3      | 12,500,000                  | 9,981,076                | 22,481,076     |
| 4      | 12,500,000                  | 9,990,298                | 22,490,298     |
| 5      | 5,963,538                   | 4,349,001                | 10,312,539     |
| 6      | 1,313,130                   | 3,704,737                | 5,017,867      |
| Total  | 57,276,668                  | 45,624,183               | 102,900,851    |

Table 6.2: Material sent to process and to waste dump per period.

It can be seen from Table 6.3 that the mining and the processing plant reached their maximum capacity during the first four periods of the mine schedule. However, a steep drop in production occurs in periods 5 and 6. The production of period 6 could be moved forward to period 5 to produce a better solution (which does not violate any of the problem constraints). A possible explanation for this behaviour is due to the small window selected, compared to the total number of periods. A larger time window might produce

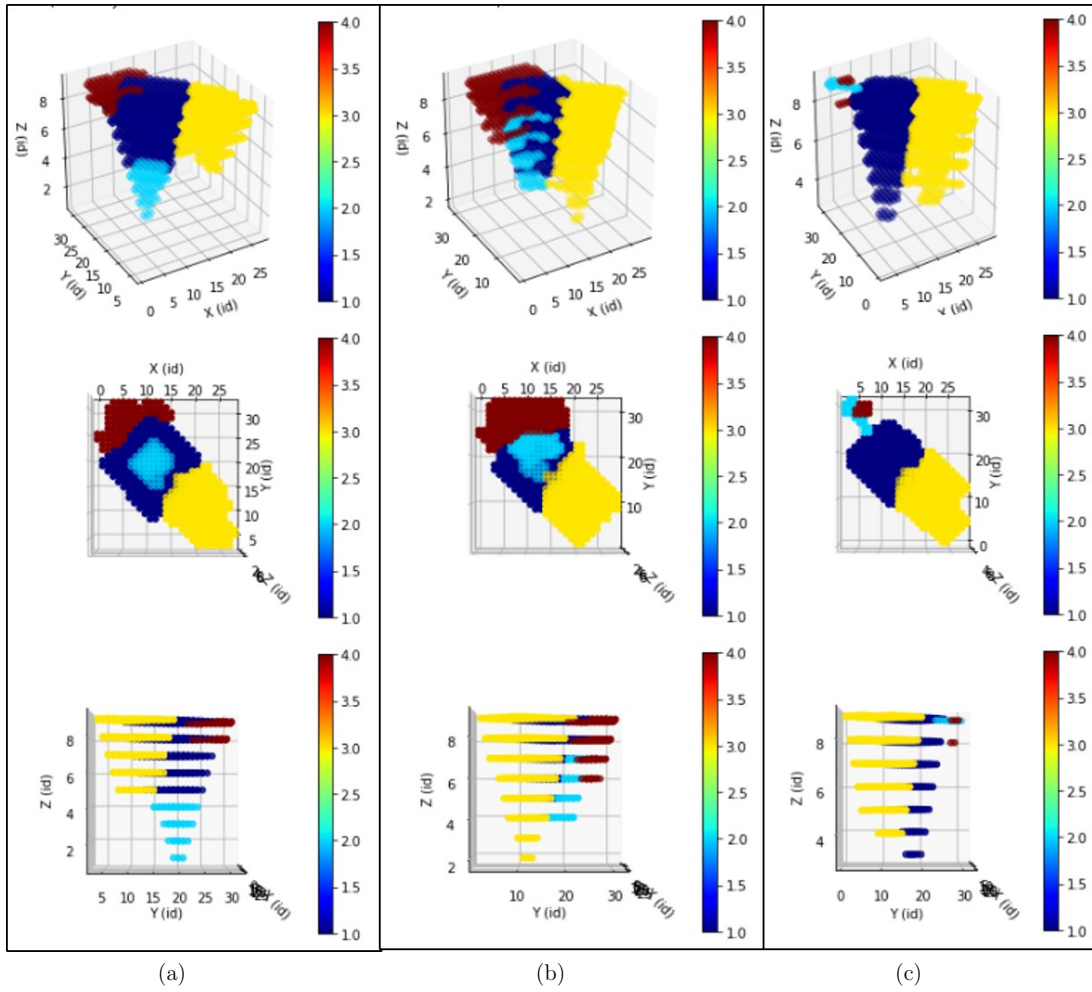


Figure 6.3: Pushbacks found using the *sampling and refining* algorithm. Colours represent different pushbacks. Figures (a), (b) and (c) show the solutions at different view points for the first, second and third iteration of the algorithm, respectively.

a better schedule, but the computational complexity of the MILP problems increases considerably when larger windows are considered (the *sampling and refining* algorithm could not produce a feasible solution in more than 72 hours when a sliding time window of parameters  $I_{win} = 1$  and  $C_{win} = 2$  was considered).

| Pushback       | Periods    |            |            |            |            |           | Total [tonnes] |
|----------------|------------|------------|------------|------------|------------|-----------|----------------|
|                | 1          | 2          | 3          | 4          | 5          | 6         |                |
| 1              | 20,099,159 | 19,962,070 | 16,315,872 |            |            |           | 56,377,102     |
| 2              |            | 2,537,841  | 6,165,204  | 393,271    |            |           | 9,096,317      |
| 3              |            |            |            | 21,128,746 | 8,675,120  |           | 29,803,866     |
| 4              |            |            |            | 968,281    | 1,637,419  | 5,017,867 | 7,623,567      |
| Total [tonnes] | 20,099,159 | 22,499,912 | 22,481,076 | 22,490,298 | 10,312,539 | 5,017,867 | 102,900,851    |

Table 6.3: Pushback schedule.

Finally, it is important to highlight that even though the problem instance studied in this chapter is almost identical to the one presented in Chapter 4, both results are not comparable for a few reasons. The first reason is that each pushback showed in Chapter 4 includes a ramp, while the pushbacks presented in this section do not have a ramp since we relaxed the ramp condition to keep the problem tractable. Second, the pushbacks in Chapter 4 are mined bench by bench (all the blocks that belong to the same pushback and bench are mined in the same period) a constraint that is not modelled in the *OPLRMPP – Mod* to keep the model linear. Third, the model introduced in Chapter 4 considers a fixed cut-off grade (which means that the best destination for the blocks is defined beforehand), whereas the *OPLRMPP – Mod* is able to find the best destination for the blocks during the optimisation. However, the NPV of the solution found was approximately 50% greater than the NPV of the solution shown in Chapter 4. Although the two instances are not 100% comparable, this number serves as an estimation of the value that can be gained by solving the OPLRMPP as an integrated problem.

## 6.4 Conclusions

One of the key problems in open-pit long-range mine planning consists of defining the sequence of pushbacks, the production schedule and the cut-off strategy that maximises the net present value (NPV), defined here as the open-pit long-range mine planning problem (OPLRMPP).

In this chapter, a mixed integer linear programming model (MILP) for the OPLRMPP is introduced. The model allows us to find the set of pushbacks and the cut-off strategy that maximises the NPV of the production schedule of the mine. Additionally, a heuristic method (*sampling and refining*) is introduced. The *sampling and refining* heuristic is used in combination with other techniques such as the early and late start algorithms [39] and the Sliding Time Window Heuristic [24] to improve the tractability of the problem.

A mining instance almost identical to the one considered in Chapter 4 was considered. The solution time for the problem instance that consisted of 3,445 aggregated blocks, six periods and four pushbacks was slightly less than 8 hours. The NPV obtained was 53% greater than the value obtained in Chapter 4. Even though the solutions cannot be directly compared (because the value obtained in Chapter 4 includes the ramp design and considers a pushback-bench schedule instead of a direct block schedule), the significant difference in value obtained serves as a strong indication of the value that could be captured when an integrated approach is considered.

However, the shape of the pushbacks (obtained by using this approach) is limited by the shape of the rectangular templates used in the formulation of the problem. Also, the significant number of binary variables associated with each block, and the mathematical structure of the MILP formulation, precludes the solution of problem instances with more than 5,000 blocks (within the pit boundary) in less than 24 hours with the methods introduced here.

## 6.5 Summary

One of the key tasks in open-pit long-range mine planning is to define the mining sequence (pushbacks), the production schedule and the cut-off strategy that maximise the NPV associated with the extraction of the mine. This problem is referred to as the open-pit long-range mine planning problem (OPLRMPP). Traditionally, the solution to the OPLRMPP has been addressed by solving a series of sequential sub-problems or by solving production schedules that not consider pushbacks and workability conditions (such as connectivity between the blocks, minimum operational width, and access roads), which usually ends in sub-optimal or impractical solutions. To the best of our knowledge, no models for the OPLRMPP have appeared in the literature before.

In this chapter, a mixed integer linear programming model for the OPLRMPP which can be solved exactly is introduced. The model allows us to find the set of pushbacks and the cut-off strategy that maximises the NPV of the production schedule of the mine. A computational experiment showed that significant value could be captured when the integrated problem is considered.



# Chapter 7

## Conclusions

*There are, however, virtually unlimited numbers of ways of reaching a final contour.  
— Helmut Lerchs & Ingo Grossmann [60].*

**T**HE work presented in this thesis has made some contributions towards models, methods and approaches for optimising mine planning and design of open-pit mines. Chapters 1 and 2 described some of the challenges associated with long-range open-pit mine planning, such as pushback design optimisation, haulage road designs and production scheduling. Early models and approaches from the literature and standard industry practices were discussed. Additionally, a formal definition of both the open-pit long-range mine planning problem (OPLRMPP) and of mining pushbacks was introduced, and the relevance of including workability conditions into optimisation models for pushback designs was established.

In Chapter 3, a solution for the open pit ramp design problem was introduced. A shortest path method was developed for the ex-pit road design, and a binary linear programming approach was developed for the in-pit ramp design. Both models allow gradient and curvature constraints to be handled within the given discretisation. They can incorporate construction and haulage costs, and the formulations are flexible enough to find an optimal ramp that avoids any given forbidden region.

Some limitations are worth noting. The in-pit model assumes that the optimal ramp occurs on the boundary of a precomputed pushback, which might be suboptimal compared to the problem of finding a ramp and a pushback simultaneously (this problem is studied in Chapter 4). Also, a combined in-pit ramp design and scheduling optimisation problem, that takes into account the way properties of the haulage network interact with the mining schedule, would be another worthwhile avenue for future work.

Chapter 4 introduced a mixed integer linear programming model for the pushback design problem. The central idea of the new formulation was to use the in-pit ramp to control the shape of the pushback to give more practical designs. The proposed formulation extends previous mixed integer linear programming formulations for the pushback design problem such as [7] or [66]. Compared to the traditional methodology, an improvement of 5.4% for the NPV was obtained.

However, the formulation presented enforces the connectivity and mining width constraints by biasing the objective function, requiring an extra parameter  $\lambda$  to be calibrated. A different approach that does not require a parameter  $\lambda$  might be worth seeking. Also, other heuristic methods might improve the results obtained from the greedy approach. Solutions for these issues are proposed in Chapter 5.

In Chapter 5, two formulations for semi-practical pushbacks that maximise an approximate NPV were studied. Both binary linear programming models enforced mining width conditions and block connectivity within the same pushback without requiring extra parameter calibration (like the solution proposed in Chapter 4). The formulations differ in the way in which the connectivity condition was modelled. Computational experiments showed that the use of techniques based on the Sliding Time Window Heuristic (STWH) [24] and the early- and late-start algorithms [39], significantly reduced the CPU time for the solution of the problem. The largest size instance tested consisted of 11,076 blocks with five pushbacks and took approximately 1.7 hours to find a solution within a 10% gap from the linear relaxation.

However, the efficiency of the solution method based on the STWH is limited to the size of the sub-problems. According to preliminary tests, the method reached its limit when instances with more than 20,000 blocks were considered. Additionally, since the production schedule is not embodied in the formulation (it maximises an approximate NPV), there is no guarantee that the pushbacks obtained with this method will drive the optimal NPV when a detailed schedule is considered. To tackle the latter issue, a new formulation that considers the mine production schedule is introduced in Chapter 6.

Finally, in Chapter 6, a mathematical formulation for the OPLRMPP is introduced. The mathematical model -which is a combination of the approaches taken in Chapters 4 and 5- answers the three main questions described in Chapter 1 regarding the OPLRMPP. The model finds the set of pushbacks (according to the definition introduced in Chapter 1) and the cut-off grade strategy that maximises the NPV of the production schedule. The solution method proposed combines the relaxation of the haulage ramp's gradient, the use of the STWH, and the use of a Sampling and Refining heuristic. Computational experiments, conducted on an instance almost identical to the one considered in Chapter 4, took slightly less than 8 hours to solve. The NPV of the solution found was approximately 50% greater than the NPV of the solution shown in Chapter 4. Although the two instances are not 100% comparable, this number serves as an estimation of the value that can be gained by solving the OPLRMPP as an integrated problem.

Nevertheless, the shapes obtained by the new model are restricted to the shape of the templates, and only instances with less than 5000 blocks (4 pushbacks and six periods) were able to be solved in less than 24 hours using the solution method described.

This thesis has focused on the identification, conceptualisation, and modelling of some of the critical operating conditions of open-pit mines thus developing new mathematical approaches for the OPLRMPP that produce operationally feasible solutions. The question of whether there are other solution techniques or alternative mathematical formulations that will allow solving problem instances with millions of blocks remains open.

# Bibliography

- [1] Ravindra K. Ahuja, Thomas Magnanti, and James B. Orlin. *Network Flows*. Upper Saddle River, New Jersey: Prentice-Hall, 1993, p. 846 (cit. on pp. 29, 38).
- [2] Atsushi Akaike. “Strategic planning of Long term production schedule using 4D network relaxation method”. PhD thesis. Colorado School of Mines, 1999 (cit. on p. 23).
- [3] Abdullah E Akay et al. “Using optimization techniques in designing forest roads and road networks”. In: *Journal of the Faculty of Forestry Istanbul University* 15.1 (2013), pp. 49–62 (cit. on p. 21).
- [4] Christopher Alford and Jeff Whittle. “Application of lerchs-grossman pit optimization to the design of open pit mines”. In: *Large Open Pit Mining Conference* October (1986), pp. 201–207. ISSN: 03146154 (cit. on pp. 13, 14, 16).
- [5] Axel E Anderson and John Nelson. “Projecting vector-based road networks with a shortest path algorithm”. In: *Canadian Journal of Forest Research* 34.7 (2004), pp. 1444–1457. ISSN: 0045-5067. DOI: 10.1139/x04-030 (cit. on pp. 22, 31).
- [6] Felipe Arteaga, Micah Nehring, and Peter Knights. “The equipment utilisation versus mining rate trade-off in open pit mining”. In: *International Journal of Mining, Reclamation and Environment* 32.7 (2018), pp. 495–518. ISSN: 17480949. DOI: 10.1080/17480930.2017.1306674. URL: <http://dx.doi.org/10.1080/17480930.2017.1306674> (cit. on p. 15).
- [7] Hooman Askari-Nasab, Clemens Mieth, and Mohammad Tabesh. “A hybrid local search/mixed integer programming approach to open pit controlled phase-design”. In: *36th Applications of Computers and Operations Research in Mineral Industry*. Porto Alegre, 2013, pp. 502–514 (cit. on pp. 18, 63, 109).
- [8] Thomas Atkinson. “Design and Layout of Haul Roads”. In: *SME Mining Engineering Handbook*. Ed. by Howard L. Hartman. 2nd Editio. Littleton, Colorado: Society for Mining, Metallurgy, and Exploration, 1992. Chap. 13.4, pp. 1334–1342 (cit. on pp. 5, 22).
- [9] Xiaoyu Bai et al. *Pseudoflow Method for Pit Optimization*. Tech. rep. 2017, pp. 1–9. URL: <https://www.3ds.com/fileadmin/PRODUCTS-SERVICES/GEOVIA/PDF/whitepaper/2017-GEOVIA-WHITEPAPER-PSEUDOFLOW.pdf> (cit. on p. 15).

- [10] X Bai et al. "Automatic generation of feasible mining pushbacks for open pit strategic planning". In: *SAIMM* 118.5 (2018), pp. 515–530 (cit. on p. 21).
- [11] Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Belmont, Massachusetts: Athena Scientific, 1997, p. 587 (cit. on p. 29).
- [12] Daniel Bienstock and Mark Zuckerberg. "Solving LP relaxations of large-scale precedence constrained problems". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6080 LNCS (2010), pp. 1–14. ISSN: 03029743 (cit. on pp. 9, 24).
- [13] Andreas Bley et al. "A strengthened formulation and cutting planes for the open pit mine production scheduling problem". In: *Computers and Operations Research* 37.9 (2010), pp. 1641–1647. ISSN: 03050548. DOI: 10.1016/j.cor.2009.12.008 (cit. on pp. 20, 24, 49, 52, 71).
- [14] Natashia Boland et al. "LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity". In: *Computers and Operations Research* 36.4 (2009), pp. 1064–1089. ISSN: 03050548. DOI: 10.1016/j.cor.2007.12.006 (cit. on p. 93).
- [15] R. Burdett, E. Kozan, and R. Kenley. "Block models for improved earthwork allocation planning in linear infrastructure construction". In: *Engineering Optimization* 47.3 (2015), pp. 347–369. ISSN: 10290273. DOI: 10.1080/0305215X.2014.892594 (cit. on p. 22).
- [16] L Caccetta and LM Giannini. "An application of discrete mathematics in the design of an open pit mine". In: *Discrete applied mathematics* 21 (1988), pp. 1–19. DOI: 10.1016/0166-218X(88)90030-3. URL: <http://www.sciencedirect.com/science/article/pii/0166218X88900303> (cit. on p. 14).
- [17] Louis Caccetta and Stephen P. Hill. "An application of branch and cut to open pit mine scheduling". In: *Journal of Global Optimization* 27.2-3 (2003), pp. 349–365. ISSN: 09255001. DOI: 10.1023/A:1024835022186 (cit. on p. 23).
- [18] Thomas Carlson et al. "Computer Techniques in Mine Planning". In: *Mining Engineering* 5.18 (1966), pp. 53–56 (cit. on p. 12).
- [19] Gerardo Casal, Duarte Santamarina, and Miguel E. Vázquez-Méndez. "Optimization of horizontal alignment geometry in road design and reconstruction". In: *Transportation Research Part C: Emerging Technologies* 74 (2017), pp. 261–274. ISSN: 0968090X. DOI: 10.1016/j.trc.2016.11.019 (cit. on p. 21).
- [20] Alan J. Chang et al. "Curvature-constrained directional-cost paths in the plane". In: *Journal of Global Optimization* 53.4 (2012), pp. 663–681. ISSN: 09255001. DOI: 10.1007/s10898-011-9730-1 (cit. on p. 21).
- [21] Renaud Chicoisne et al. "A new algorithm for the open-pit mine production scheduling problem". In: *Operations Research* 60.3 (2012), pp. 517–528. ISSN: 0030-364X. DOI: 10.1287/opre.1120.1050 (cit. on p. 52).

- [22] F. R. Albor Consuegra and Roussos Dimitrakopoulos. "Algorithmic approach to pushback design based on stochastic programming: method, application and comparisons". In: *Mining Technology* 119.2 (2010), pp. 88–101. ISSN: 1474-9009. DOI: 10.1179/037178410X12780655704761 (cit. on pp. 15, 19).
- [23] T. Couzens. "Aspects of production planning: Operating layout and phase plans". In: *SME Mining Engineering Handbook*. Ed. by J. Crawford and W A Hustrulid. New York: Society of Mining Engineering, 1979. Chap. 16, pp. 217–231 (cit. on pp. 15, 22).
- [24] Christopher Cullenbine, R. Kevin Wood, and Alexandra Newman. "A sliding time window heuristic for open pit mine block sequencing". In: *Optimization Letters* 5.3 (2011), pp. 365–377. ISSN: 18624472. DOI: 10.1007/s11590-011-0306-2 (cit. on pp. 23, 24, 63, 78, 88, 89, 98, 106, 110).
- [25] Kadri Dagdelen. "Optimum multi-period open-pit mine production scheduling". PhD thesis. Colorado School of Mines, 1985 (cit. on p. 23).
- [26] M.. David, P. Dowd, and S. Korobov. "Forecasting departure from planning in open pit design and grade control". In: *Application of Computers Methods in the Mineral Industry*. 1974, pp. 1–23 (cit. on p. 13).
- [27] Roussos Dimitrakopoulos, L Martinez, and S. Ramazan. "A maximum upside minimum downside approach to the traditional optimization of open pit mine design". In: *Journal of Mining Science* 43.1 (2007), pp. 73–82. ISSN: 1062-7391. DOI: 10.1007/s10913-007-0009-3 (cit. on p. 17).
- [28] P. Dowd and A. Onur. "Optimising Open Pit Design and Sequencing". In: *23rd International Symposium of Application of Computers and Operations Research*. 1992, pp. 411–422. ISBN: 0-87335-110-X (cit. on pp. 13, 15, 22, 27).
- [29] Anita Eccles. "Take time for pit phase design". In: *2013 SME Annual Meeting & Exhibit*. Denver, Colorado: SME, 2013, pp. 468–472 (cit. on pp. 4, 11, 15, 45).
- [30] T. Elkington and R. Durham. "Integrated open pit pushback selection and production capacity optimization". In: *Journal of Mining Science* 47.2 (2011), pp. 177–190 (cit. on pp. 15, 18).
- [31] R. Epstein et al. "Optimizing long-term production plans in underground and open-pit copper mines". In: *Operations Research* 60.1 (2012), pp. 4–17. ISSN: 0030-364X. DOI: 10.1287/opre.1110.1003 (cit. on p. 93).
- [32] Rafael Epstein et al. "A Combinatorial Heuristic Approach for Solving Real-Size Machinery Location and Road Design Problems in Forestry Planning". In: *Operations Research* 54.6 (2007), pp. 1017–1027. ISSN: 0030-364X. DOI: 10.1287/opre.1060.0331 (cit. on p. 29).
- [33] Rafael Epstein et al. "PLANEX: A system to identify landing locations and access". In: *Proc. of the International Mountain Logging and 11th Pacific Northwest Skyline Symposium* (2001), pp. 190–193 (cit. on pp. 22, 29).
- [34] Daniel Espinoza et al. "MineLib: A library of open pit mining problems". In: *Annals of Operations Research* 206.1 (2013), pp. 93–114. ISSN: 02545330. DOI: 10.1007/s10479-012-1258-3 (cit. on pp. 66, 77, 82, 83).

- [35] Iain Farmer and Roussos Dimitrakopoulos. "Schedule-based pushback design within the stochastic optimisation framework". In: *International Journal of Mining, Reclamation and Environment* 32.5 (2018), pp. 327–340. ISSN: 17480949. DOI: 10.1080/17480930.2017.1289606 (cit. on p. 20).
- [36] Dominique François-Bongarçon and Alain Merechal. "A new method for open pit design: parametrization of the final pit contour". In: *Application of Computers Methods in the Mineral Industry - 14th International Symposium*. 14. 1976, pp. 573–583 (cit. on pp. 14, 16).
- [37] Christopher Fricke. "Applications of Integer Programming in Open Pit Mining". In: August (2006) (cit. on p. 15).
- [38] Gary Froyland and Merab Menabde. *System and method(s) of mine planing, design and processing*. US Patent 7,519,515. 2011 (cit. on pp. 20, 25).
- [39] Martin Gaupp. "Methods for improving the tractability of the block sequencing problem for open pit mining". PhD thesis. Colorado School of Mines, 2008, pp. 1–159 (cit. on pp. 23, 78, 87, 98, 106, 110).
- [40] J Gholamnejad and M. Osanloo. "Uncertainty Based Push-Back Design Algorithm in Open Pit Mining". In: *SME Annual Meeting*. Denver, Colorado: SME, 2007, Paper 07–095 (cit. on pp. 11, 15, 16).
- [41] Tim Gill. "Express Road routing: The application of an optimal Haul road generator to real world data". In: *Optimizing with Whittle*. Ed. by Jeff Whittle. Perth, Western Australia: Whittle Programming Pty. Ltd., 1999, pp. 71–80 (cit. on p. 22).
- [42] Tim Gill. "Express Road routing: The application of an optimal Haul road generator to real world data". In: *Strategic mine planing*. Ed. by Jeff Whittle. Perth, Western Australia: AusIMM, 2001, pp. 71–80 (cit. on p. 22).
- [43] Tim Gill, M Robey, and T Caelll. "A survey of current approaches to the optimum ultimate open-pit planning problem". In: 1 (1996), pp. 29–37 (cit. on pp. 15, 91).
- [44] R Goodfellow and Roussos Dimitrakopoulos. "Algorithmic integration of geological uncertainty in pushback designs for complex multiprocess open pit mines." In: *Mining Technology* 122.2 (2013), pp. 67–77. ISSN: 14749009. DOI: 10.1179/1743286312Y.0000000020 (cit. on p. 19).
- [45] Hans Rudolf Heinimann, Jürg Andreas Stükelberger, and Woodam Chung. "Improving Automatic Grid Cell Based Road Route Location Procedures". In: *Austro2003: High Tech Forest Operations for Mountainous Terrain*. Ed. by K. Stampfer. Schlaegl - Austria, 2003, pp. 1–10 (cit. on p. 22).
- [46] D S Hochbaum and a Chen. "Performance analysis and best implementations of old and new algorithms for the open-pit mining problem". In: *Operations Research* 48.6 (2000), pp. 894–914. ISSN: 0030-364X. DOI: 10.1287/opre.48.6.894.12392 (cit. on pp. 4, 14, 15).
- [47] Dorit S. Hochbaum. "The pseudoflow algorithm: a new algorithm for the maximum-flow problem". In: *Operations Research* 56.4 (2008), pp. 992–1009. ISSN: 0030-364X. DOI: 10.1287/opre.1080.0524 (cit. on p. 14).

- [48] W A Hustrulid, M Kuchta, and R. Martin. *Open pit mine planning and design*. 3rd. CRC Press, 2013, p. 1288. ISBN: 9781466575127 (cit. on pp. 1, 15).
- [49] E. Jelvez, N. Morales, and Hooman Askari-Nasab. "A new model for automated pushback selection". In: *Computers and Operations Research* (2018), pp. 1–9. DOI: <https://doi.org/10.1016/j.cor.2018.04.015> (cit. on p. 18).
- [50] TB Johnson. "Optimum open pit mine production scheduling." In: *A Decade Of Digital Computing In The Mineral Industry* (1969), pp. 539–563. URL: <http://oai.dtic.mil/oai/oai?verb=getRecord%7B%5C%7DmetadataPrefix=html%7B%5C%7Didentifier=AD0672094> (cit. on p. 23).
- [51] Thys B Johnson and Wiliam R Sharp. *A Three-Dimensional Dynamic Programing Method for Optimal Ultimate Open Pit Design*. Tech. rep. U.S. Dept. of the Interior, Bureau of Mines, 1971 (cit. on pp. 13, 14).
- [52] G Juarez et al. "Open pit strategic mine planning with automatic phase generation". In: *Orebody Modelling and Strategic Mine Planning Symposium*. Ed. by R. Dimitrakopoulos. Perth, WA: AusIMM, 2014, pp. 147–153 (cit. on p. 19).
- [53] Walter W. Kaufman and James C. Ault. *Design of Surface Mine Haulage Roads: A Manual*. Tech. rep. Washington: United States. Department of the Interior, Bureau of Mines., 1977, pp. 1–50 (cit. on p. 22).
- [54] Y C Kim and Y Zhao. "Optimum open pit production sequencing - current state of art". In: *Society for Mining, Metallurgy, and Exploration*. Albuquerque, New Mexico: SME, 1994, Paper number 94–224 (cit. on p. 11).
- [55] Young C Kim. "Ultimate Pit Limit Design Methodologies Using Computer Models - The State of the Art". In: October 1978 (1979) (cit. on p. 15).
- [56] Benjamin Koskieniemi. "Hand methods". In: *SME Mining Engineering Handbook*. Ed. by J. Crawford and W A Hustrulid. New York: Society of Mining Engineers of American Institute of Mining, Metallurgical, and Petroleum Engineers, 1979. Chap. 13, pp. 189–194 (cit. on p. 12).
- [57] W. Brian Lambert et al. "Open-pit block-sequencing formulations: A tutorial". In: *Interfaces* 44.2 (2014), pp. 127–142. ISSN: 1526551X. DOI: 10.1287/inte.2013.0731 (cit. on p. 24).
- [58] Marc Lemieux. "A different method of modeling a mineral deposit for a three-dimensional open pit computer design application". In: *Application of Computers Methods in the Mineral Industry - 14th International Symposium*. New York: Society of Mining Engineers of American Institute of Mining, Metallurgical, and Petroleum Engineers, 1977, pp. 557–572 (cit. on pp. 12, 13, 24).
- [59] Marc Lemieux. "Moving Cone Optimizing Algorithm". In: *Computer Methods for the 80's In the Mineral Industry* (1979), pp. 329–345 (cit. on p. 13).
- [60] H. Lerchs and I. F. Grossmann. "Optimum design of open pit mines". In: *Canadian Institute of Mining and Metallurgy Bulletin* 58.633 (1965), pp. 47–54 (cit. on pp. 4, 9, 13–16, 48, 109).

- [61] Michael Lipkewich and Leon Borgman. "Two- And Three- Dimensional Pit Design Optimization Techniques". In: *Application of Computers and Operations Research in the Mineral Industry*. The American Institute of Mining, Metallurgical, and Petroleum Engineers, 1969, pp. 505–523. arXiv: arXiv : 1011 . 1669v3 (cit. on p. 14).
- [62] Kevin Liu and John Sessions. "Preliminary Planning of Road Systems Using Digital Terrain Models". In: *Journal of Forest Engineering* 4.2 (1993), pp. 27–32 (cit. on p. 22).
- [63] Shi Qiang Liu and Erhan Kozan. "New graph-based algorithms to efficiently solve large scale open pit mining optimisation problems". In: *Expert Systems with Applications* 43 (2016), pp. 59–65. ISSN: 09574174. DOI: 10 . 1016 / j . eswa . 2015 . 08 . 044. URL: <http://dx.doi.org/10.1016/j.eswa.2015.08.044> (cit. on p. 52).
- [64] C Meagher, Roussos Dimitrakopoulos, and D Avis. "Optimized open pit mine design, pushbacks and the gap problem-a review." In: *Journal of Mining Science* 50.3 (2014), pp. 508–526. ISSN: 10627391. DOI: 10 . 1134 / S1062739114030132 (cit. on pp. 15, 63).
- [65] C Meagher, S a Abdel Sabour, and R Dimitrakopoulos. "Pushback Design of Open Pit Mines Under Geological and Market Uncertainties". In: *Orebody Modelling and Strategic Mine Planning* 17.March (2009), pp. 16–18 (cit. on p. 17).
- [66] Conor Meagher, Roussos Dimitrakopoulos, and Victor Vidal. "A new approach to constrained open pit pushback design using dynamic cut-off grades". In: *Journal of Mining Science* 50.4 (2014), pp. 733–744. DOI: 10 . 1134 / S1062739114040140 (cit. on pp. 15, 18, 85, 88, 109).
- [67] David Meignan et al. "A heuristic approach to automated forest road location". In: *Canadian Journal of Forest Research* 42.12 (2012), pp. 2130–2141. ISSN: 0045-5067. DOI: 10 . 1139 / x2012-140 (cit. on p. 22).
- [68] Manfred Meyer. "APPLYING LINEAR PROGRAMMING TO THE DESIGN OF ULTIMATE PIT LIMITS". In: *Management Science* Oct 16.2 (1969), pp. 1986–121 (cit. on pp. 12, 14).
- [69] Sukanto Mondal, Yves Lucet, and Warren Hare. "Optimizing horizontal alignment of roads in a specified corridor". In: *Computers and Operations Research* 64 (2015), pp. 130–138. ISSN: 03050548. DOI: 10 . 1016 / j . cor . 2015 . 05 . 018. arXiv: arXiv:1507.02714v1 (cit. on p. 22).
- [70] Eduardo Moreno et al. "Linear models for stockpiling in open-pit mine production scheduling problems". In: *European Journal of Operational Research* 260.1 (2017), pp. 212–221. ISSN: 03772217. DOI: 10 . 1016 / j . ejor . 2016 . 12 . 014. URL: <http://dx.doi.org/10.1016/j.ejor.2016.12.014> (cit. on p. 93).
- [71] D. C. W. Muir. "Pseudoflow, new life for Lerchs-Grossmann pit optimisation". In: *Orebody Modelling and Strategic Mine Planning*. Ed. by R. Dimitrakopoulos and Salih Ramazan. Perth, WA: Australasian Institute of Mining and Metallurgy, 2004, pp. 39–46 (cit. on p. 15).

- [72] Gonzalo Muñoz et al. *A study of the Bienstock–Zuckerberg algorithm: applications in mining and resource constrained project scheduling*. Vol. 69. 2. Springer US, 2018, pp. 501–534. ISBN: 1058901799461. DOI: 10.1007/s10589-017-9946-1. arXiv: 1607.01104 (cit. on p. 24).
- [73] Akbar Najafi and Evelyn W. Richards. “Designing a forest road network using mixed integer programming”. In: *Croatian Journal of Forest Engineering*. 34.1 (2013), pp. 17–30 (cit. on p. 22).
- [74] Alexandra M. Newman et al. “A review of operations research in mine planning”. In: *Interfaces* 40.3 (2010), pp. 222–245. ISSN: 00922102. DOI: 10.1287/inte.1090.0492 (cit. on pp. 22, 32).
- [75] Noel O’Brien and F.J. Nowak. “An Application of a Computer to Open-Pit Mine Design”. In: *Pit and Quarry* 58.2 (1966), pp. 128–135. ISSN: 0010-4620. DOI: 10.1093/comjnl/1.2.64. URL: <http://comjnl.oupjournals.org/cgi/doi/10.1093/comjnl/1.2.64> (cit. on p. 12).
- [76] Jean-Claude Picard. “Maximal Closure of a Graph and Applications to Combinatorial Problems”. In: *Management Science* 22.11 (1976), pp. 1268–1272. ISSN: 00251909. DOI: 10.1287/mnsc.22.11.1268. URL: <http://www.jstor.org/stable/2630227> (cit. on p. 14).
- [77] Cristian J Poblete et al. “Impact of the starting point and of the direction of open pit exploitation on the mining plan”. In: *Sixth International Conference on Innovation in Mine Operations*. May. Santiago, Chile., 2016 (cit. on p. 16).
- [78] Julian Poniewierski. *Pseudoflow Explained*. Tech. rep. Deswik, 2017, pp. 1–10. URL: <https://www.deswik.com/wp-content/uploads/2017/05/Pseudoflow-Explained.pdf> (cit. on p. 15).
- [79] Yasha Pushak, Warren Hare, and Yves Lucet. “Multiple-path selection for new highway alignments using discrete algorithms”. In: *European Journal of Operational Research* 248.2 (2016), pp. 415–427. ISSN: 03772217. DOI: 10.1016/j.ejor.2015.07.039 (cit. on p. 22).
- [80] S. Ramazan. “A new push back design algorithm in open pit mining”. PhD thesis. Colorado School of Mines, 1996, p. 150 (cit. on pp. 4, 15).
- [81] S. Ramazan, K. Dagdelen, and Thys B. Johnson. “Fundamental tree algorithm in optimising production scheduling for open pit mine design”. In: *Mining Technology* 114.1 (2005), pp. 45–54. ISSN: 03717844. DOI: 10.1179/037178405X44511 (cit. on p. 25).
- [82] S Ramazan and K Dagdelen. “A new push back design algorithm in open pit mining”. In: *Proceedings of 7th MPES conference, Calgary, Canada*. 1998, pp. 119–124 (cit. on p. 16).
- [83] H. Ren and Erkan Topal. “Using Clustering Methods for Open Pit Mine Production Planning”. In: *Journal of research project review* 3.1 (2014), pp. 45–49 (cit. on p. 25).

- [84] M. Rezaee, M. Osanloo, and R.Q. Honaker. "Optimizing the number of open pit mine push backs". In: *2012 SME Annual Meeting and Exhibit 2012, SME 2012, Meeting Preprints* (2012), pp. 1–8 (cit. on p. 15).
- [85] Orlando Rivera Letelier et al. "(to appear) Production scheduling for strategic open pit mine planning: A mixed integer programming approach". In: *Operational Research* () (cit. on p. 63).
- [86] Mike Robb. "Automated methods of final pit delimitation". In: *SME Mining Engineering Handbook*. Ed. by J. Crawford and W A Hustrulid. New York: Society of Mining Engineers of American Institute of Mining, Metallurgical, and Petroleum Engineers, 1979. Chap. 14, pp. 197–202 (cit. on pp. 11, 12, 15, 16, 65).
- [87] Chotipong Somrit and Kadri Dagdelen. "Max flow – Lagragian based phase design algorithm for open pit production scheduling optimization". In: *36th Applications of Computers and Operations Research in the Mineral Industry*. 2013, pp. 401–414 (cit. on p. 17).
- [88] Jürg Andreas Stükelberger. "A Weighted-Graph Optimization Approach for Automatic Location of Forest Road Networks". PhD thesis. ETH Zurich, 2008, p. 127. ISBN: 9783728132178 (cit. on p. 22).
- [89] M Tabesh and H Askari-Nasab. "Two-stage clustering algorithm for block aggregation in open pit mines". In: *Mining Technology* 120.3 (2011), pp. 158–169. ISSN: 1474-9009. DOI: 10.1179/1743286311Y.0000000009 (cit. on pp. 18, 25).
- [90] Mohammad Tabesh and Hooman Askari-Nasab. *Open pit production planning using controled pushbacks and aggregates*. Montreal, 2013 (cit. on pp. 17, 18, 85, 88).
- [91] Beyime Tachefine and François Soumis. "Maximal closure on a graph with resource constraints". In: *Computers & Operations Research* 24.10 (1997), pp. 981–990. DOI: 10.1016/S0305-0548(97)00008-7. URL: <http://www.sciencedirect.com/science/article/pii/S0305054897000087> (cit. on p. 17).
- [92] Dwayne Tannant and Bruce Regensburg. *Guidelines for Mine Haul Road Design*. University. Edmonton: University of Alberta Press, 2001, pp. 1–111 (cit. on p. 22).
- [93] Robert Tarjan. "Depth-First Search and Linear Graph Algorithms". In: *SIAM Journal on Computing* 1.2 (1972), pp. 146–160. ISSN: 0097-5397. DOI: 10.1137/0201010 (cit. on p. 35).
- [94] Michael D. Teter, Alexandra M. Newman, and Martin Weiss. "Consistent notation for presenting complex optimization models in technical writing". In: *Surveys in Operations Research and Management Science* 21.1 (2016), pp. 1–17. ISSN: 18767354. DOI: 10.1016/j.sorms.2016.05.001 (cit. on pp. 47, 93).
- [95] Roger J Thompson. "Design , Construction , and Maintenance of Haul Roads". In: *SME Mining Engineering Handbook*. Ed. by P. Darling. Society for Mining, Metallurgy, and Exploration, 2011. Chap. 10.6, pp. 957–976 (cit. on p. 5).
- [96] Robert Underwood and Boleslaw Tolwinski. "A mathematical programming viewpoint for solving the ultimate pit problem". In: *European Journal of Operational Research* 107.1 (1998), pp. 96–107. ISSN: 03772217. DOI: 10.1016/S0377-2217(97)00141-0 (cit. on p. 14).

- [97] Q Wang and H Sevim. "Alternative to parameterization in finding a series of maximum-metal pits for production planning". In: *Mining engineering* 47.2 (1995), pp. 178–182 (cit. on p. 17).
- [98] Andrew Wetherelt and Klaas Peter Van Der Wielen. "Introduction to open-pit mining". In: *SME - Mining Engineering Handbook*. Ed. by P. Darling. 3rd. Society for Mining, Metallurgy, and Exploration, 2011. Chap. 10.1, pp. 857–876 (cit. on pp. 2, 3, 11).
- [99] Christopher Wharton and Jeff Whittle. "The effect of minimum mining width on NPV". In: *Optimizing with Whittle*. Ed. by Jeff Whittle. Perth, Western Australia: Whittle Programming Pty. Ltd., 1997, pp. 173–178 (cit. on pp. 18, 20, 21).
- [100] David Whittle. "Open-pit planning and design". In: *SME Mining Engineering Handbook*. Ed. by Peter Darling. 3rd. Society for Mining, Metallurgy, and Exploration, 2011. Chap. 10.2, pp. 877–901 (cit. on pp. 4, 16, 55).
- [101] J Whittle and L I Rozman. "Open Pit Design in the 90s". In: June (1991), pp. 13–20 (cit. on pp. 15, 16).
- [102] Juan Yarmuch et al. "Evaluating crusher system location in an open pit mine using Markov chains". In: *International Journal of Mining, Reclamation and Environment* 31.1 (2017), pp. 24–37. ISSN: 1748-0930. DOI: 10.1080/17480930.2015.1105649 (cit. on p. 6).
- [103] Meimei Zhang. "Applying simulated annealing to practical mining phase design". In: *34th Application of Computers and Operations Research in the Mineral Industry*. Vancouver: CIM, 2009, pp. 266–273 (cit. on pp. 15, 20).
- [104] Yixian Zhao and Young C Kim. "A new graph theory algorithm for optimal ultimate pit design". In: *Society for Mining, Metallurgy, and Exploration*. Salt Lake City, Utah: SME, 1990, Paper number 90–9 (cit. on p. 14).



# Appendix A

## PCT patent: Mine Planning Method and System

We include here a PCT patent which was written during the PhD candidature. This patent introduces a novel method to model the workability constraints -mining width and connectivity- by using the physical behaviour of soap bubbles.



(51) International Patent Classification:

G06Q 10/04 (2012.01) G06T 17/05 (2011.01)  
E21C 41/00 (2006.01)

(21) International Application Number:

PCT/AU2019/050529

(22) International Filing Date:

28 May 2019 (28.05.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

2018901873 28 May 2018 (28.05.2018) AU

(71) Applicant: THE UNIVERSITY OF MELBOURNE

[AU/AU]; Grattan Street, University Of Melbourne, Victoria 3010 (AU).

(72) Inventors: RUBINSTEIN, Joachim Hyam; c/o The Uni-

versity of Melbourne, Intellectual Property & Licences, Research Innovation & Commercialisation, The University of Melbourne, Victoria 3010 (AU). YARMUCH, Juan Luis; c/o The University of Melbourne, Intellectual Property & Licences, Research Innovation & Commercialisation, The University of Melbourne, Victoria 3010 (AU).

(74) Agent: GRIFFITH HACK; GPO Box 4164, Sydney, New

South Wales 2001 (AU).

(81) Designated States (unless otherwise indicated, for every

kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP,

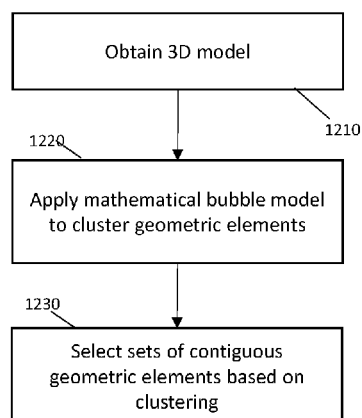
KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: MINE PLANNING METHOD AND SYSTEM



(57) Abstract: A method and system for mine design utilising mathematical modelling and optimisation based on mathematical modelling of connected bubbles. A mathematical model, based on mathematical modelling of behavior of connected bubbles, is applied to the 3D block model for a mine to cluster geometric elements from the 3D block model based on physical location and properties of the physical material for each of the geometric elements, and selecting at least one set of a plurality of contiguous geometric elements for extraction based on the clustering. Each set of a plurality of contiguous blocks can be associated with a phase of a mining process.

Figure 12

## **Mine Planning Method and System**

### **Technical field**

- 5 The field of the invention is methods and systems for mining design and planning. Applications of the invention are in designing for extraction plans, pushbacks and/or ramps and scheduling.

### **Background**

- 10 Mining is the process of extracting valuable material from the earth. Materials extracted from mines include some of the modern world's most important commodities: for example, coal, copper, diamonds, iron, gold, silver, molybdenum, and uranium. After prospecting and geological exploration, mine planners evaluate the mining method,  
15 estimate the production capacity and produce detailed exploitation plans.

- Mine planning and design aim to plan extraction of material from the mine with the objectives of maximizing yield, minimizing operational costs and compliance with regulatory and safety requirements. Ore deposits are extracted using a variety of  
20 mining methods. Among surface mines, quarries and open pit are the most common, while block, panel, sub-level caving and sub-level stopping are usually selected as mining methods in underground mines. The choice of mining method must consider internal and external factors. Natural conditions, investment capability of companies, public policies of the region and the state of art technology should be taken into  
25 consideration to evaluate different methods. Among the natural factors, the spatial location and quality of the geological resources, the rock strength, and the topography are the most relevant aspects to be considered.

- A typical input to the mine planning is a geological model; in this model geological  
30 resources are represented as a block model, a regular-spaced grid of blocks. A 3D block is typically a prismatic shape that is represented by the coordinates of its centroid (x,y,z). This model contains the main information for the planning stage. For instance, rock alteration, mineralisation zones, rock densities and element concentration are fundamental in defining the value of the mine. Mine planners, with the help of other  
35 professional disciplines, are responsible for transforming geological resources to ore reserves.

- In open cut mining, the Ultimate Pit Limit (UPL) defines the mineable volume of material that generates profits. The size and shape of the pit mostly depends on  
40 geotechnical, processing and economic factors. Smaller inner pits are known as intermediate pits. The most common technique used to define the inner pits corresponds to the variation (a fraction) of the commodity selling price. Thus, the ultimate pit limit is calculated for each price. Similar considerations also apply to underground mining or excavation. Working underground can also increase  
45 operational constraints and costs.

The UPL is extracted following a development sequence. Pushbacks, phases or mining cuts, are sub-volumes inside the UPL. These volumes must be logically designed, allowing safe operation of the mining equipment (loaders, wheel-dozer, hauling

trucks). The main components of the pushbacks are ramps and benches. A bench can be defined as the set of blocks that shares a level inside of a pushback. The pushback should be fully connected by a ramp, and every bench of each pushback must have a minimum width defined by the equipment selected. Therefore, the size and amount of equipment need to be taken into account when pushbacks are designed. The ore tonnage inside pushbacks must be enough to sustain the plant capacity over a period, where typically 1 - 3 years time is taken.

The output of a successive calculation of inner pits is called the nested pits. Nested pits are used as guidelines to design pushbacks. A 'Pushback' represents a region that can be mined in a single continuous operation as defined within the Ultimate Pit. The basic objective is to create a pushback shape which attempts to meet defined primary targets, such as ore tonnage, stripping ratios or mined grade. However, because there is no explicit control in their shape and tonnage, nested pits cannot guarantee a minimum mining width. Therefore, the pushback design requires considerable intervention by mining engineers.

In open cut mining, pushbacks represent phases in open-cut mining, i.e. a series of pits satisfying production requirements and operational constraints for the mining operation. Much value can be gained or lost depending on how well the pushbacks are designed. In particular, due to high capital costs and fluctuations in input costs and commodity prices, mining is inherently risky. To overcome this, mining companies incorporate substantial discount factors into mine planning. Hence early production of high-grade ore is crucial to maximize Net Present Value (NPV) and minimize risk.

During the last 50 years, operations research techniques have been used in the industry to support the mine planning process. However, open pit optimisation models have been limited to a very small number of practical conditions. A challenge for open pit mine planning and design is finding a set of practical pushbacks, their extraction sequence and allocating the material to the appropriate destination.

In 1965, the IBM researchers Lerchs and Grossman (abbreviated LG) published an important graph theoretic method to compute the best value open pit from knowledge of the location of the ore zone. Jeff Whittle was the first person to code a fully 3-dimensional version of the LG algorithm in 1985 and his software became an industry standard. Since then, there have been many improvements and elaborations, but all methods produce approximately the same results. There are now more than ten major software supplier companies to the mining industry, which all have their own approaches to the design of open pit mines. Moreover, some of the big mining companies have their own proprietary software. As well as the design of the pit, scheduling, including logistics, management of stock piles and blending of ore from different mines, are all important features of modern mine planning software.

A key unsolved problem for open cut mines, except for the very smallest ones, is how to produce optimal pushbacks, or mining phases. A pit is developed over a period of anywhere from 2 to 30 or 40 years, depending on the size and depth of the deposit. Pushbacks are then a sequence of nested pits each representing around 2 years of production. For operational and geotechnical reasons, designing these pushbacks is a

crucial part of mine planning. If the life of the mine is more than 3-4 years, then pushbacks are required. The majority of open cut mines are of this type.

The LG approach is to (artificially) increase the cost of production, so that smaller and smaller pits are the outcome of their algorithm. This gives a formal way of producing nested pits – an inflated cost of production shrinks the pit in size. However, these nested pits suffer from a number of drawbacks.

- The pits may be too narrow in parts, hence operationally infeasible for mining equipment access
- The pits may be disconnected, which makes for very inefficient designs, due to the problem of equipment access via ramps.
- The size of the nested pits may vary widely, contrary to the requirement of steady production for the refining mill. Typically, each pit should represent about two years of production.

The current practice is that the output of the LG algorithm is used by mine planners as a starting point to produce a practical pushback design, satisfying all production and access requirements. Ramps are required to be added to the design at this stage. Ramps are roads that allow access to the working places. Their maximum slope is constrained by the haulage equipment. The aim of the ramp design is to minimise the construction and operational cost, which includes minimising the hauling distance and traffic congestion. The road design should be included as early as possible because there is a significant difference between a pushback design with and without ramps. To define the layout, engineers need to take into account strategic factors. Pit exit locations, pit geometry, the evolution of the pit shape and the influence of the topography are crucial to minimise the haulage cost. Roads can change the overall shape of the pit dramatically.

This conversion process from an initial semi practical pit design to a practical design is time consuming and value destroying. For example, typically a mining/consultant company might take 2 weeks or more to design pushbacks and ramps. Every mining company has one corporate planning group and each mine site has another planning group. Each group may do around 4 design scenarios a year. Moreover, changing optimised nested pits into practical pushbacks can change the NPV by a substantial amount. Thus, there are time/personnel costs and potentially significant NPV reduction costs associated with conversion to a practical design.

There are no accepted standards and an inexperienced mine planner might produce a design which is far from optimal for NPV. Similarly, there is no reproducibility of good design to provide benchmarking across mining company operations.

There is a need for improvements in mine planning.

## **Summary of the Invention**

According to one aspect there is provided a computer implemented method of mine design, the method comprising obtaining a three dimensional (3D) model for a mine, the 3D model characterizing physical material of the mine as a plurality of geometric elements, each geometric element representing a portion of the physical material extractable from the mine, applying a mathematical model, based on mathematical modelling of behavior of connected bubbles, to cluster geometric elements from the 3D model based on physical location and properties of the physical material for each of the geometric elements, and selecting at least one set of a plurality of contiguous geometric elements for extraction based on the clustering. Each set of a plurality of contiguous blocks can be associated with a phase of a mining process.

The mathematical model can include mathematical constraints reflecting operational constraints for the mine type.

In embodiments for an open cut mine the bubble model can be used for pushback design and operational constraints include connectivity, minimum bench width and appropriate angles between pushbacks.

In some embodiments for each pushback the mathematical model balances selection of contiguous geometric elements to minimise geometric surface area while maximizing economic value for each pushback.

In some embodiments, the mathematical model includes a geometric compactness tradeoff factor to enable an operator controllable weighting between maximizing economic value and pushback geometry to be defined and input to the mathematical model.

In some embodiments, the mathematical model comprises:

$$\max z = (1 - \lambda) \sum_{p \in P} W_p \sum_{b \in B} V_b \cdot x_b - \lambda \cdot \left( \sum_{p \in P} W_p \left( \sum_{b \in B} S_b \cdot x_{bp} - \sum_{\substack{b \in B \\ \hat{b} \in \hat{B}_b}} I_{b\hat{b}} \cdot y_{b\hat{b}p} \right) \right)$$

subject to:

$$\sum_{b \in B} A_p \cdot x_{bp} = \bar{A}_p \quad \forall p \in P \quad (6)$$

$$x_{bp} + x_{\hat{b}p} - y_{b\hat{b}p} \leq 1 \quad \forall b \in B, \forall \hat{b} \in \hat{B}_b, \forall p \in P \quad (7)$$

$$y_{b\hat{b}p} \leq x_{bp} \quad \forall b \in B, \forall \hat{b} \in \hat{B}_b, \forall p \in P \quad (8)$$

$$y_{b\hat{b}p} \leq x_{\hat{b}p} \quad \forall b \in B, \forall \hat{b} \in \hat{B}_b, \forall p \in P \quad (9)$$

$$\sum_{\rho \leq p} x_{b\rho} \leq \sum_{\rho \leq p} x_{\bar{b}\rho} \quad \forall b \in B, \forall \bar{b} \in \bar{B}_b, \forall p \in P \quad (10)$$

$$\sum_{p \in P} x_{bp} \leq 1 \quad \forall b \in B \quad (11)$$

$$(x, y) \in \Omega \quad (\text{other constraints}) \quad (12)$$

5 Where:

*Sets and notation:*

$b \in B$  : Set of blocks  $\{0, 1, \dots, B\}$ .

$\hat{b} \in \hat{B}_b$  : Set of blocks that are adjacent to block  $b$ .

$p \in P$  : Set of pushbacks (clusters)  $\{0, 1, \dots, P\}$ .

10  $\bar{b} \in \bar{B}_b$  : Set of slope precedences for block  $b$ .

*Parameters:*

$S_b$ : external surface area of block  $b$

$I_{b\hat{b}}$ : intersecting surface area between block  $b$  and  $\hat{b}$

$A_b$ : amount of attribute A associated to block  $b$

15  $\bar{A}_p$ : total amount of attribute A required in pushback  $p$

$W_p$ : weight for the pushback  $p$

$V_b$  : economic value of block  $b$

$\lambda$  : compactness factor. Weight (ranging from 0 to 1) to balance the importance of the two objectives in the optimisation problem

20 *Variables:*

$x_{bp}$ : binary, equal to one if the block  $b$  is assigned to pushback  $p$ , zero otherwise

$y_{b\hat{b}p}$ : binary, equal to one if blocks  $b$  and  $\hat{b}$  belongs to pushback  $p$ , zero otherwise.

25 Some embodiments of the method further comprise the steps of adjusting the compactness factor and producing a further pushback design. For underground applications the compactness factor can also be adjusted to produce further excavation plans. In some embodiments the compactness factor is adjusted incrementally to produce a plurality of pushback designs or excavation plans.

30 According to another aspect, there is provided a method of designing ramps for a set of pushbacks based on a linear programming formulation to find a minimum cost ramp with vertical and horizontal alignment constraints for a given pit and a given ramp width, taking into account stripping associated with ramp excavation, by applying a geological model which is assumed to be represented as a regularly spaced set of blocks  
35 including the topography of the pit, and ramp width equivalent to the block size in  $x$  or  $y$  and the dimension in  $z$  represents the maximum ramp slope ( $\max. \text{slope} \leq \frac{z}{x} \text{ or } \frac{z}{y}$ ).

In an embodiment the ramp design is calculated in accordance with:

40 Sets:

$N$ : Set of blocks in the block model.

$A_i^+$ : Set of blocks  $j$  such that there is an arc from  $j$  to  $i$ .

$A_i^-$ : Set of blocks  $i$  such that there is an arc from  $i$  to  $j$ .

45  $\Gamma_i^+$ : Set of vertical precedences of block  $i$  (upward).

$\Gamma_i^-$ : Set of vertical precedences of block  $i$  (downward).

$R_i$ : Set of all possible ramp directions from block  $i$ .

$I$ : Set of possible ramp starting blocks.

$E$ : Set of possible ramp ending blocks.

5 Parameters:

$c_i$ : extraction cost of block  $i$ .

$p_i^{h,j}$ : change of direction cost in ramp segment  $i$ , where  $h$  is the incoming direction and  $j$  is the outgoing direction at  $i$  with  $h, j \in R_i$  and  $h \neq j$ .

10 Variables:

$r_i$ : binary, equal to 1 if the block  $i$  is selected as a ramp, 0 otherwise.

$x_i$ : binary, equal to 1 if the block  $i$  is extracted, 0 otherwise.

$v_i^{h,j}$ : binary, equal to 1 if the ramp changes direction from  $h$  to  $j$  (or from  $j$  to  $h$ )

15 at block  $i$ , 0 otherwise ( $h, j \in R_i$  and  $h \neq j$ ).

$a_{i,j}$ : binary, for each arc from a block  $i$  to a block  $j$  equal to 1 if  $i$  and  $j$  are selected as ramp blocks, 0 otherwise ( $i \neq j$ ).

Objective function:

$$\min_{x,v} \left\{ \sum_{i \in N} c_i \cdot x_i + \sum_{\substack{i \in N \\ h, j \in R_i \\ h \neq j}} p_i^{h,j} \cdot v_i^{h,j} \right\} \quad (1)$$

20

Subject to:

$$x_i \leq x_j \quad \forall i \in N, \forall j \in \Gamma_i^+ \quad (2)$$

$$r_i \leq x_i \quad \forall i \in N \quad (3)$$

$$r_i \leq 1 - x_j \quad \forall i \in N, \forall j \in \Gamma_i^- \quad (4)$$

$$a_{i,j} \leq r_i \quad \forall i \in N, \forall j \in R_i \quad (5)$$

$$a_{i,j} \leq r_j \quad \forall i \in N, \forall j \in R_i \quad (6)$$

$$r_i + r_j \leq 1 + a_{i,j} \quad \forall i \in N, \forall j \in R_i \quad (7)$$

$$\sum_{j \in I} a_{s,j} = 1 \quad (8)$$

$$\sum_{i \in E} a_{i,t} = 1 \quad (9)$$

$$\sum_{i \in I} a_{i,s} = 0 \quad (10)$$

$$\sum_{j \in E} a_{i,j} = 0 \quad (11)$$

$$\sum_{j \in A^-} a_{i,j} = \sum_{i \in A^+} a_{j,i} \quad \forall i \in N \quad (12)$$

$$a_{h,i} + a_{i,j} + a_{j,i} + a_{i,h} \leq v_i^{h,j} + 1 \quad \forall i \in N, \forall h, j \in R_i, h \neq j \quad (13)$$

$$v_i^{h,j} \leq a_{h,i} + a_{i,j} \quad \forall i \in N, \forall h, j \in R_i, h \neq j \quad (14)$$

$$v_i^{h,j} \leq a_{j,i} + a_{i,h} \quad \forall i \in N, \forall h, j \in R_i, h \neq j \quad (15)$$

$$a_{i,j} + a_{j,i} \leq 1 \quad \forall i, j \in N \quad (16)$$

$$x_i, r_i, a_{i,j}, v_i^{h,j} \in \{0, 1\}, \quad \forall i, j \in N \quad (17)$$

According to another aspect there is provided a computer implemented mine planning method comprising the steps of:

- 5 obtaining a mine block model defining three dimensional geological resources of a mine as a regular grid of a plurality prismatic blocks, wherein each block represents a defined volume of material, and wherein block characterization data for each block includes at least coordinate data, physical characteristic data and assigned economic value data;
- 10 defining a target pushback tonnage for each pushback; and
- determining one or more pushbacks for the mine using the block characterization data, the target pushback tonnage, a plurality of contiguous blocks to comprise the pushback by applying a mathematical model for operational constraints of connectivity, minimum bench width and appropriate angles between pushbacks to select from the mine block model, based on physical location and economic value estimates for each
- 15 block, a plurality of contiguous blocks having a combined weight equal to or less than the target pushback tonnage to comprise each pushback.

In an embodiment, the mathematical model is based on mathematical modelling of behavior of connected bubbles, and for each pushback the mathematical model balances

selection of contiguous blocks to minimise geometric surface area while maximizing economic value for each pushback.

5 In some embodiments, the mathematical model includes a geometric compactness tradeoff factor to enable an operator controllable weighting between maximizing economic value and pushback geometry to be defined and input to the mathematical model.

In some embodiments the mathematical model comprises:

$$10 \quad \max_{x,y} \left\{ \sum_{i \in N} v_i \cdot x_i - c \cdot \left( \sum_{i \in N} 6 \cdot l_i^2 \cdot x_i - \sum_{\substack{i \in N \\ j \in \Omega_i}} l_i^2 \cdot y_{i,j} \right) \right\}$$

subject to:

$$\sum_{i \in N} w_i \cdot x_i - W \leq 0 \quad (4)$$

$$x_i - x_j \leq 0 \quad \forall i \in N, \forall j \in \Gamma_i^+ \quad (5)$$

$$15 \quad x_i + x_j - y_{i,j} \leq 1 \quad \forall i \in N, \forall j \in \Omega_i \quad (6)$$

$$x_i, y_{i,j} \in \{0,1\} \quad \forall i, j \in N \quad (7)$$

Where:

- $N$  : Set of blocks in the block model
  - 20 •  $\Omega_i$  : Set of adjacent blocks to block  $i$
  - $\Gamma_i^+$  : Set of vertical precedences of block  $i$  (upwards)
  - $c$  : compactness factor
  - $v_i$  : economic value of block  $i$
  - $l_i$  : length of the block  $i$
  - 25 •  $w_i$  : tonnage of the block  $i$
  - $W$  : total tonnage of the bubble pit
  - $x_i$  : binary, equal to one if the block  $i$  is extracted, zero otherwise
  - $y_{i,j}$  : binary, equal to one if blocks  $i$  and  $j$  are extracted, zero otherwise.
- 30 Some embodiments of the method include further comprising the steps of adjusting the compactness factor and producing a further pushback design. In some embodiments the compactness factor is adjusted incrementally to produce a plurality of pushback designs and further comprising the step of outputting one or more of the plurality of pushback designs.
- 35 Some embodiments of the method further comprise the step of performing ramp design for one or more pushback designs. For example, in some embodiment the ramp design is performed in accordance with the method as described above.
- 40 According to another aspect there is provided a mine planning system implemented using computer processing and memory resources, the system comprising at least one module configured to implement the methods as described above.

## **Brief Description of the Drawings**

Figure 1a is a flowchart illustrating the current mine planning process;

- 5 Figure 1b is a flowchart showing the change of the mine planning process using an embodiment of the pushback design method the invention.

Figure 2 illustrates the main outputs of stages of the open pit mine planning process.

- 10 Figure 3 illustrates a flowchart of a bubble pit optimization process.

Figure 4 illustrates bubble pit graphs showing effect of compactness factor.

- 15 Figure 5 shows the variation of the NPV versus the compactness factor for a prototype test of the pushback design algorithm compared with an engineer prepared design using the same mine data.

- 20 Figure 6 illustrates impact of compactness factor (before introducing unit transformation factor, so ranging from 0 to 200) in the output of the bubble pit model (top view).

Figure 7 illustrates an example of drawpoint sequencing (Mine sequence optimization for Block caving using concept of 'Best and worst case.

- 25 Figure 8 illustrates a caving propagation slope.

Figures 9a and 9b illustrate scheduled block removal times with two different compactness values.

- 30 Figure 10 illustrates open pit ramps.

Figure 11 is a block diagram of an embodiment of a system configured to implement the mine planning method.

- 35 Figure 12 is a high level flowchart of the bubble model method.

## **Detailed Description**

- 40 Disclosed are methods of mine design using a mathematical model based on mathematical modelling of connected bubbles. The disclosed methods can be used for designing pushbacks for open cut mines, or for planning caving or stoping for underground mining. The bubble model can include mathematical constraints reflecting operational constraints for the different mine types.

- 45 A high level example of the method is shown in Figure 12, the process starts by obtaining a three-dimensional (3D) model for a mine 1210, for example by retrieving from a database, system memory or storage media, the 3D model may also be input to the system, for example via email or file transfer. The 3D model characterizes physical material of the mine as a plurality of geometric elements, each geometric element representing a portion of the physical material extractable from the mine. Such models

are often referred to as block models. A system processor, appropriately programmed with a mathematical model based on modelling of connected bubbles, applies the mathematical model 1220 to cluster geometric elements from the three-dimensional (3D) model based on physical location and properties of the physical material for each of the geometric elements – this data being extracted from the 3D model. Based on this clustering the processor then selects at least one set of a plurality of contiguous geometric elements for extraction 1230. The properties of the physical material can include any data captured in the 3D model, including economic value for each element.

In the example of an open cut pushback design, the bubble model can include mathematical constraints reflecting the operational constraints of connectivity, minimum bench width and appropriate angles between pushbacks. This model is applied in conjunction with economic data for pushback design. For underground mining the bubble model can also be used.

The present method enables mathematical model based mine planning, which simultaneously satisfy operational constraints and maximize NPV. This could be used by large mining companies with in-house design capability, mining consulting companies and mining software providers.

Embodiments may also be utilised in scheduling aspects of mine planning, for example for planning temporal as well as physical aspects of extraction. It should be appreciated that planning includes extraction of both valuable target material (ore or “pay dirt” carrying the target minerals) and waste material, often referred to as “overburden”, which must be extracted to get to the valuable material. Other required infrastructure such as accessways, roads and or ramps also form part of a comprehensive mine plan and are encompassed within the scope of the present invention.

An input to the mine planning method, as is common to most mine planning, is a 3-dimensional model of the mine characterizing the target material as well as surrounding rock and dirt. The 3D model characterizes physical material of the mine as a plurality of geometric elements, each geometric element representing a portion of the physical material extractable from the mine. The 3D model may be based on core samples and/or seismic data for the mine area. Such 3D models are often referred to as block models.

A 3-dimensional volume is the input to the invention, where the volume is divided into geometrical elements, i.e. small pieces, often called blocks, but could be panels or other larger or irregular sub-regions. The composition of the geometrical elements is known and classified into valuable material, waste and contaminants. The problem is to decide which elements to extract and in what sequence. Constraints include equipment requirements, access, geotechnical stability and others.

The aim of mine planning is to balance maximizing value extracted from the mine with the practical limitations and costs associated with the extraction. In principle, the aim is to extract the maximum volume of valuable material while minimizing total volume extracted. Embodiments of the disclosed methods apply to this optimisation problem; a mathematical model based on mathematical modelling of connected bubbles. The bubble clustering method consists of generating a cluster of elements such that the

external surface area of the cluster is minimized; this is done to generate compact and most likely connected clusters. Our clustering method requires that the surface area of each element as well as the surface area of the combination of elements can be computed or estimated.

For simplicity, in the examples given below, the bubble cluster method is applied to a block model, a regular-spaced grid of blocks. A 3D block is typically a prismatic shape that is represented by the coordinates of its centroid (x,y,z) which contains the main information for the planning stage. Therefore, blocks are the basic unit (or elements) that can be part of the clusters. It is worth noting that the bubble cluster are not limited to block models, and the clustering method applies to any framework in which the area of the elements can be minimised. The bubble cluster model has several advantages when applied in the context of mine planning.

Embodiments of the disclosed mine planning methods operate to minimise the boundary surface area of the set of elements chosen to be extracted in a given period of time or the set of elements that satisfies a given characteristic, as a key part of the optimisation process. Such a minimisation produces clusters of elements. There are several advantages of using the disclosed bubble model method for clustering of the chosen elements.

1. If a cluster of elements of a given volume minimises boundary surface area, then the cluster behaves mathematically like a bubble and so will be similar to a round sphere enclosing a given volume. The restriction on size of the cluster corresponds to a mining constraint such as the volume required to be extracted in a given time period. This bubble behavior gives the clusters geometrical shapes with desirable properties, for example, when designing pushbacks or cutbacks, which are a series of nested pits forming an open pit mine. Each such nested pit represents a fixed amount of volume for production, typically around 2-4 years of feed to the processing plant.
2. When scheduling extraction of the chosen elements, minimising boundary surface area of clusters in a similar time period produces regions which are practical for extraction either in surface mining or underground mining. For the efficient use of mining equipment and personnel, the chosen elements should be relatively close together during each time period of production and bubble clusters satisfy this criterion.
3. If blending is required, such as having a minimum ore grade, maximum contaminant, similar rock hardness and geochemical qualities for crushing and refining in a given time period, then using bubble clusters can also produce schedules for efficient extraction. Optimisation then gives a bonus for choosing elements with similar characteristics in a given time period and either a bonus for maximal overlap between the surface areas of the chosen elements or a penalty for the total surface area of the cluster of elements chosen.
4. In underground mining, propagation of rock stress must be controlled using a suitable extraction sequence. Bubble clusters will tend to reduce stress as the clusters will not have 'sharp corners' where stress concentrates. This property can be used to optimise the extraction sequence of caving mines as well as defining the mining boundary and sequence of selective mining methods such as

stopping mines or any other method that might be benefited by using bubble-like clusters.

5. Dump management and stockpiling sequencing consist of moving material from the mine to external facilities (or internal in the case of backfilling) permanently or temporarily. Similarly to the mining process, the dumping and re-handling operations require connected and wide areas to ensure the safe operation of the equipment. The bubble model can also be applied in this context.

Additionally, blending and volume constraints might be required in the sequencing problem for dumps and stockpiles. By subdividing the volume of dumps and stockpiles into geometrical elements and minimising the external surface area of the cluster of elements handled at a similar period, sequences can be generated which are efficient for operations.

### **Mathematical Formulation**

To generate compact clusters, the following model is based on the bubble principle which is to minimise the surface area for a given volume in a non-homogeneous distribution field. This problem can be formulated as a linear model as follows.

#### **Bubble cluster**

To introduce the concept of the bubble clusters let us consider a simple version of the problem, where we want to minimise the weighted external surface area of a set of clusters of geometrical elements, such that each cluster contains a given amount of an attribute  $A$ . This is a very useful model as the bubble analogy works precisely and there is a lot known about the structure of clusters of planar bubbles.

Let us consider a set of geometrical elements  $e \in \mathbf{E}$ , such that the surface of each element  $S_e$  is known and the surface area of any possible clusters can be computed. Despite the fact that there are many ways of computing areas of geometrical surfaces, we will consider that the surface area of a cluster is the sum of the external surface areas of all elements minus the intersecting surface areas of adjacent elements. The latter consideration implicitly assumes that we know (or we can compute) the intersecting surface areas of adjacent elements. Note that our approach is not limited to this method for computing surface areas.

Formally, the simple version of the bubble cluster can be illustrated by the following linear programming formulation.

#### **Sets and notation:**

- $e \in \mathbf{E}$  : Set of elements  $\{0, 1, \dots, E\}$ .
- $\hat{e} \in \widehat{\mathbf{E}}_e$  : Set of elements  $\hat{e}$ , which are adjacent to element  $e$ .
- $c \in \mathbf{C}$  : Set of clusters  $\{0, 1, \dots, C\}$ .

#### **Parameters:**

- $S_e$ : external surface area of element  $e$ .
- $I_{e\hat{e}}$ : intersecting surface area between elements  $e$  and  $\hat{e}$ .
- $A_e$ : amount of attribute  $A$  associated to element  $e$ .
- $\bar{A}_c$ : total amount of attribute  $A$  required in cluster  $c$ .
- $W_c$ : weight for the cluster  $c$ .

Variables:

- $x_{ec}$ : binary, equal to one if the element  $e$  is assigned to cluster  $c$ , zero otherwise.
- 5 •  $y_{e\hat{e}c}$ : binary, equal to one if elements  $e$  and  $\hat{e}$  are assigned to cluster  $c$ , zero otherwise.

Objective function 1:

$$\min z = \sum_{c \in \mathcal{C}} W_c \left( \sum_{e \in \mathcal{E}} S_e \cdot x_{ec} - \sum_{\substack{e \in \mathcal{E} \\ \hat{e} \in \hat{\mathcal{E}}_e}} I_{e\hat{e}} \cdot y_{e\hat{e}c} \right)$$

10 subject to:

$$\sum_{e \in \mathcal{E}} A_e \cdot x_{ec} = \bar{A}_c \quad \forall c \in \mathcal{C} \quad (1)$$

$$x_{ec} + x_{\hat{e}c} - y_{e\hat{e}c} \leq 1 \quad \forall e \in \mathcal{E}, \forall \hat{e} \in \hat{\mathcal{E}}_e, \forall c \in \mathcal{C} \quad (2)$$

$$y_{e\hat{e}c} \leq x_{ec} \quad \forall e \in \mathcal{E}, \forall \hat{e} \in \hat{\mathcal{E}}_e, \forall c \in \mathcal{C} \quad (3)$$

15

$$y_{e\hat{e}c} \leq x_{\hat{e}c} \quad \forall e \in \mathcal{E}, \forall \hat{e} \in \hat{\mathcal{E}}_e, \forall c \in \mathcal{C} \quad (4)$$

$$\sum_{c \in \mathcal{C}} x_{ec} \leq 1 \quad \forall e \in \mathcal{E} \quad (5)$$

20 *Explanation of objective function and constraints:*

The objective function minimises the weighted surface area of the clusters. Constraint (1) ensures that cluster satisfies the total amount of attribute A required, this type of constraint can be generalised for all attributes desired within the clusters. Constraints (2)-(4) identify which adjacent elements belong to the same cluster. Finally, constraint (5) ensures that the elements can only belong to one cluster.

*Open cut pushbacks*

30 Open pit pushbacks are “clusters” of blocks that are: 1) connected, 2) satisfy minimum mining width (due to the mining equipment required for the extraction process), 3) the horizontal angle formed by the intersection of two pushbacks is not too small.

Each pushback may have to contain a given amount of different materials such as ore, contaminants, waste. The total volume within the pushbacks is typically between 2-4 years of production. In most cases, traditional methods to define pushbacks (like the Lerchs and Grossman parametrization method) gives impractical pushbacks, i.e. do not satisfy all (or even any) of the three conditions already mentioned. The Bubble cluster method can be used to model the shape constraints of the pushback problem. The following formulation shows a simplified example of the pushback problem, in which the minimisation of the surface area of the bubble clusters corresponds to the second term of the objective function.

A key feature of the disclosed method is to use a proxy, in the form of a mathematical model based on the behavior of soap bubbles, for the operational constraints of connectivity, minimum bench width and appropriate angles between pushbacks. For efficient mining production, each pushback should be a connected region, i.e. not have separated parts. The bench width is set by the requirements of mining equipment for ore production. Narrow angles between pushbacks give regions which are hard to access – having a sufficiently large angle eliminates this problem.

In embodiments of the disclosed method and system, a sequence of pushbacks is modeled as a cluster of soap bubbles. Surface tension of soap bubbles drives them to have the smallest area enclosing a given volume. In the context of mining, the bubble model defines the geometry of a pushback for mining the maximum value material with a minimal geometric surface area.

Using a mathematical bubble model approach gives rise to processes which utilise features inherent to the bubble model to design pushbacks which better address requirements of connectivity, bench width and angles. These requirements are not treated successfully in current mining planning software.

Important mathematical properties of soap bubbles were proved by Jean Taylor in 1976 – three soap bubbles meet at an angle of 120 degrees and four soap bubbles meet at a corner at 109.47 degrees. This explains why using a soap bubble model allows us to construct pushbacks with large angles, where the pushbacks meet.

Since soap bubbles have smallest area enclosing a given volume, such a volume will be associated with a single soap bubble in a cluster. For otherwise one can slide two pieces of bubbles enclosing parts of a given volume together to merge to form a larger bubble, enclosing the same volume but with less area. Further, since soap bubbles have constant mean curvature (this is a measure of 'bending' of the soap bubble surface) and form shapes like spheres, the minimal bench width property applies, hence giving large bench width.

These geometric features of a soap bubble model inherently address the requirements for pushback design of connectivity, minimum bench width and appropriate angles. Thus, enabling the mathematical bubble model to be applied, gives a proxy for these constraints and provides a geometric framework for the economic analysis involved in pushback design.

Further in some embodiments the practical application of the mathematical model in a mine planning algorithm incorporates a compactness factor, which can be a variable adjusted by the mine planner. The compactness factor is a measure of the dispersion of the blocks that compose the pushback. Or considered another way, it is a mechanism to mathematically enable relaxation of the bubble model geometry from a spherical approximation based on economic aspects. This is explained further in the paragraphs below.

In the bubble model approach, there are two potentially competing objectives – maximizing NPV by taking high value blocks early and minimizing the area of the

boundary surface of each pushback, while enclosing a given volume. As discussed above, minimizing surface area to enclose a given volume is an inherent property of a bubble, and so the mathematical model of soap bubble geometry can address the physical objective of minimizing the boundary surface for mine pushbacks. But a purely  
5 geometric bubble model approach does not take into consideration the varying economic value of the material to be mined. Thus, the mathematical model of a soap bubble needs to be altered to allow relaxing of the bubble geometry to take into consideration the objective to preferentially mine the more valuable regions and improve NPV. For example, it can be desirable to allow the geometry of the bubble to  
10 "stretch" from a perfect sphere to enable encompassing of higher value material. In the mathematical model described herein, a compactness factor is incorporated into the mathematical bubble model to accommodate economic factor based modification of the bubble geometry.

15 In some embodiments of the current method, the objectives of maximizing NPV by taking high value blocks early and minimizing the area of the boundary surface of each pushback, while enclosing a given volume are combined by taking a weighted sum of the objectives. Hence a percentage of the objective is maximizing NPV and the complementary percentage is for the objective of minimizing area. If we start with 0%  
20 contribution of minimizing area and 100% maximizing NPV, the result will be very similar to classical Lerchs Grossman nested pits, with capacity constraints of the bubble model having the only influence on the resulting pushback design. The geometric constraints of the bubble model are negated by the 100% weighting on NPV. At the other end of the spectrum, 100% area and 0% NPV will give perfect soap bubble  
25 pushbacks, but will not necessarily give a good value for NPV. It is anticipated that for most mines there will be at least one % compromise where the trade-off between maximizing NPV and minimizing area will provide superior workable pushback designs. It is envisaged that a system embodying the mathematical bubble model method may be used to automatically generate multiple workable pushback designs, each for a different  
30 % compromise, so the mine planner may choose a preferred design.

The mine designer can adjust the degree of compactness (i.e. relative percentages of area and NPV) to achieve an appropriate design. In this process, the designer is able to understand the tradeoff between the operational constraints and the current  
35 methodology of nested pits, which optimises NPV but does not give practical designs. In particular, this means that for a given mine and ore body, the designer is able to experiment by running multiple designs with different compactness factors to see which one gives the largest NPV but still satisfies the required operational constraints.

40 Note that area and NPV have different units and are very different in magnitude. To overcome these issues, one disclosed method uses a unit transformation factor, denoted  $f$ , which transforms between area of a pushback and dollars of NPV. This enables a compactness factor varying between 0 and 100% to be used.

45 In another embodiment there is no unit transformation factor needed to compare units of area with units of dollars in NPV so the two terms can be combined. This is on a scale from around 0 to 1000 weighting of the area. Weighting of the area is used since the area is much smaller than the number of dollars in the units typically used. The

weighting combines the NPV and area tradeoff into a single value which can be applied to all blocks.

A unit transformation factor can be introduced to convert units of area to units of dollars, then the compactness factor can be thought of as a percentage, as discussed above, saying then we have a certain percent NPV and the complementary percent area.

Underlying this approach is a new algorithm which converts the economic values of the mining blocks into pushbacks, so that there is a balance between the operational constraints and mining the highest value blocks as soon as possible.

Figures 1a and 1b show an example of current mine planning processes in Figure 1a compared with an alternative mine planning process using embodiments of the disclosed pushback planning method in Figure 1b. Figure 2 illustrates the outputs of different phases of mine design process.

Figure 1a represents the current typical planning procedure. This starts with the input data 102 including geological resource data (for example, a three-dimensional block model 210 produced from exploration which characterizes the type and location of the geological resources illustrated in Figure 2), precedence constraints and economical models. This data is used to define the ultimate pit limit 108 and geometric ultimate pit shell 222 which defines the 3D boundary of the mine (i.e. shape of the cone) for extraction of the valuable material. This ultimate pit shell 108 is input to a process for designing nested pits 110 using conventional methods. Also input 112 to the nested pit design 110 are geological resource data (block model), precedence constraints, economical models, and revenue factors.

The output from the nested pit design 110 is an initial design of pit shells 115, 225, which is input to a process for designing semi-practical pushbacks 120. Also input to the semi-practical pushback design process is geological resource data (including the block model 210), mining width constraints, and pushback target parameters (for example the tonnage for removal for each pushback). The output 122 from the semi practical pushback design is a set of semi-practical pushbacks 230 and pushback precedence (order) constraints. This semi-practical pushback design 122 is input to a scheduling planning process 125 along with scheduling parameters 128 to determine whether or not the design 122 is feasible 130 for practical implementation. If the design is not feasible 130 then the pushback design 120 and scheduling 125 processing is repeated until a feasible solution is found.

Once a feasible 130 semi-practical pushback design 230 is found, the next step is to convert this to a practical pushback design 140 by adding ramps, based on ramp parameters (for example minimum width and turning angles based on equipment to be used and traffic) to provide a practical pushback design and precedence constraints 148. This is then input to a second scheduling process 150 using scheduling parameters 152 to produce a mine schedule 240 and dynamic cut-off policy 155. This is assessed for feasibility 160 and repeated 165, if necessary, and the design planning process ends 170 once a feasible design and schedule is found. It should be appreciated that the mine designing process is iterative and can be highly manual, taking significant time and resources to produce a feasible design. This entire process may also be repeated to

generate multiple feasible plans which may be compared by a mine designer and a preferred one of these feasible designs chosen for implementation.

Embodiments of the disclosed pushback design method can replace the steps of the design process for designing nested pits 110 and designing semi-practical pushbacks 122 (as shown in Figure 1a), with a single process 180 for designing semi-practical pushbacks 188 based on a bubble model (referred to as bubble pits) as shown in Figure 1b. Inputs to the bubble pit process 180 include the ultimate pit shell 108, geological resource data (block model), precedence constraints, and economical models 182, similarly to nested pit designs. Also input to the bubble pit processing is shape factor data (mining width constraints and constraints based on 3D cone models) and pushback targets 185. The output from the bubble pit modelling is a semi-practical pushback design and precedence constraints 188 which can be subject to scheduling 190 based on scheduling parameters 192 to assess the feasibility 194 of the design. And the process is repeated 195 if necessary to derive a feasible design or multiple feasible pushback designs for comparison.

The steps for conversion of a semi-practical pushback design to a practical pushback design can be the same as for traditional mine planning processes. An embodiment also provides a ramp design method which may also be used with traditional pushback design methods.

The pushback design method 180 provides a method to design pushbacks for open cut mines, which simultaneously satisfies operational constraints and maximizes NPV, the simultaneous satisfaction of these objectives is not known to be possible with any other mine planning method. Embodiments could be used by large mining companies with in-house design capability, mining consulting companies and mining software providers.

Embodiments fit in the core of the open pit mine planning value chain and allow designers to both optimise and automate the pushback design stage.

The inventors believe their algorithm can solve the optimal pushback design problem. The disclosed method produces designs which can automatically satisfy connectivity and minimum mining width, and optimises NPV with these additional constraints. Embodiments also include a weighting factor which can be selected by a mine planner to adjust trade-offs between optimal NPV and operational constraints, allowing these to be fully explored.

Most mining researchers do not believe that such a solution is possible. An explanation of this is that operational constraints are inherently geometric, so are difficult to capture by standard methods of optimisation (linear and integer programming) used in mining.

The inventor's breakthrough is to use a physical model which forces the operational constraints indirectly but has the correct properties to achieve large scale optimisation, involving huge numbers of variables and constraints. The mathematical characterization of the physical model is used to enable a computer implemented process to automatically or semi-automatically produce pushback designs. This can significantly decrease the time and manual effort associated with mine planning.

One can use the bubble clusters methodology to introduce the connectivity and width constraints to the pushback design problem.

- 5 Let us consider a simple instance of the bubble pit problem defined as: find the minimum surface area pit (inverted cone shape) that maximises the economic profit of the total extracted blocks, such that each pit must have a given amount of attribute  $A$ . Note that other constraints can be added to represent specific conditions of different mining requirements, but we show this simple problem to illustrate how the bubble clusters can be used in the pushback design problem.

15 Let us assume a 3D block model, a regularly spaced 3D grid for which every element of the grid is a block (a 3D prism), for this formulation. Our bubble pits (or pushbacks) will be the result of merging a set of prismatic elements, which are the blocks. Note that the slope constraints are superimposed on the problem so that the spherical shapes of bubbles are replaced by cones on planar bubble clusters. These cones have least surface area enclosing a given volume amongst shapes satisfying the slope bounds. Since the problem has multiple objectives, we introduce a factor  $\lambda$  to weight both objectives. However, our method does not rely on any particular approach to solve multiple-objective optimisation problems.

*Sets and notation:*

- $b \in \mathbf{B}$  : Set of blocks  $\{0,1,...,B\}$ .
- $\hat{b} \in \widehat{\mathbf{B}}_b$  : Set of blocks that are adjacent to block  $b$ .
- 25 •  $p \in \mathbf{P}$  : Set of pushbacks (clusters)  $\{0,1,...,P\}$ .
- $\bar{b} \in \bar{\mathbf{B}}_b$  : Set of slope precedences for block  $b$ .

*Parameters:*

- $S_b$ : external surface area of block  $b$ .
- $I_{b\hat{b}}$ : intersecting surface area between block  $b$  and  $\hat{b}$ .
- 30 •  $A_b$ : amount of attribute  $A$  associated to block  $b$ .
- $\bar{A}_p$ : total amount of attribute  $A$  required in pushback  $p$ .
- $W_p$ : weight for the pushback  $p$ .
- $V_b$  : economic value of block  $b$ .
- $\lambda$  : *compactness factor*. Weight (ranging from 0 to 1) to balance the importance of the two objectives in the optimisation problem.

*Variables:*

- 40 •  $x_{bp}$ : binary, equal to one if the block  $b$  is assigned to pushback  $p$ , zero otherwise.
- $y_{b\hat{b}p}$ : binary, equal to one if blocks  $b$  and  $\hat{b}$  belongs to pushback  $p$ , zero otherwise.

*Objective function 2:*

$$\max z = (1 - \lambda) \sum_{p \in \mathbf{P}} W_p \sum_{b \in \mathbf{B}} V_b \cdot x_b - \lambda \cdot \left( \sum_{p \in \mathbf{P}} W_p \left( \sum_{b \in \mathbf{B}} S_b \cdot x_{bp} - \sum_{\substack{\hat{b} \in \widehat{\mathbf{B}}_b \\ b \in \mathbf{B}}} I_{b\hat{b}} \cdot y_{b\hat{b}p} \right) \right)$$

45

subject to:

$$\sum_{b \in B} A_p \cdot x_{bp} = \bar{A}_p \quad \forall p \in P \quad (6)$$

$$x_{bp} + x_{\hat{b}p} - y_{b\hat{b}p} \leq 1 \quad \forall b \in B, \forall \hat{b} \in \hat{B}_b, \forall p \in P \quad (7)$$

$$y_{b\hat{b}p} \leq x_{bp} \quad \forall b \in B, \forall \hat{b} \in \hat{B}_b, \forall p \in P \quad (8)$$

$$y_{b\hat{b}p} \leq x_{\hat{b}p} \quad \forall b \in B, \forall \hat{b} \in \hat{B}_b, \forall p \in P \quad (9)$$

$$\sum_{\rho \leq p} x_{b\rho} \leq \sum_{\rho \leq p} x_{\bar{b}\rho} \quad \forall b \in B, \forall \bar{b} \in \bar{B}_b, \forall p \in P \quad (10)$$

$$\sum_{p \in P} x_{bp} \leq 1 \quad \forall b \in B \quad (11)$$

$$(x, y) \in \Omega \quad (\text{other constraints}) \quad (12)$$

*Explanation of objective function and constraints:*

The multi-objective function maximises the weighted value of the extracted blocks and minimises the area of the bubble pit, combined by a weighting given by the compactness factor  $\lambda$ . Constraint (6) ensures that the pits contain a given amount of attribute A. More than one attribute can be considered in a similar manner, and a tolerance can be added to the mass constraint by replacing the equality constraint with two inequality constraints. Constraints (7)-(9) identify which adjacent elements belong to the same pushback. Constraint (10) represents the vertical precedence (slope wall constraint). Constraint (11) ensures that the blocks can only belong to one pushback. Finally, Constraint (12) might represent other operational restrictions (e.g., block-level or bench-pushback scheduling, mining and processing bounds, variable cut-off grade, blending constraints, and/or the maximum number of pushback opened), and could involve bounds on  $x$  and  $y$ .

The process of transforming the block model to a valid set of bubble pits is as follow. First, compute the set of parameters of the formulation ( $B, \hat{B}_b, P, \bar{B}_b, S_b, I_{b\hat{b}}, A_b, \bar{A}_p, W_p, V_b$ ). Second, generate a set of problem instances (for the desired values of compactness factor  $\lambda$ ). Third, perform the optimisation of the mathematical model for the set of problem instances. Finally, transform the result from the mathematical model to a block model file.

The process of transforming the block model to a valid set of bubble pits is illustrated in the flowchart of Figure 3. The inputs to the process are:

1. A mine block model
2. Economic parameters – such as the current or predicted ore prices and excavation costs

3. Geotechnical constraints – characterizing requirements for practical application such as not being able to extract a block without having suitable blocks above extracted first. Such constraints are known in the mining industry.

- 5 In a first step 320 sets of parameters required for the bubble pit formulation are determined; these parameters include but are not limited to the target weight (tonnage) or volume for each pushback, and the compactness factor.

10 Based on the input data 310 and determined parameter sets 320 a set of pushback instances are generated using the 3D mathematical bubble model, for example as defined in objective function 2. The set of pushbacks for the ultimate pit is generated by performing the bubble model optimisation for one pushback 340, removing the set of blocks comprising the pushback from the block model 350, and checking whether or not the block model is empty 355. If the block model is not empty the optimisation function will be applied to select blocks for the next pushback. This process is repeated until the block model is empty – all blocks of the block model have been selected and assigned to a pushback. It should be appreciated that the output of the mathematical modelling is a plurality of data sets, one for each pushback with each set defining the blocks selected to comprise the pushback.

20 Once all blocks have been assigned to a pushback by this modelling process, the data sets defining each pushback can be transformed 360 to a block model format for display and input to further steps of the mine planning process. The transformation to a block model format may include a step of adding to the block characterisation, data for each block within the block model data defining the pushback for which the block has been selected. The steps of scheduling and feasibility assessment, and conversion of feasible designs to practical pushbacks can then proceed in accordance with standard processing.

30 In some embodiments the pushback design process can include the step of selecting a different compactness factor and repeating the pushback design process using the different compactness factor. This may be done for a range of compactness factor values. The examples below show the impact of varying compactness factor on the pushback designs.

### 35 *Computational Experiments*

A small size mine, with 2,400 blocks, has been used to test the bubble pit formulation. A maximum total tonnage of the pit was defined as 8,000,000. Figure 4 shows the effect of the compactness factor for the first pushback. Graph 410 shows the pushback using a compactness factor of 0 ( $c=0$ ) allowing maximum flexibility in the bubble geometry, and 100% weighting for NPV and 0% for geometric volume. Graph 440 shown a pushback design using a compactness factor of 1000 ( $c=1000$ ) which represents around 50% weighting to minimise geometric surface area of the pushback, and 50% weighting for maximising NPV. Graphs 420,  $c=10$  and 430  $c=100$  represent compromise positions and different trade-offs between minimising surface area and maximising NPV, representing around a 15% and 30% weighting for minimising surface area respectively.

### *Pushback Design Case Study*

A prototype version of a mine planning system has implemented the bubble pit design method in software capable of running on small mines.

The inventors have also been exploring measures which can be used to compare different design approaches, so that the prototype system and software can be better benchmarked against commercial competitors. The idea is to compare when a pushback is mined in a final design as compared to optimised nested pits. The percentage overlap gives a good measure of how far the design is from the unconstrained optimised solution. The prototype software testing has given outcomes of around 235 MUS\$ as compared to 222 MUS\$ for an engineer design. So, this demonstrates how much better our algorithm is in forcing practical constraints without deviating too much from the highest value solution. Figure 5 shows the variation of the NPV versus the compactness factor for a prototype test of the pushback design algorithm compared with an engineer prepared design using the same mine data. In the example of Figure 5 a compactness factor of 100 was the minimum required to obtain practical designs. (The compactness factor of 100 after introducing the unit transformation factor was 50%).

A 3D representation of the output of the bubble pit solution is represented in Figure 6. Colors represent the position of the pushback generated iteratively with the 3D bubble pit formulation.

### ***Bubble Schedule***

Another application of the bubble clusters methodology is to introduce the connectivity and width constraints to the direct block scheduling problem.

A mine schedule consists of an extraction sequence that satisfies the mining and processing capacity/blending constraints. The extraction sequence must be mineable, i.e. every portion of the mine that is assigned to a period must satisfy practical constraints such as have a minimum width, be connected, etc. The traditional approach for open pit mine scheduling consists of subdividing the pushbacks by bench (or levels) and sequencing the mine using bench-level as the minimum granularity for the problem. Another approach, known as the direct block scheduling (DBS), consists of scheduling the mine block by block over a discretized time frame (periods). This approach has been widely studied in the literature, and recently has been implemented in some commercial software. However, most schedules obtained with the DBS methodology are not mineable. Therefore, the DBS is not suitable to solve the scheduling problem.

The bubble cluster method can be used to control the shape of the set of extracted blocks at each period, treating them as clusters in the DBS problem. As an example, we extend the formulation of the DBS problem, considering a second term in the objective function that stand for the minimisation of the cluster's surface area.

Consider a simple instance of the bubble scheduling problem defined as: find the mining sequence that maximises the economic profit (or other attribute) and minimises the external surface area of the blocks mined at each period, such that production targets

are satisfied (single process). Note that other constraints can be added to represent specific conditions of different mining requirements, but we show this simple problem to illustrate how the bubble clusters can be used in the bubble scheduling problem.

- 5 Let us assume a 3D block model, a regularly spaced 3D grid for which every element of the grid is a block (a 3D prism), for this formulation. The bubble clusters will be the result of merging a set of prismatic elements, which are the blocks. Note that the slope constraints are superimposed on the problem so that the spherical shapes of bubbles are replaced by cones on planar bubble clusters. These cones have least surface area enclosing a given volume amongst shapes satisfying the slope bounds. Since the  
10 problem has multiple objectives, we introduce a factor  $\lambda$  to weight both objectives. However, our method does not rely on any particular approach to solve multiple-objective optimisation problems.

15 *Sets and notation:*

- $b \in \mathbf{B}$  : Set of blocks  $\{0,1,...,B\}$ .
- $\hat{b} \in \widehat{\mathbf{B}}_b$  : Set of blocks that are adjacent to block  $b$ .
- $t \in T$  : Set of periods (clusters)  $\{0,1,...,T\}$ .
- $k \in K$  : Set of attributes  $\{0,1,...,K\}$ .
- $\bar{b} \in \bar{\mathbf{B}}_b$  : Set of slope precedence for block  $b$ .

20 *Parameters:*

- $S_b$ : external surface area of block  $b$ .
- $I_{b\hat{b}}$ : intersecting surface area between block  $b$  and  $\hat{b}$ .
- $\bar{M}_t$  : maximum mining capacity (tonnage) at period  $t$ .
- $M_b$  : tonnage of block  $b$ .
- $A_b^k$ : amount of attribute  $k$  associated to block  $b$ .
- $\bar{A}_t^k$ : maximum amount of attribute  $k$  accepted to be process at period  $t$ .
- $\underline{A}_t^k$ : minimum amount of attribute  $k$  required to be process at period  $t$ .
- $W_t$ : weight for the period  $t$ .
- $P_b$  : profit obtained after processing a ton of material.
- $C_b$  : mining cost of a tonnage of material.
- $\lambda$  : *compactness factor*. Weight (ranging from 0 to 1) to balance the importance of the two objectives in the optimisation problem.

35

*Variables:*

- $x_{bt}$ : binary, equal to one if the block  $b$  is mined at period  $t$ , zero otherwise.
- $f_{bt}^p$ : fraction of the block  $b$  that is sent to process at period  $t$ .
- $f_{bt}^w$ : fraction of the block  $b$  that is sent to waste at period  $t$ .
- $y_{b\hat{b}t}$ : binary, equal to one if blocks  $b$  and  $\hat{b}$  are mined in period  $t$ , zero otherwise.

*Objective function 3:*

$$\max z = (1 - \lambda) \sum_{t \in T} W_t \sum_{b \in \mathbf{B}} M_b \cdot (P_b \cdot f_{bt}^p - C_b \cdot x_{bt}) - \lambda \cdot \left( \sum_{t \in T} W_t \left( \sum_{b \in \mathbf{B}} S_b \cdot x_{bt} - \sum_{\substack{b \in \mathbf{B} \\ \hat{b} \in \widehat{\mathbf{B}}_b}} I_{b\hat{b}} \cdot y_{b\hat{b}t} \right) \right)$$

subject to:

$$\sum_{b \in B} M_b \cdot x_{bt} \leq \bar{M}_t \quad \forall p \in P \quad (13)$$

$$x_{bt} + x_{\hat{b}t} - y_{b\hat{b}t} \leq 1 \quad \forall b \in B, \forall \hat{b} \in \hat{B}_b, \forall t \in T \quad (14)$$

$$y_{b\hat{b}t} \leq x_{bt} \quad \forall b \in B, \forall \hat{b} \in \hat{B}_b, \forall t \in T \quad (15)$$

$$y_{b\hat{b}t} \leq x_{\hat{b}t} \quad \forall b \in B, \forall \hat{b} \in \hat{B}_b, \forall t \in T \quad (16)$$

$$\sum_{\tau \leq t} x_{b\tau} \leq \sum_{\tau \leq t} x_{\bar{b}\tau} \quad \forall b \in B, \forall \bar{b} \in \bar{B}_b, \forall t \in T \quad (17)$$

$$\sum_{t \in T} x_{bt} \leq 1 \quad \forall b \in B \quad (18)$$

$$\sum_{t \in T} f_{bt}^p \leq 1 \quad \forall b \in B \quad (19)$$

$$\sum_{t \in T} f_{bt}^w \leq 1 \quad \forall b \in B \quad (20)$$

$$x_{bt} = f_{bt}^p + f_{bt}^w \quad \forall b \in B, \forall t \in T \quad (21)$$

$$\underline{A}_t^k \leq \sum_{b \in B} A_b^k \cdot f_{bt}^p \leq \bar{A}_t^k \quad \forall t \in T, \forall k \in K \quad (22)$$

$$(x, y) \in \Omega \quad (\text{other constraints}) \quad (23)$$

*Explanation of objective function and constraints:*

The multi-objective function maximises the weighted value of the scheduled blocks and minimises the surface area of the blocks extracted in each period, combined by a weighting given by the compactness factor  $\lambda$ . Constraint (13) ensures that extraction does not exceed the maximum mining capacity. Constraints (14)-(16) identify which adjacent elements are extracted at the same period. Constraint (17) represents the vertical precedence (slope wall constraint). Constraints (18-21) ensure that the blocks can be extracted and processed in only one period. Constraint (22) enforces that processed material satisfies the processing blending requirements. Finally, constraint (23) might represent other operational restrictions (e.g., block-level or bench-pushback scheduling, stockpiling/re-handling constraints, and/or dumps capacity), and could involve bounds on  $x$  and  $y$ .

Similarly to the bubble pit procedure, the process of transforming the block model to a valid bubble scheduling is as follow. First, compute the set of parameters of the formulation  $(\mathbf{B}, \widehat{\mathbf{B}}_b, \mathbf{T}, \widehat{\mathbf{B}}_b, \mathbf{K}, \bar{\mathbf{M}}_t, A_b^k, \bar{A}_t^k, \underline{A}_t^k, W_t, P_b, C_b, \lambda)$ . Second, generate a set of problem instances (for the desired values of compactness factor  $\lambda$ ). Third, perform the optimisation of the mathematical model for the set of problem instances. This may be performed iteratively until a satisfactory or optimal extraction schedule solution is found. Finally, transform the result from the mathematical model to a block model file. In some embodiments, the scheduling process can include the step of selecting a different compactness factor and repeating the optimisation. This may be done for a range of compactness factor values.

### **Bubble cave**

There are a variety of underground mining methods, where the selection of the method mostly depends on the geomechanical characteristics of the ore and the host rock. Underground methods can broadly be divided into selective and massive (or caving) methods. Examples of caving methods are block caving, panel caving and sublevel caving. The sequencing problem for block and panel caving mines consists of defining the sequence of contiguous drawpoints (extraction points) that need to be *opened* during the life of the mine, some examples are shown in Figure 7.

Once the drawpoints are opened, the change of the in-situ stresses induces the propagation of the caving, beginning the gravitational flow of the material. The material that is above a drawpoint is called a *column*. Generally, the drawpoints share the same extraction level (footprint). Additionally, the caving propagation slope (the vertical slope that forms the broken material) mainly depends on geomechanical factors as well as the mining rates; an example is shown in Figure 8. The caving propagation slope can be modelled as a precedence constraint for a given set of geomechanical conditions and mining factors. Thus, considering both the drawpoints and the extraction columns as geometrical elements (for example regular blocks), the bubble cluster method can be used to optimise the mining sequence of caving mines. The precedence constraints defined by the caving propagation act similarly to the wall slope constraint in open pit mines. Other sets of operational constraints might be required to model other mining methods or situations. Note that the use of bubble clusters in the model tends to generate rounded shapes which may reduce the stress concentration in sharp corners of excavated volumes.

Underground ore extraction methods such as block caving and panel caving can be modelled as an inverted open pit mine, where the slope constraint is given by the caving propagation angle. Therefore, the same mathematical formulation can be applied to approach and solve caving design problems. Moreover, if constraint (5) from the 3D bubble pit formulation is removed, the resulting model might be sufficient to design the optimal boundary of underground stoping mines.

In large scale underground mines, there are several key problems which the bubble approach may successfully overcome. The first is to minimise geotechnical stress which can build up from ore removal. There are standard methods to deal with this, such as leaving pillars and filling cavities with waste and paste mix. A bubble shape may give substantial improvement on the distribution of geotechnical stress.

Another problem is over-break, where blasting can mix unwanted waste material with ore. Again, using bubble designs for stopes (mineable units) may decrease over-break, as the designs should yield more stable shapes than conventional stope designs.

5

### *Computational Experiments*

Real mining data was used to test the bubble scheduling formulation. A particular bench (level) of an iron ore mine was selected. The problem instance consisted of scheduling 3076 blocks over 5 periods. A mining and processing capacity of 1,200,000 and 800,000 tons per period was considered. Additionally, two blending constraints over attribute Fe (Iron) and Si (Silica) were required to satisfy the processing plant requirements. Thus, the processed material must have a minimum content of Fe of 55%, and a maximum content of Si of 5%.

15

Two cases of compactness factors were studied, the first one considering  $\lambda$  equal to zero (Figure 9a) and the second, considering a value of  $\lambda$  equal to 1 (Figure 9b). Colors represent the time when the blocks should be extracted. The value of  $\lambda$  equal to zero means that we are not imposing any constraints over the clusters (no bubble clusters), therefore it is not surprising to see that blocks scheduled at each period are quite scattered in location. On the other hand, when  $\lambda$  is equal to 1 we are purely minimising the external surface area of the set of blocks scheduled by period.

20

A significant advantage is that bubble scheduling for a problem requiring blending constraints is able to construct sequences of blocks per period, which are practical to extract, as they are contiguously located. Scheduling for maximum NPV does not produce such sequences and so manual smoothing of the schedule is required, which will be time-consuming and destroy value.

25

### ***Ramp design***

During the mine exploitation, the ore is hauled to processing plants, while waste material is transported to waste dumps. Thus, open pit mines require a complete road network which connects the extraction sectors inside the pit to all possible destinations outside the pit. In-pit ramps are roads that connect the working phases to the pit exits; these require the removal of a great amount of material for their construction. The total tonnage of ore and waste and the pit shape can change dramatically after the addition of the in-pit ramps.

35

In-pit ramps must connect the bottom of the pit to a given pit exit at the top of the excavation (see Figure 10). It is important to note that conditions, such as poor rock quality strength, could disqualify some sectors of the pit for ramp construction. In addition, material above the ramp must be extracted to satisfy the maximum condition of pit wall slope. Decision variables of the problem are, the ramp starting point (at the bottom of the pit), the direction and slope of each section of the ramp spiral, and the number and locations of switchbacks, where the ramp makes a sharp turn.

45

Each practical pushback must have a ramp from top to bottom. Therefore, a natural step after the calculation of semi-practical pushbacks with the bubble pit formulation is to design a ramp for each pushback.

- 5 Current practice is for ramps to be designed using CAD tools. There are no commercial tools available currently to design ramps automatically.

10 A major advantage of incorporating an automatic ramp design tool together with the bubble pit pushback method is that realistic scheduling production of the mine can then be achieved. In particular, an accurate economic valuation of the design can be performed and mine planners can run multiple designs to test economic assumptions about commodity prices. Current practice is to perform a rough best- and worst-case analysis, which gives rather crude bounds on the value of a design.

15 For this purpose, we present a linear programming formulation to find a minimum cost ramp with vertical and horizontal alignment constraints for a given pit and a given ramp width. This formulation takes into account the stripping associated with the ramp excavation. (Stripping is the removal of material above the ramp to satisfy the slope conditions).

20 The objective is to find the least cost ramp, with the physical constraint of slope bound, where the major cost is stripping but also an additional term is added for each sharp turn, i.e. a switch-back. This can take into account extra maintenance and haulage costs. The ramp is built then to run from the top to the bottom of the pit. The initial and final points can be either be pre-determined or determined by the design.

25 The excavation cost, the topography and the initial and final ramp coordinates are the main inputs of the problem. The geological model is assumed to be represented as a regularly spaced set of blocks (equal dimensions in coordinates  $x$  and  $y$ ) and must include the topography of the pit. We assume the ramp width is equivalent to the block size in  $x$  or  $y$  and the dimension in  $z$  represents the maximum ramp slope (max. slope  $\leq \frac{z}{x}$  or  $\frac{z}{y}$ ). For a standard block model this may necessitate, first, dividing every block into several horizontal slices. An arc is a small segment of path from a block to an adjacent block. The blocks are labelled by integers.

35 To consider the stripping associated with the ramp, we define a set of vertical precedences (upward), or blocks above the ramp, that need to be extracted to ensure a minimum wall slope. Similarly, to avoid the construction of the ramp above excavated blocks, we consider a set of vertical precedences (downward) to ensure a feasible design. Additionally, two artificial blocks, the source block  $s$  and the sink block  $t$ , are created to preserve the flow in the graph.

Sets:

- 45  $N$ : Set of blocks in the block model.  
 $A_i^+$ : Set of blocks  $j$  such that there is an arc from  $j$  to  $i$ .  
 $A_i^-$ : Set of blocks  $j$  such that there is an arc from  $i$  to  $j$ .  
 $\Gamma_i^+$ : Set of vertical precedences of block  $i$  (upward).  
 $\Gamma_i^-$ : Set of vertical precedences of block  $i$  (downward).

$R_i$ : Set of all possible ramp directions from block  $i$ .

$I$ : Set of possible ramp starting blocks.

$E$ : Set of possible ramp ending blocks.

5 Parameters:

$c_i$ : extraction cost of block  $i$ .

$p_i^{h,j}$ : change of direction cost in ramp segment  $i$ , where  $h$  is the incoming direction and  $j$  is the outgoing direction at  $i$  with  $h, j \in R_i$  and  $h \neq j$ .

10 Variables:

$r_i$ : binary, equal to 1 if the block  $i$  is selected as a ramp, 0 otherwise.

$x_i$ : binary, equal to 1 if the block  $i$  is extracted, 0 otherwise.

$v_i^{h,j}$ : binary, equal to 1 if the ramp changes direction from  $h$  to  $j$  (or from  $j$  to  $h$ )

15 at block  $i$ , 0 otherwise ( $h, j \in R_i$  and  $h \neq j$ ).

$a_{i,j}$ : binary, for each arc from a block  $i$  to a block  $j$  equal to 1 if  $i$  and  $j$  are selected as ramp blocks, 0 otherwise ( $i \neq j$ ).

Objective function 4:

$$\min_{x,r} \left\{ \sum_{i \in N} c_i \cdot x_i + \sum_{\substack{i \in N \\ h, j \in R_i \\ h \neq j}} p_i^{h,j} \cdot v_i^{h,j} \right\} \quad (1)$$

20

Subject to:

$$x_i \leq x_j \quad \forall i \in N, \forall j \in \Gamma_i^+ \quad (2)$$

$$r_i \leq x_i \quad \forall i \in N \quad (3)$$

$$r_i \leq 1 - x_j \quad \forall i \in N, \forall j \in \Gamma_i^- \quad (4)$$

$$a_{i,j} \leq r_i \quad \forall i \in N, \forall j \in R_i \quad (5)$$

$$a_{i,j} \leq r_j \quad \forall i \in N, \forall j \in R_i \quad (6)$$

$$r_i + r_j \leq 1 + a_{i,j} \quad \forall i \in N, \forall j \in R_i \quad (7)$$

$$\sum_{j \in I} a_{s,j} = 1 \quad (8)$$

$$\sum_{i \in E} a_{i,t} = 1 \quad (9)$$

$$\sum_{i \in I} a_{i,s} = 0 \quad (10)$$

$$\sum_{j \in E} a_{t,j} = 0 \quad (11)$$

$$\sum_{j \in A^-} a_{i,j} = \sum_{i \in A^+} a_{j,i} \quad \forall i \in N \quad (12)$$

$$a_{h,i} + a_{i,j} + a_{j,i} + a_{i,h} \leq v_i^{h,j} + 1 \quad \forall i \in N, \forall h, j \in R_i, h \neq j \quad (13)$$

$$v_i^{h,j} \leq a_{h,i} + a_{i,j} \quad \forall i \in N, \forall h, j \in R_i, h \neq j \quad (14)$$

$$v_i^{h,j} \leq a_{j,i} + a_{i,h} \quad \forall i \in N, \forall h, j \in R_i, h \neq j \quad (15)$$

$$a_{i,j} + a_{j,i} \leq 1 \quad \forall i, j \in N \quad (16)$$

$$x_i, r_i, a_{i,j}, v_i^{h,j} \in \{0, 1\}, \quad \forall i, j \in N \quad (17)$$

Objective function (1) minimises the cost of the blocks extracted due to the ramp construction plus the cost of the changes of directions (switch-backs) in the ramp.

- 5 Constraint (2) ensures the vertical precedences in the extraction of any block. Constraint (3) imposes that every block flagged as ramp must be extracted. Constraint (4) avoids building a ramp in the air.

- 10 Constraints (5) to (7) flag the blocks  $r_i$  and  $r_j$  if only if the arc  $a_{i,j}$  is selected. Constraints (8) and (9) force one of the blocks in  $I$  and one of the blocks in  $E$  to be selected as a starting and finishing point for the ramp respectively. Constraints (10), (11) and (12) ensure the connectivity of the ramp. Constraints (13) to (15) identify the change of direction at each segment of the ramp. Constraint (16) avoids the ramp returning back to the same position. Finally, constraint (17) defines the scope of the variables.

15

In an alternative embodiment a binary linear model to find the minimum cost ramp is formulated. The inputs for the model are divided into two types: the geological inputs, which will include the geological block model, the topography, the contour of the pit, and the dependencies between blocks based on the required pit slopes; and operational

20 inputs, which include various costs, the ramp width, and the set of possible starting and

final points for the ramp. The geological model is assumed to be represented as a regularly spaced set of blocks (equal dimensions in coordinates  $x$  and  $y$ ), and must include the contour of the pit shape we are aiming to achieve. To reduce the size of the problem, blocks that are too far from the contour of the pit can be discarded. We  
 5 assume the ramp width is equal to the block size in  $x$  ( $dx$ ) or  $y$  ( $dy$ )) and the dimension in  $z$  ( $dz$ ) represents the maximum ramp gradient. For a standard block model this may necessitate first dividing every block into several horizontal slices.

To consider the stripping associated with the ramp, we define a set of vertical  
 10 precedences (upward), or blocks above the ramp, that need to be extracted to ensure a minimum wall slope. Similarly, to avoid the construction of the ramp above excavated blocks, we consider a set of vertical precedences (downward) to ensure that no block corresponding to the floor of a ramp is extracted.

Formally, the precedence relationships are represented in terms of the digraph  $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{A}_1)$  where  $\mathcal{V}_1$  is the set of blocks and  $(u, v) \in \mathcal{A}_1$  only if the block  $u$  must be extracted before the block  $v$ . Similarly, the condition that the ramp cannot be built above excavated blocks is represented by a digraph  $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{A}_2)$  where  $(u, v) \in \mathcal{A}_2$  represents the constraint that if the block  $u$  is selected as ramp, then the block  $v$  must  
 20 not be extracted.

The ramp is represented as a connected path of adjacent blocks from  $S$  to  $T$  (two artificial blocks that represent the starting and ending points of the ramp). Formally, the possible adjacencies between blocks is represented in terms of the digraph  $\mathcal{G}_3 =$   
 25  $(\mathcal{V}_3, \mathcal{A}_3)$ , where there is one vertex  $v \in \mathcal{V}_3$  for each block that can be selected as part of a ramp, plus the initial vertex  $S$  and the final vertex  $T$ . The set of arcs  $(u, v) \in \mathcal{A}_3$  consists of all pairs of vertices that can be selected as adjacent blocks in any feasible ramp. Additionally, we add one edge from  $S$  to every node that can be selected to begin the ramp, and one edge per vertex that can end the ramp to  $T$ .

Indices, sets and parameters:

$d \in \mathcal{D}$  : types of change of direction (e.g. slight left turn, right sharp turn, switchbacks, etc.).

$\mathcal{A}'_3$  : Set of arcs in  $\mathcal{A}_3$  other than the arcs that end at vertex  $T$ ,  $\mathcal{A}'_3 = \mathcal{A}_3 - \{(i, j) \text{ such that } j = T\}$ .

$\mathcal{A}''_3$  : Set of arcs in  $\mathcal{A}_3$  other than the arcs that start at vertex  $S$  or end at vertex  $T$ ,  $\mathcal{A}''_3 = \mathcal{A}_3 - \{(i, j) \text{ such that } i = S \text{ or } j = T\}$ .

$\mathcal{F}$  : a mapping function  $\mathcal{F} : \{(i, j, k) | (i, j), (j, k) \in \mathcal{A}''_3\} \rightarrow \mathcal{D}$ .

$C_i$  : extraction cost of block  $i$ .

$H_{i,j}$  : haulage cost associated with arc  $(i, j) \in \mathcal{A}''_3$ . Note that the haulage cost  $H_{i,j}$  of any arc  $(i, j)$  should account for the total tonnage that will traverse that arc over the life of the mine.

$P_i^d$  : change of direction cost at block  $i$ , where  $d \in \mathcal{D}$  is the type of change of direction.

Variables:

$r_i$  : binary, equal to 1 if the block  $i$  is selected as a ramp, 0 otherwise.

$x_i$  : binary, equal to 1 if the block  $i$  is extracted, 0 otherwise.

$v_i^d$  : binary, equal to 1 if the ramp undergoes a change in direction of type  $d$  at block  $i$ , 0 otherwise.

$a_{i,j}$  : binary, equal to 1 if the arc  $(i,j) \in \mathcal{A}_3$  is selected as a segment of the ramp, 0 otherwise.

Objective function 5:

$$\text{Min: } \sum_{i \in \mathcal{V}_1} C_i \cdot x_i + \sum_{(i,j) \in \mathcal{A}_3^g} H_{i,j} \cdot a_{i,j} + \sum_{i \in \mathcal{V}_3} \sum_{d \in \mathcal{D}} P_i^d \cdot v_i^d$$

5

The objective function 5 minimises the cost of the blocks extracted due to the ramp construction together with the haulage cost and those costs associated with changes of direction of the ramp.

Geotechnical constraints:

$$x_i \leq x_j \quad (i,j) \in \mathcal{A}_1 \quad (1)$$

$$r_i \leq x_i \quad i \in \mathcal{V}_1 \quad (2)$$

$$r_i \leq 1 - x_j \quad (i,j) \in \mathcal{A}_2 \quad (3)$$

10

Constraint(1) ensures the vertical precedences are honoured in the extraction of any block. Constraint (2) ensures that every block flagged as part of the ramp must be extracted. Constraint(3) avoids building a ramp in the air.

Ramp constraints:

$$\sum_{(i,j) \in \mathcal{A}_3^g} a_{i,j} \leq r_j \quad (4)$$

$$r_i + r_j \leq 1 + a_{i,j} \quad (i,j) \in \mathcal{A}_3'' \quad (5)$$

$$\sum_{(i,j) \in \mathcal{A}_3} a_{i,j} - \sum_{(j,k) \in \mathcal{A}_3} a_{j,k} = \begin{cases} -1, & \text{if } j = S \\ 1, & \text{if } j = T \\ 0, & \text{if } j \neq S \text{ and } j \neq T \end{cases} \quad (6)$$

15

Constraints (4) to (5) flags the block  $j$  as belonging to the ramp if and only if the block  $i$  belongs to the ramp and the arc  $a_{i,j}$  has been selected. Constraint (6) ensures the connectivity of the ramp.

Change of direction constraints:

$$a_{i,j} + a_{j,k} \leq v_j^d + 1 \quad (i,j), (j,k) \in \mathcal{A}_3'', \quad d = \mathcal{F}(i,j,k) \quad (7)$$

20

Constraint (7) identifies the change of direction at each segment of the ramp.

Scope of the variables:

$$x_i, r_i, a_{u,v}, v_j^d \in \{0,1\} \quad i \in V_1, d \in D, (u,v) \in A, j \in V_2 \quad (8)$$

Finally, Constraint (8) defines the scope of the variables, all of which are binary.

Embodiments of mine planning systems, for example implemented in software executable in a computer system (either using stand-alone processing and memory hardware, cloud processing and memory resources, or a combination of both) can utilise any or all of the disclosed bubble model excavation, pushback, scheduling and ramp design methods within planning modules. Such systems 1100 can include:

an input interface 1132 to enable input of user-controlled parameters, such as

target tonnage for pushbacks, compactness factor, economic data etc;

a data access interface 1145 for accessing databases 1140 or other storage repositories storing data such as mine block models, scheduling data, and optionally existing mine plans;

processing 1110 and memory 1120 resources for executing one or more software modules for mine planning processes; such modules can include a clustering module 1150, scheduling module 1155, excavation module 1160, bubble pit pushback design module 1170, and a ramp design module 1165;

an output interface 1135 to enable output of mine planning data to a user for assessment and use in implementation of a mine plan. For example, the output interface may include a display module or a formatting module configured to prepare data in a format enabling output and display via external systems. The excavation module may be configured to implement planning for caving, stoping or other excavation embodiments. The mathematical model 1190 may be stored in system memory 1120 for application by the various modules. Resulting mine plans, designs or schedules generated by the system can be stored in memory 1120 or output via the user interface 1130.

In an embodiment the system may be implemented in a computer having a user interface 1130, processor 1110 and memory 1120. Alternatively, a computer, tablet or other device may be utilised as a user interface to access an online server implementing the system. Thin or thick client embodiments may be utilised.

In an embodiment, the described mining design methods may be implemented in a software as a service embodiment, to enable existing mine designs (for example for mines already in operation) to be input to enable pushback and/or ramp redesign and optimisation for ongoing operations.

The inventors believe they have devised the first and only method to design pushbacks for open cut mines, which simultaneously satisfy operational constraints and maximise NPV. This could be used by large mining companies with in-house design capability, mining consulting companies and mining software providers. Our invention fits in the core of the open pit mine planning value chain and allows to both optimise and automate the pushback design stage.

It will be understood to persons skilled in the art of the invention that many modifications may be made without departing from the spirit and scope of the invention.

5 In the claims which follow and in the preceding description of the invention, except where the context requires otherwise due to express language or necessary implication, the word “comprise” or variations such as “comprises” or “comprising” is used in an inclusive sense, i.e. to specify the presence of the stated features but not to preclude the presence or addition of further features in various embodiments of the invention.

It is to be understood that, if any prior art publication is referred to herein, such reference does not constitute an admission that the publication forms a part of the common general knowledge in the art, in Australia or any other country.

## Claims

1. A computer implemented method of mine design, the method comprising:  
 obtaining a three-dimensional (3D) model for a mine, the 3D model  
 5 characterizing physical material of the mine as a plurality of geometric elements, each  
 geometric element representing a portion of the physical material extractable from the  
 mine;  
 applying a mathematical model, based on mathematical modelling of behavior of  
 connected bubbles, to cluster geometric elements from the three-dimensional (3D)  
 10 model based on physical location and properties of the physical material for each of the  
 geometric elements; and  
 selecting at least one set of a plurality of contiguous geometric elements for  
 extraction.
- 15 2. The method as claimed in claim 1 wherein each set of a plurality of contiguous  
 blocks is associated with a phase of a mining process.
3. The method as claimed in claim 1 or claim 2 wherein the mathematical model  
 includes mathematical constraints reflecting operational constraints for the mine type.
- 20 4. The method as claimed in claim 3 wherein the mine type is an open cut mine and the  
 mine design comprises push backs, the operational constraints include: connectivity,  
 minimum bench width and appropriate angles between pushbacks.
- 25 5. The method as claimed in claim 4 wherein for each pushback the mathematical  
 model balances selection of contiguous blocks to minimise geometric surface area while  
 maximising economic value for each pushback.
- 30 6. The method as claimed in claim 5 wherein the mathematical model includes a  
 geometric compactness tradeoff factor to enable an operator controllable weighting  
 between maximising economic value and pushback geometry to be defined and input to  
 the mathematical model.

7. The method as claimed in claim 6 wherein the mathematical model comprises:

$$35 \quad \max \quad z = (1 - \lambda) \sum_{p \in P} W_p \sum_{b \in B} V_b \cdot x_b - \lambda \cdot \left( \sum_{p \in P} W_p \left( \sum_{b \in B} S_b \cdot x_{bp} - \sum_{\substack{\hat{b} \in \hat{B}_b \\ \hat{b} \in \hat{B}_b}} I_{b\hat{b}} \cdot y_{b\hat{b}p} \right) \right)$$

subject to:

$$\sum_{b \in B} A_p \cdot x_{bp} = \bar{A}_p \quad \forall p \in P \quad (6)$$

$$x_{bp} + x_{\hat{b}p} - y_{b\hat{b}p} \leq 1 \quad \forall b \in B, \forall \hat{b} \in \hat{B}_b, \forall p \in P \quad (7)$$

$$40 \quad y_{b\hat{b}p} \leq x_{bp} \quad \forall b \in B, \forall \hat{b} \in \hat{B}_b, \forall p \in P \quad (8)$$

$$y_{b\hat{b}p} \leq x_{\hat{b}p} \quad \forall b \in B, \forall \hat{b} \in \hat{B}_b, \forall p \in P \quad (9)$$

$$\sum_{\rho \leq p} x_{b\rho} \leq \sum_{\rho \leq p} x_{\bar{b}\rho} \quad \forall b \in \mathbf{B}, \forall \bar{b} \in \bar{\mathbf{B}}_b, \forall p \in \mathbf{P} \quad (10)$$

$$\sum_{p \in \mathbf{P}} x_{bp} \leq 1 \quad \forall b \in \mathbf{B} \quad (11)$$

$$(x, y) \in \Omega \quad (\text{other constraints}) \quad (12)$$

Where:

*Sets and notation:*

$b \in \mathbf{B}$  : Set of blocks  $\{0, 1, \dots, B\}$ .

10  $\hat{b} \in \widehat{\mathbf{B}}_b$  : Set of blocks that are adjacent to block  $b$ .

$p \in \mathbf{P}$  : Set of pushbacks (clusters)  $\{0, 1, \dots, P\}$ .

$\bar{b} \in \bar{\mathbf{B}}_b$  : Set of slope precedences for block  $b$ .

*Parameters:*

$S_b$ : external surface area of block  $b$

15  $I_{b\hat{b}}$ : intersecting surface area between block  $b$  and  $\hat{b}$

$A_b$ : amount of attribute A associated to block  $b$

$\bar{A}_p$ : total amount of attribute A required in pushback  $p$

$W_p$ : weight for the pushback  $p$

$V_b$  : economic value of block  $b$

20  $\lambda$  : compactness factor. Weight (ranging from 0 to 1) to balance the importance of the two objectives in the optimisation problem

*Variables:*

$x_{bp}$ : binary, equal to one if the block  $b$  is assigned to pushback  $p$ , zero otherwise

$y_{b\hat{b}p}$ : binary, equal to one if blocks  $b$  and  $\hat{b}$  belongs to pushback  $p$ , zero otherwise.

25

8. The method as claimed in claim 7 further comprising the steps of adjusting the compactness factor and producing a further pushback design.

30 9. The method as claimed in claim 8 wherein the compactness factor is adjusted incrementally to produce a plurality of pushback designs.

10. A method as claimed in claim 3 wherein the mine type is an underground mine where operational constraints include any one or more of connectivity and width.

35 11. A method as claimed in claim 10 wherein the operational constraints further includes geotechnical constraints.

40 12. A method as claimed in any one of the preceding claims further comprising the step of determining a time period for extraction for each set of a plurality of contiguous blocks.

13. The method as claimed in claim 12 further comprising applying a compactness factor.

14. The method as claimed in claim 13 wherein the compactness factor is iteratively adjusted and pluralities of sets of elements reselected to search for an optimal extraction schedule.

5 15. A method of designing ramps for a set of pushbacks based on a linear programming formulation to find a minimum cost ramp with vertical and horizontal alignment constraints for a given pit and a given ramp width, taking into account stripping associated with ramp excavation, by applying a geological model is assumed to be represented as a regularly spaced set of blocks including the topography of the pit, and  
10 ramp width equivalent to the block size in  $x$  or  $y$  and the dimension in  $z$  represents the maximum ramp slope ( $\max. \text{slope} \leq \frac{z}{x} \text{ or } \frac{z}{y}$ ).

16. The method of claim 14 wherein the ramp design is calculated in accordance with:

15 Sets:

$N$ : Set of blocks in the block model.

$A_i^+$ : Set of blocks  $j$  such that there is an arc from  $j$  to  $i$ .

$A_i^-$ : Set of blocks  $i$  such that there is an arc from  $i$  to  $j$ .

20  $\Gamma_i^+$ : Set of vertical precedences of block  $i$  (upward).

$\Gamma_i^-$ : Set of vertical precedences of block  $i$  (downward).

$R_i$ : Set of all possible ramp directions from block  $i$ .

$I$ : Set of possible ramp starting blocks.

$E$ : Set of possible ramp ending blocks.

25

Parameters:

$c_i$ : extraction cost of block  $i$ .

$p_i^{h,j}$ : change of direction cost in ramp segment  $i$ , where  $h$  is the incoming direction and  $j$  is the outgoing direction at  $i$  with  $h, j \in R_i$  and  $h \neq j$ .

30

Variables:

$r_i$ : binary, equal to 1 if the block  $i$  is selected as a ramp, 0 otherwise.

$x_i$ : binary, equal to 1 if the block  $i$  is extracted, 0 otherwise.

35  $v_i^{h,j}$ : binary, equal to 1 if the ramp changes direction from  $h$  to  $j$  (or from  $j$  to  $h$ ) at block  $i$ , 0 otherwise ( $h, j \in R_i$  and  $h \neq j$ ).

$a_{i,j}$ : binary, for each arc from a block  $i$  to a block  $j$  equal to 1 if  $i$  and  $j$  are selected as ramp blocks, 0 otherwise ( $i \neq j$ ).

40 Objective function 4:

$$\min_{x,v} \left\{ \sum_{i \in N} c_i \cdot x_i + \sum_{\substack{i \in N \\ h, j \in R_i \\ h \neq j}} p_i^{h,j} \cdot v_i^{h,j} \right\} \quad (1)$$

Subject to:

$$x_i \leq x_j \quad \forall i \in N, \forall j \in \Gamma_i^+ \quad (2)$$

$$r_i \leq x_i \quad \forall i \in N \quad (3)$$

$$r_i \leq 1 - x_j \quad \forall i \in N, \forall j \in \Gamma_i^- \quad (4)$$

$$a_{i,j} \leq r_i \quad \forall i \in N, \forall j \in R_i \quad (5)$$

$$a_{i,j} \leq r_j \quad \forall i \in N, \forall j \in R_i \quad (6)$$

$$r_i + r_j \leq 1 + a_{i,j} \quad \forall i \in N, \forall j \in R_i \quad (7)$$

$$\sum_{j \in I} a_{s,j} = 1 \quad (8)$$

$$\sum_{i \in E} a_{i,t} = 1 \quad (9)$$

$$\sum_{i \in I} a_{i,s} = 0 \quad (10)$$

$$\sum_{j \in E} a_{i,j} = 0 \quad (11)$$

$$\sum_{j \in A^-} a_{i,j} = \sum_{i \in A^+} a_{j,i} \quad \forall i \in N \quad (12)$$

$$a_{h,i} + a_{i,j} + a_{j,i} + a_{i,h} \leq v_i^{h,j} + 1 \quad \forall i \in N, \forall h, j \in R_i, h \neq j \quad (13)$$

$$v_i^{h,j} \leq a_{h,i} + a_{i,j} \quad \forall i \in N, \forall h, j \in R_i, h \neq j \quad (14)$$

$$v_i^{h,j} \leq a_{j,i} + a_{i,h} \quad \forall i \in N, \forall h, j \in R_i, h \neq j \quad (15)$$

$$a_{i,j} + a_{j,i} \leq 1 \quad \forall i, j \in N \quad (16)$$

$$x_i, r_i, a_{i,j}, v_i^{h,j} \in \{0, 1\}, \quad \forall i, j \in N \quad (17)$$

- 5 17. A computer implemented mine planning method comprising the steps of:  
 obtaining a mine 3D model defining three dimensional geological resources of a mine  
 as a regular grid of a plurality prismatic blocks, wherein each block represents a  
 defined volume of material, and wherein block characterisation data for each block  
 includes at least coordinate data, physical characteristic data and assigned economic  
 10 value data;  
 defining a target pushback tonnage for each pushback; and  
 determining one or more pushbacks for the mine using the block  
 characterization data, the target pushback tonnage a plurality of contiguous blocks  
 to comprise the pushback by applying a mathematical model for operational  
 15 constraints of connectivity, minimum bench width and appropriate angles between  
 pushbacks to select from the mine block model, based on physical location and  
 economic value estimates for each block, a plurality of contiguous blocks having a  
 combined weight equal to or less than the target pushback tonnage to comprise each  
 20 pushback.

18. The method as claimed in claim 10 wherein the mathematical model is based on mathematical modelling of behavior of connected bubbles, and for each pushback the mathematical model balances selection of continuous blocks to minimise geometric surface area while maximising economic value for each pushback.

19. The method as claimed in claim 18 wherein the mathematical model includes a geometric compactness tradeoff factor to enable an operator controllable weighting between maximising economic value and pushback geometry to be defined and input to the mathematical model.

20. The method as claimed in claim 18 wherein the mathematical model comprises:

$$\max_{x,y} \left\{ \sum_{i \in N} v_i \cdot x_i - c \cdot \left( \sum_{i \in N} 6 \cdot l_i^2 \cdot x_i - \sum_{\substack{i \in N \\ j \in \Omega_i}} l_i^2 \cdot y_{i,j} \right) \right\}$$

subject to:

$$\sum_{i \in N} w_i \cdot x_i - W \leq 0 \quad (4)$$

$$x_i - x_j \leq 0 \quad \forall i \in N, \forall j \in \Gamma_i^+ \quad (5)$$

$$x_i + x_j - y_{i,j} \leq 1 \quad \forall i \in N, \forall j \in \Omega_i \quad (6)$$

$$x_i, y_{i,j} \in \{0,1\} \quad \forall i, j \in N \quad (7)$$

Where:

- $N$  : Set of blocks in the block model
- $\Omega_i$ : Set of adjacent blocks to block  $i$
- $\Gamma_i^+$ : Set of vertical precedences of block  $i$  (upwards)
- $c$  : compactness factor
- $v_i$ : economic value of block  $i$
- $l_i$ : length of the block  $i$
- $w_i$ : tonnage of the block  $i$
- $W$ : total tonnage of the bubble pit
- $x_i$ : binary, equal to one if the block  $i$  is extracted, zero otherwise
- $y_{i,j}$ : binary, equal to one if blocks  $i$  and  $j$  are extracted, zero otherwise.

21. The method as claimed in claim 20 further comprising the steps of adjusting the compactness factor and producing a further pushback design.

22. The method as claimed in claim 15 wherein the compactness factor is adjusted incrementally to produce a plurality of pushback designs and further comprising the step of outputting one or more of the plurality of pushback designs.

23. The method as claimed in claim 15 further comprising the step of performing ramp design for one or more pushback designs.

24. The method as claimed in claim 17 wherein the ramp design is performed in accordance with the method of claims 8 and 9.

25. A mine planning system implemented using computer processing and memory resources, the system comprising at least one module configured to implement the method as claimed in any one of the preceding claims.

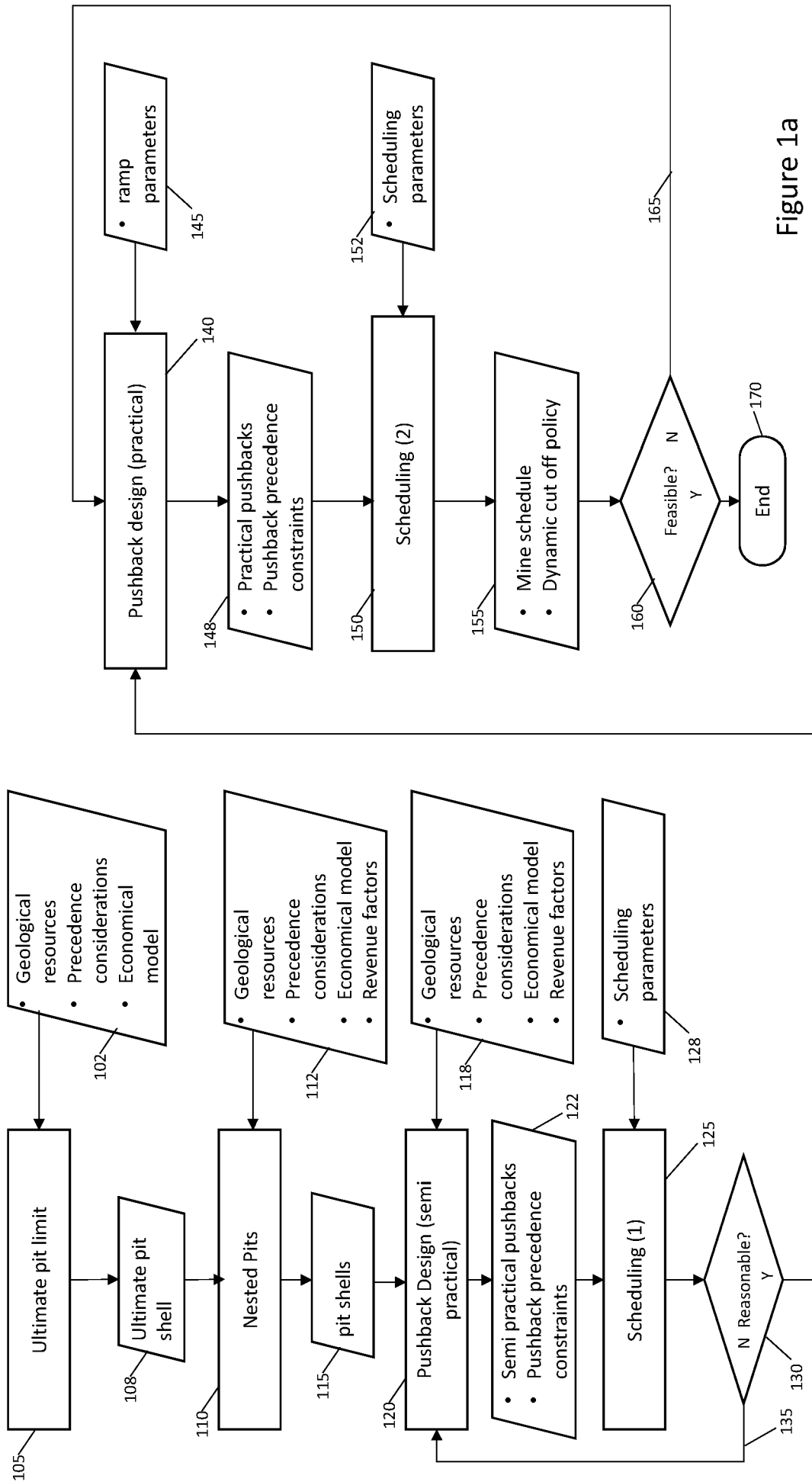
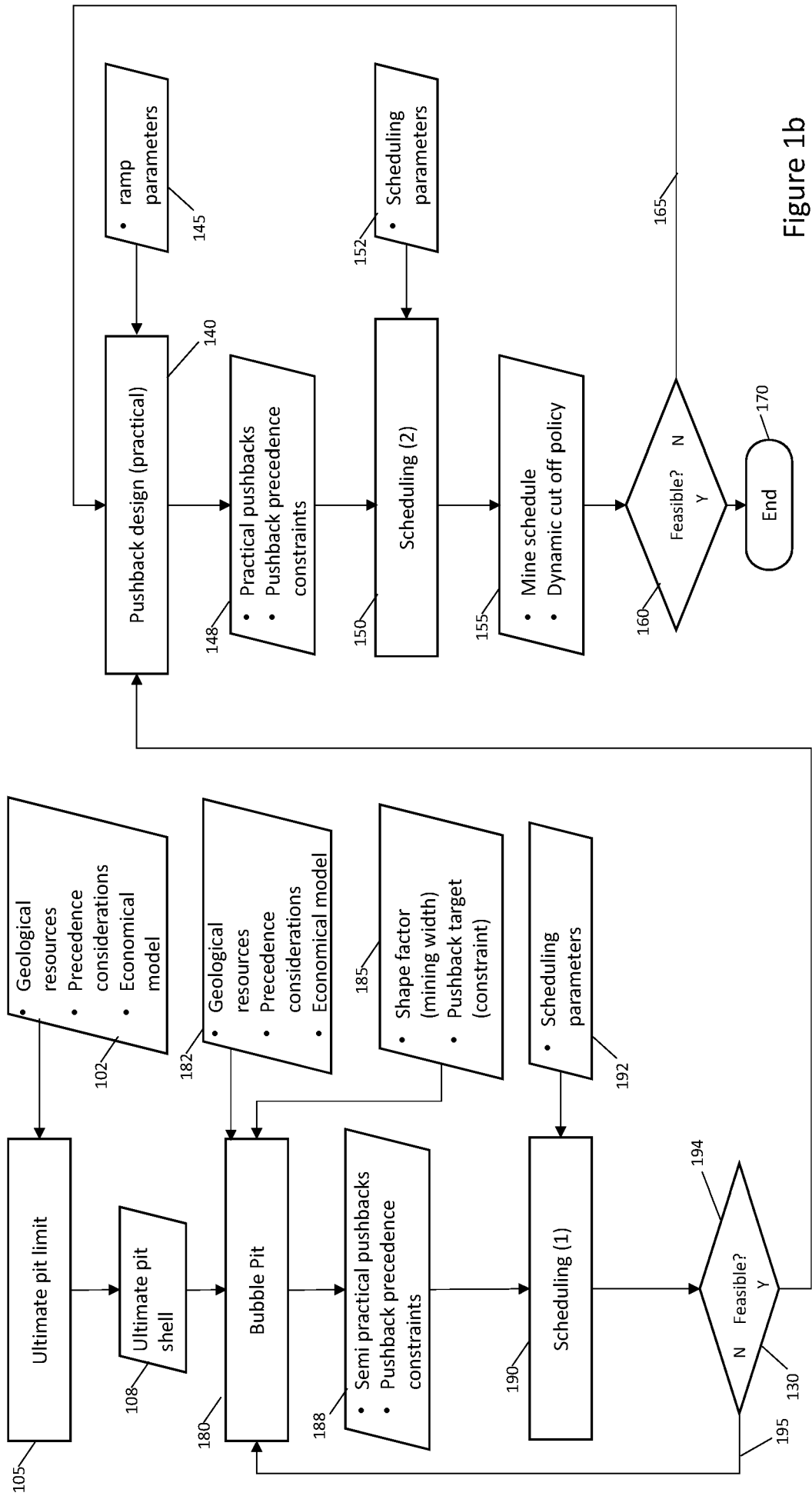


Figure 1a



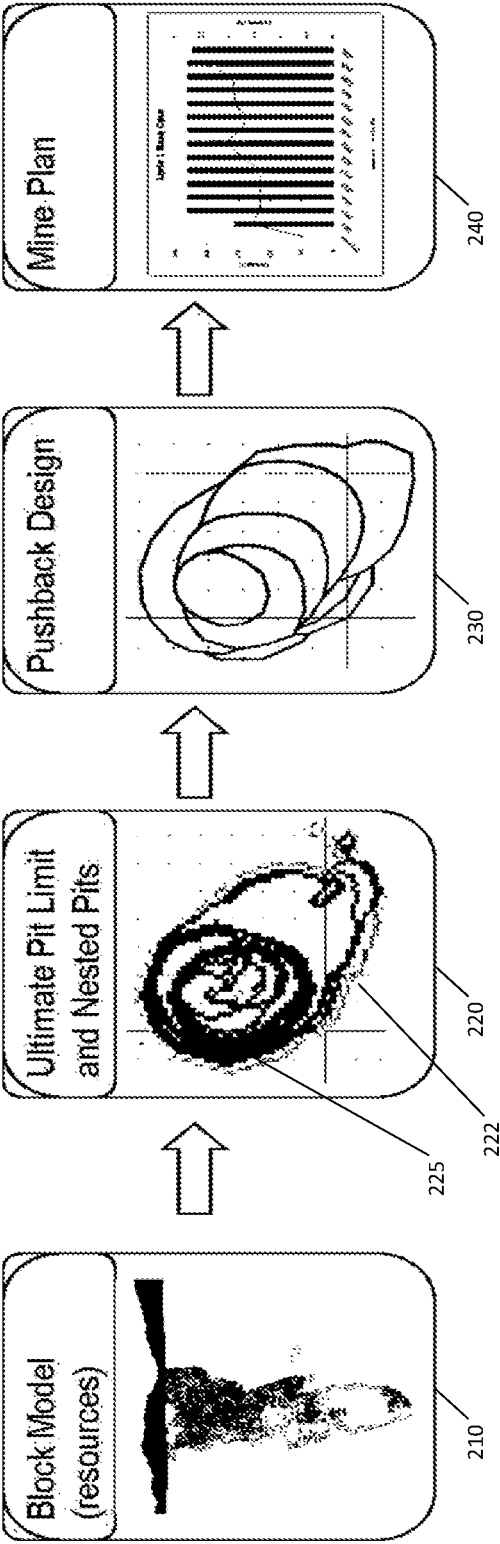


Figure 2

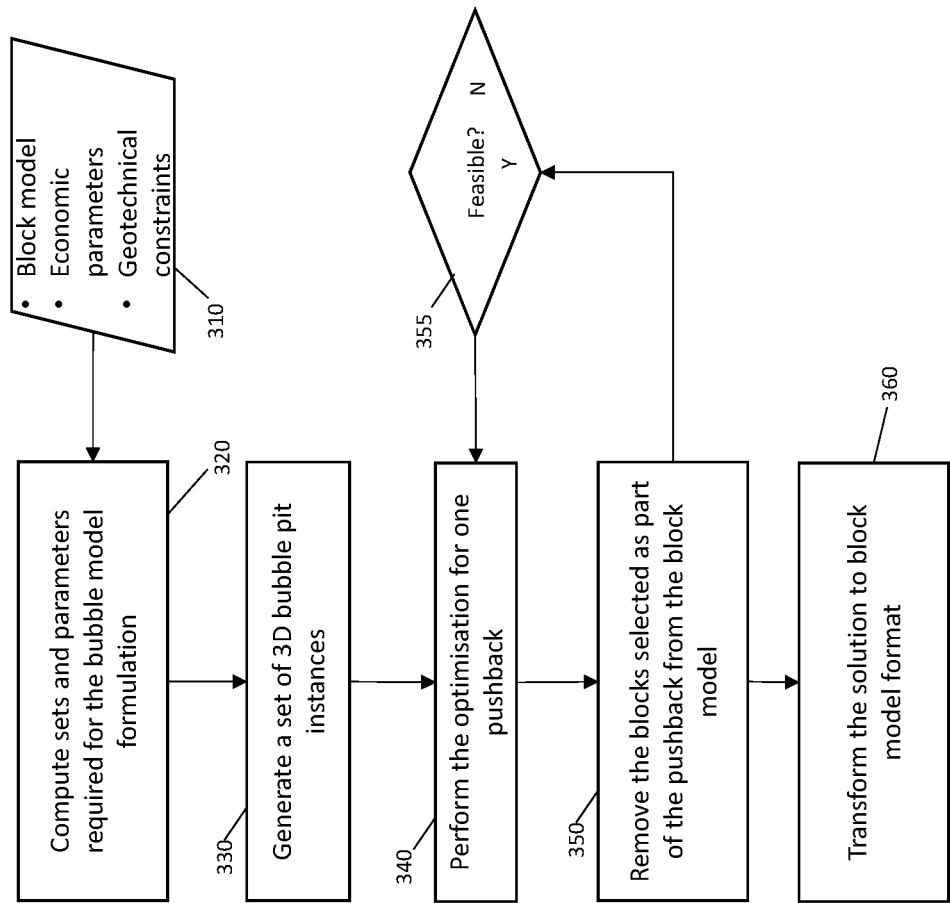


Figure 3

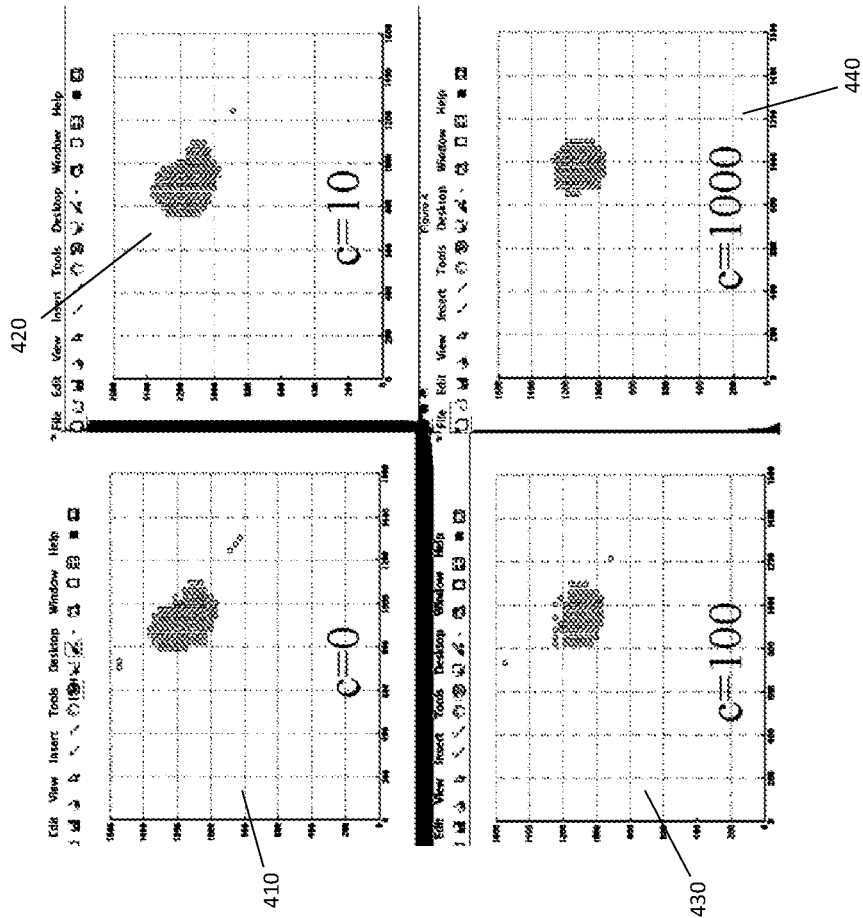


Figure 4

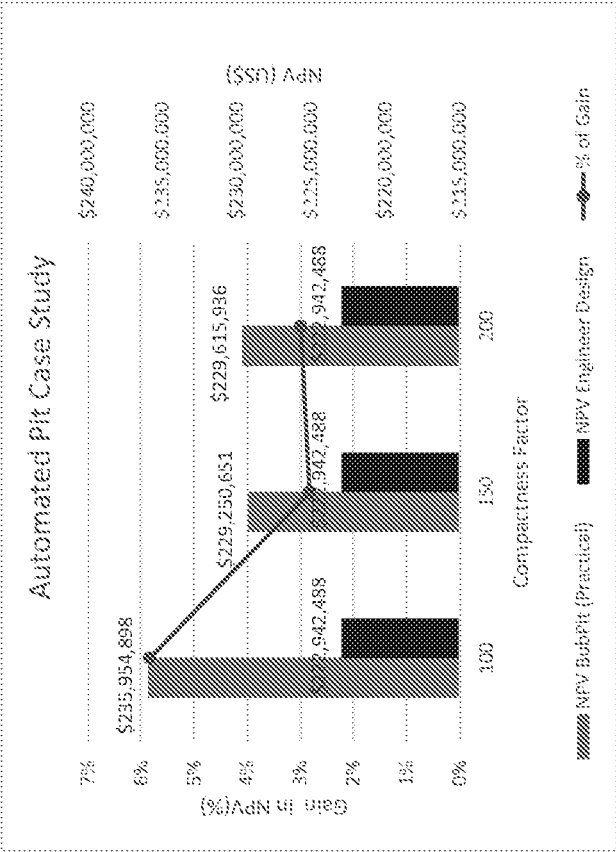


Figure 5

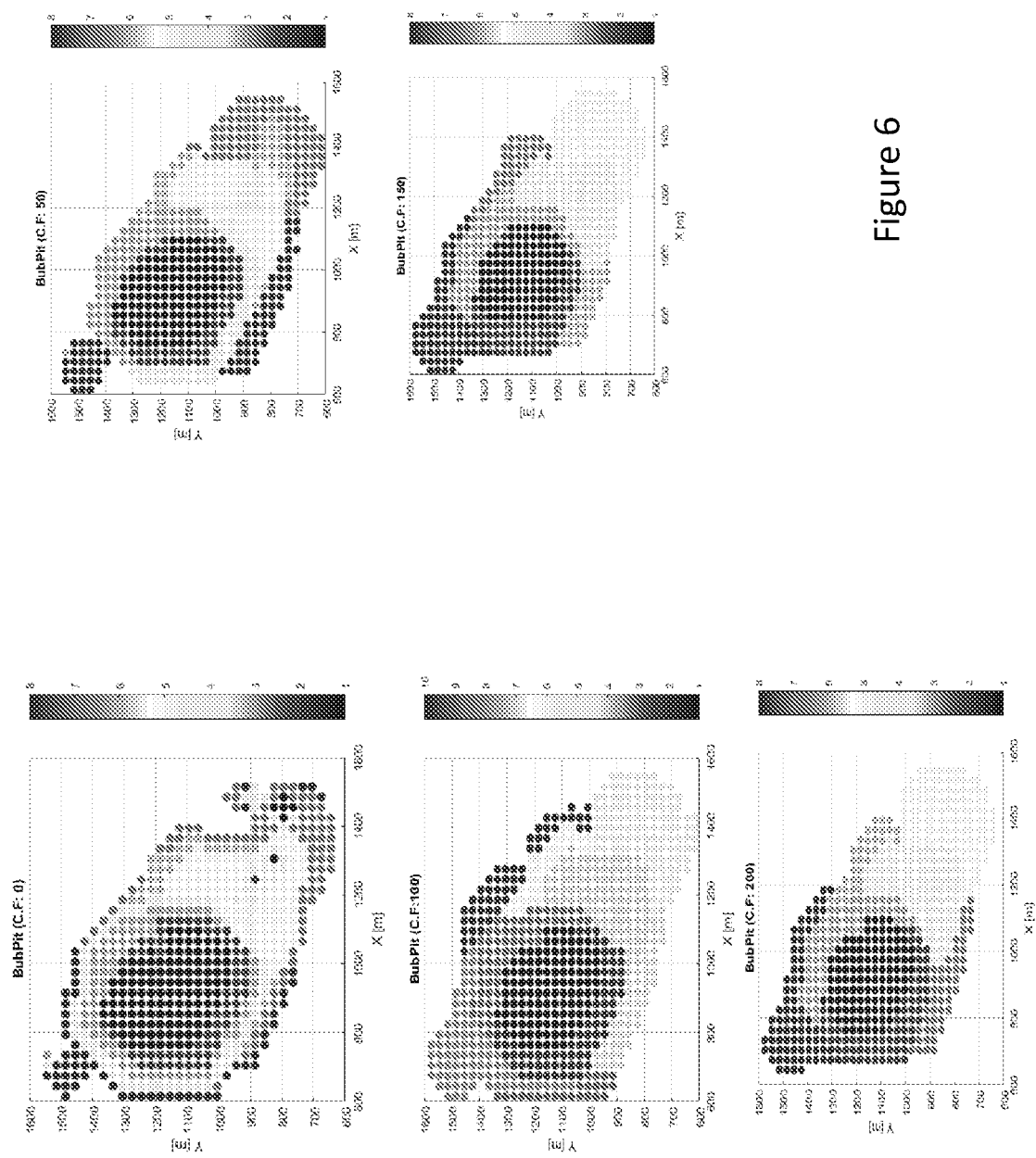


Figure 6

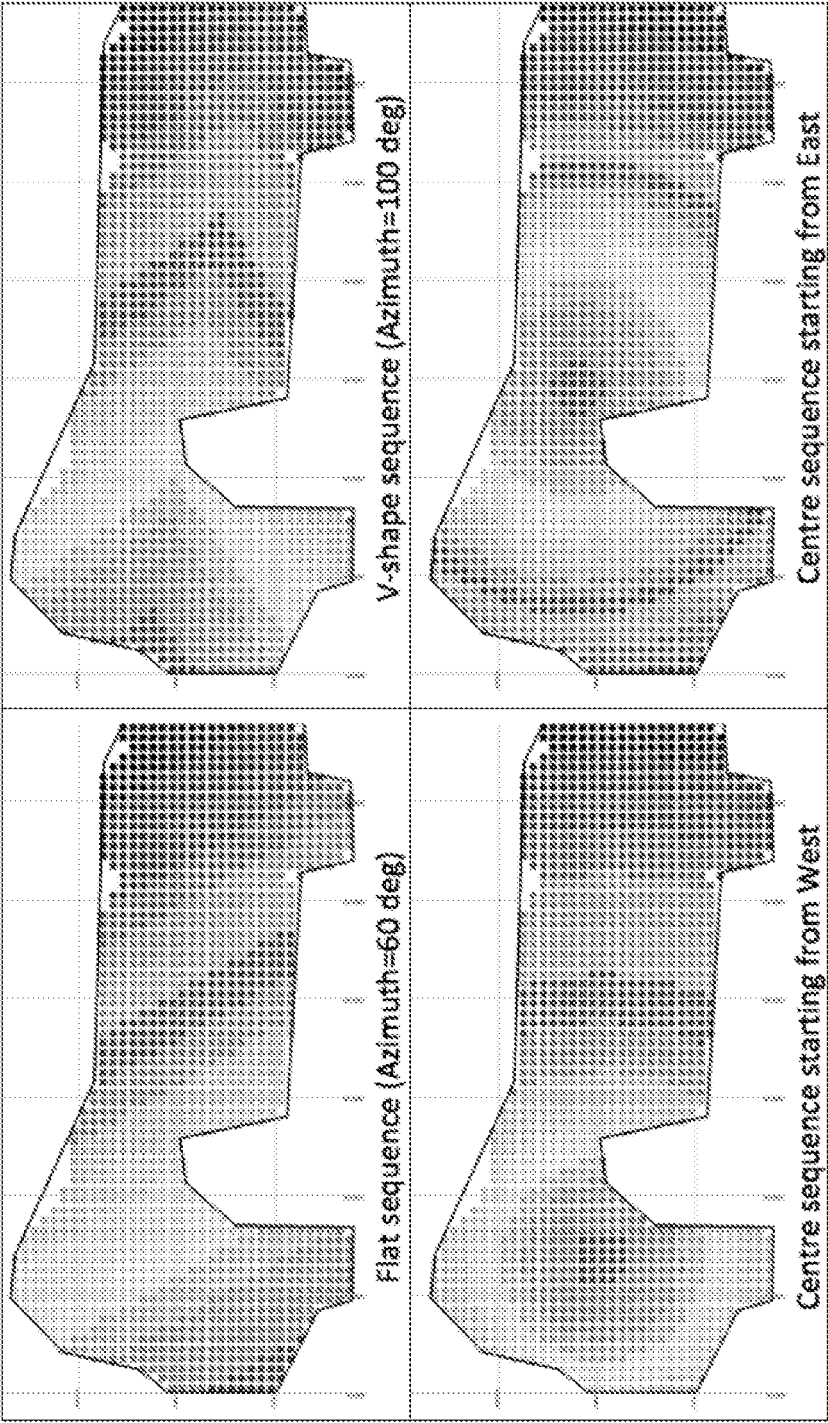


Figure 11. Four sequences used for the entire footprint.

Figure 7

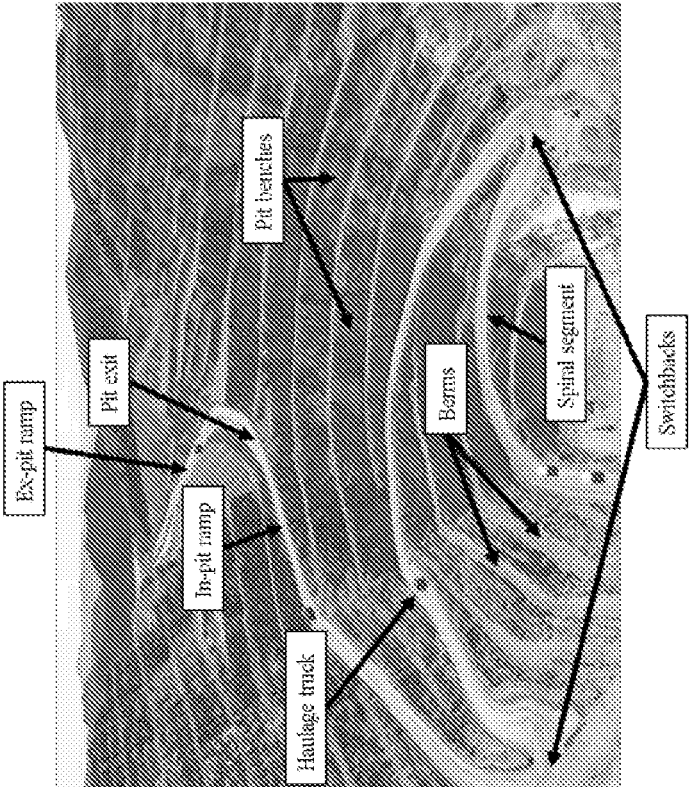


Figure 10

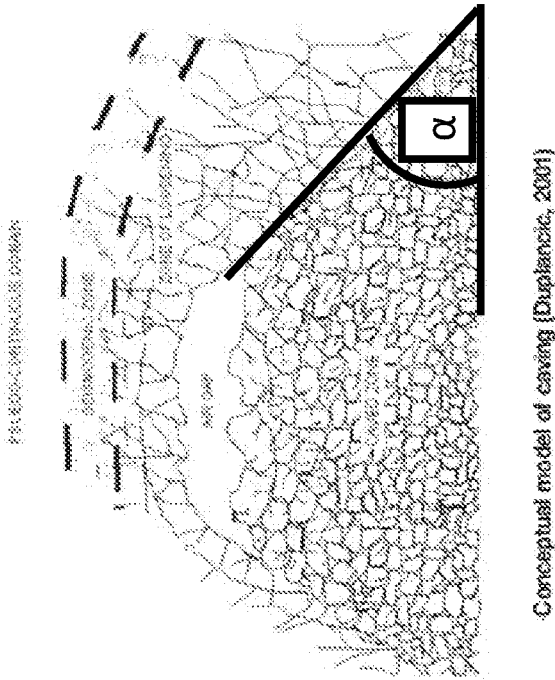


Figure 8

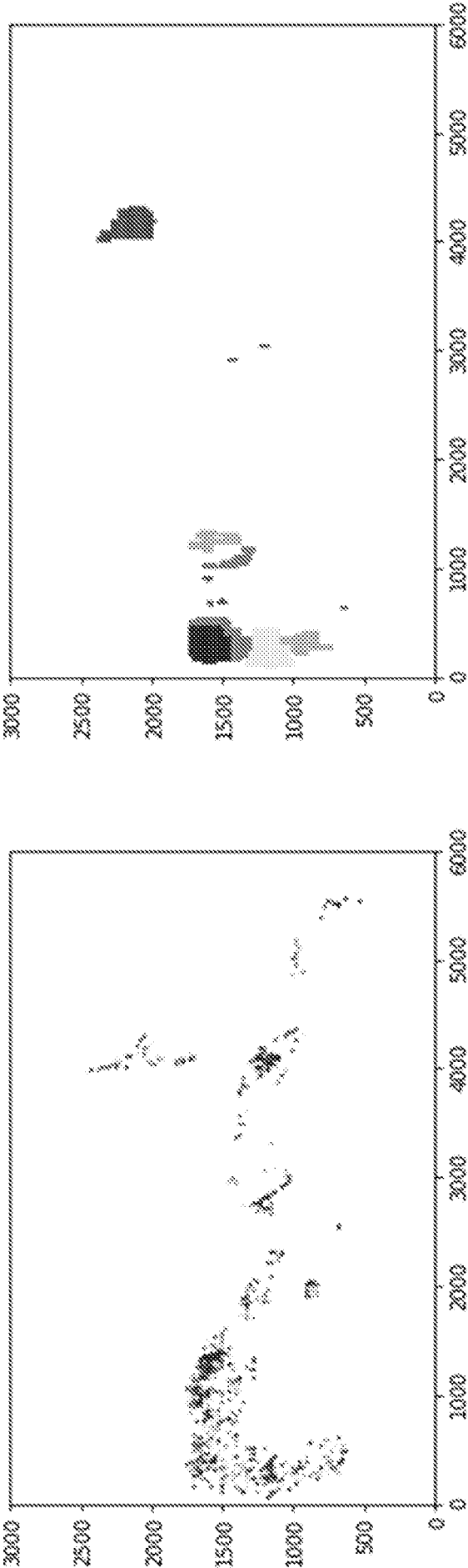


Figure 9a

Figure 9b

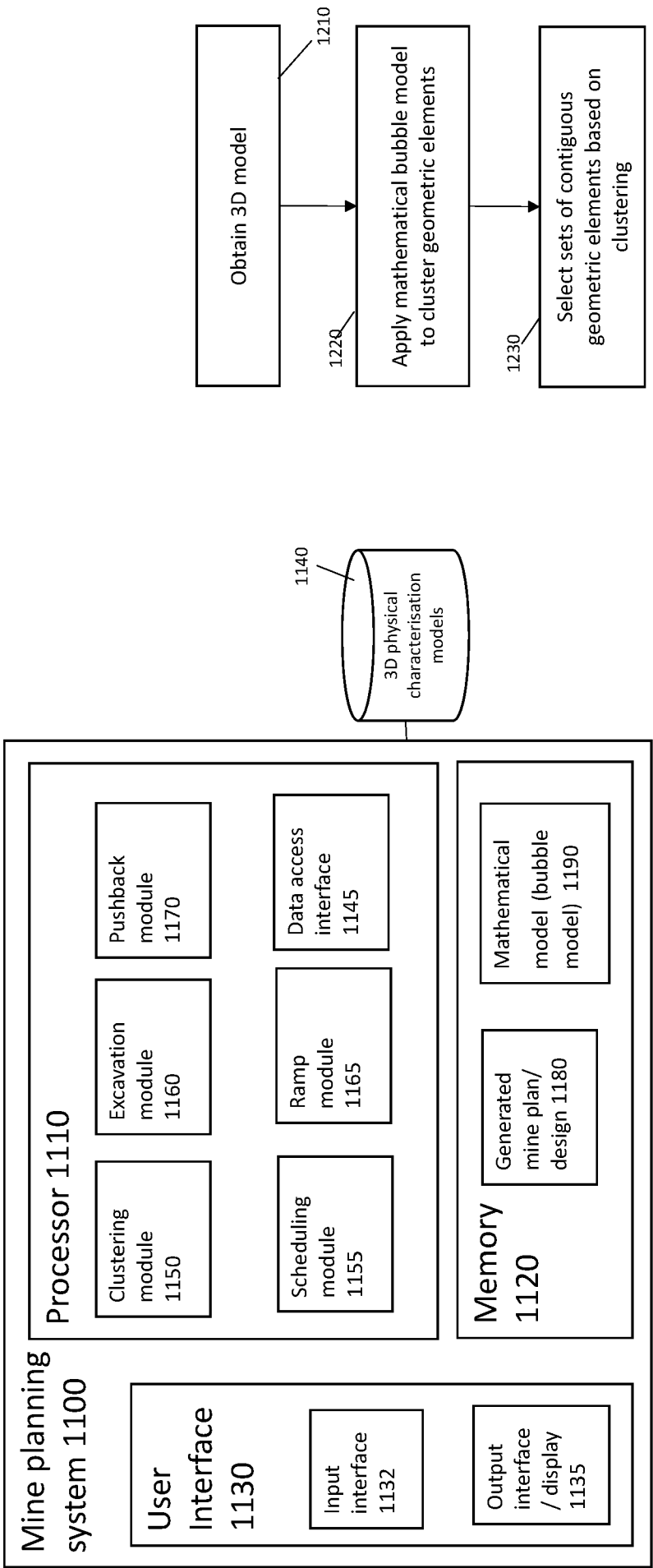


Figure 11

Figure 12