

A NEW INTEGER SOLUTION ALGORITHM TO SOLVE OPEN-
PIT MINE PRODUCTION SCHEDULING PROBLEMS

by

Canberk Aras

© Copyright by Canberk Aras, 2018

All Rights Reserved.

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School Mines in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Mining and Earth Systems Engineering).

Golden, Colorado

Date _____

Signed: _____
Canberk Aras

Signed: _____
Dr. Kadri Dagdelen
Thesis Advisor

Signed: _____
Dr. Thys Johnson
Thesis Co-Advisor

Golden, Colorado

Date _____

Signed: _____
Dr. Priscilla Nelson
Professor and Head
Department of Mining Engineering

ABSTRACT

The strategic open pit mine production scheduling problem is usually formulated as a large-scale integer programming problem that is very difficult to solve. The life of mine production scheduling problem is currently modeled on a block by block basis in order to decide which blocks should be extracted, when they should be extracted, and what to do with the blocks once they are extracted. Due to the nature of the problem, the decisions on whether or not to mine an individual block should be addressed in a binary context. However, the large size of some real instances (3–10 million blocks, 15–20 time periods) has made these models impossible to solve with currently available optimization solvers. To overcome this challenge, many attempts have been made to solve the problem with numerous heuristic and aggregation methods which cannot be proven to converge to the true optimal solution. On the other hand, linear programming relaxation of the real sized mine planning problems can be solved to a proven optimality by applying the existing exact decomposition algorithms. However, the solution obtained from the LP relaxation problems may result in fractional blocks being mined which cannot be implemented practically.

A novel integer solution algorithm is developed in this thesis which can solve the mine production scheduling problems modeled with multi capacities, grade blending, grade uncertainty, stockpiles, variable pit slopes, multi destinations and truck hours. It should be emphasized that the blocks will not have any pre-determined destinations based on grades, cycle times, material type or some other criteria since the best destination selection per block will be done automatically during the optimization process to maximize the NPV, in other words the dynamic cutoff concept is employed. Presently there is no known algorithm, either commercially available or presented in the literature, that can provide an optimal integer solution to the open pit mine production scheduling problem with capacity constraints together with lower and upper bound blending constraints. Therefore, the solution algorithm that can generate an optimal integer solution to a mine production scheduling problem that has never been solved will be a milestone in operations research. Moreover, a new cone pattern generation scheme is developed in order to integrate the variable pit slope angles based on complex geotechnical zones and multiple azimuths with any size block dimensions to the new integer solution algorithm.

TABLE OF CONTENTS

ABSTRACT.....	iii
LIST OF FIGURES	vi
LIST OF TABLES	xi
ACKNOWLEDGMENTS	xiii
CHAPTER 1. INTRODUCTION.....	1
1.1 Need for A Mine Production Scheduling Optimization.....	1
1.2 Scope of Work.....	3
CHAPTER 2. LITERATURE REVIEW	5
2.1 Ultimate Pit Limit Problem.....	5
2.2 Pushback Design	8
2.2.1 Shortcomings of the Pushback Design Methods	9
2.3 Mathematical Programming and Integer Solution Techniques for Production Scheduling	10
2.3.1 Linear Programming Solution Techniques	11
2.3.2 Mixed Integer Linear Programming Solution Techniques	14
2.3.2.1 Shortcomings of the Mixed Integer Linear Programming Solution Techniques	16
2.3.3 Integer Solution Techniques	17
2.3.3.1 Shortcomings of the Integer Solution Techniques.....	22
2.3.3.1.1 Lagrange Relaxation Technique	23
2.3.3.1.2 Fundamental Tree.....	23
2.3.3.1.3 Heuristic Techniques.....	24
2.3.3.1.3.1 Heuristic Techniques to Obtain Initial Integer Feasible Solution	24
2.3.3.1.3.2 Solution Improvement Heuristic Techniques.....	25
2.4 Summary of the Current State of Open Pit Mine Production Scheduling	27
CHAPTER 3. MATHEMATICAL MODEL FOR OPEN PIT MINE PRODUCTION SCHEDULING PROBLEM UNDER GRADE UNCERTAINTY	29
3.1 Mathematical Model Incorporated with Dynamic Cutoff Grade Strategy and Stockpiles Formulated with “At” Variables	31

3.2	Variable Substitution for Single Path “By” Variables	34
3.3	Variable Substitution for New Dual Path Double “By” Variables	39
3.4	Variable Substitution for Single Path Double “By” Variables	49
CHAPTER 4. NEW INTEGER SOLUTION ALGORITHM TO SOLVE MULTI-DESTINATION OPEN PIT MINE PRODUCTION SCHEDULING PROBLEM		57
4.1	The Bienstock-Zuckerberg Decomposition Algorithm.....	58
4.1.1	Small 2D BZ Example.....	65
4.2	Pseudoflow Algorithm	82
4.3	The New Integer Solution Algorithm.....	83
4.3.1	Small 2D Example	95
CHAPTER 5. IMPLEMENTATION OF THE NEW INTEGER SOLUTION ALGORITHM TO THE LARGE-SCALE OPEN PIT MINING PROBLEMS.....		124
5.1	Case Study 1 (McLaughlin Deposit).....	124
5.1.1	The Ultimate Pit.....	127
5.1.2	Mine Production Schedule.....	130
5.2	Case Study 2 (Gold Deposit).....	137
5.2.1	The Ultimate Pits	138
5.2.2	Mine Production Schedules	141
CHAPTER 6. CONCLUSIONS		150
6.1	Recommended Future Work	152
REFERENCES		154
APPENDIX A. MODELING WITH VARIABLE PIT SLOPE ANGLES		159
A.1	Background on Variable Pit Slope Angles.....	161
A.2	Constant Slope Angle Cone Pattern Generation	165
A.2.1	Complex Slope Angles Based on Geotechnical Zone.....	177
A.3	Cone Pattern Generation for Complex Slopes Based on Multiple Azimuths	179

LIST OF FIGURES

Figure 2.1: Most influential researchers on the development of the new integer solution algorithm	28
Figure 3.1: Process Flow of the Mine Production Scheduling Problem.....	30
Figure 3.2 : Node structure of the “at” variables and the “by” variables.....	34
Figure 3.3 : Node structure of the single path “by” variables and the dual path double “by” variables	40
Figure 3.4 : Node structure of the dual path double “by” variables and the single path double “by” variables.....	50
Figure 3.5 : Example showing variable reduction process using “ <i>zbd</i> ” variable substitution ..	51
Figure 4.1: Cross section of a polytope illustrating the subproblem solution space (outer polytope) and the original problem solution space (inner polytope). Blue dots are the extreme points of the subproblem solutions, the red dot is the optimum extreme point of the original problem solution	60
Figure 4.2: Node structure of the original problem variables and the subproblem variables	64
Figure 4.3: Economic block model for the potential destinations of the blocks.....	65
Figure 4.4: Production scheduling requirements	66
Figure 4.5: Fixing the destinations based on the highest block value. The blue colors represent the most valuable destinations for a block in time period 1 and period 2.....	69
Figure 4.6: Subproblem solution column and the mining plans generated per period	70
Figure 4.7: Subproblem solution column on the left is split into destination segments on the right in order to represent the original problem variable <i>zbd</i> node structure in the master problem	71
Figure 4.8: Modified block values by the duals for period 1 and period 2. The blue colors represent the most valuable destinations for a block in time period 1 and period 2	75
Figure 4.9: Subproblem solution column and the mining plans generated per period	75
Figure 4.10: Subproblem solution column on the left is split into destination segments on the right in order to represent the original problem variable <i>zbd</i> node structure in the master problem	76

Figure 4.11: Current state of the partitions after appending the v_3 solution column to the previous set of partitions. At this stage v_3 is not orthogonal to v_1 and v_2 yet.....	77
Figure 4.12: New set of partitions after the orthogonalization process	78
Figure 4.13: Flowchart of the solution methodology. The blue colored boxes represent the original BZ algorithm steps, the red colored boxes illustrate the steps of the new integer solution algorithm	84
Figure 4.14: Cross section of a polytope illustrating the solution space of the new integer solution algorithm	85
Figure 4.15: Orthogonal partitions of C_m and C_n on a plane.....	87
Figure 4.16: Primal-dual relationship of the master problem.....	88
Figure 4.17: Solution space of the original problem and the final set of partitions.....	95
Figure 4.18: Classification of the blocks into ore and waste	95
Figure 4.19: Undiscounted economic values of the blocks.	96
Figure 4.20: Production scheduling requirements.	96
Figure 4.21: Iteration 1- Integer feasible mine plan for period 1 with penalized values.....	99
Figure 4.22: Iteration 1- Integer feasible mine plan for period 2 with penalized values.....	100
Figure 4.23: Iteration 1- Integer feasible mine plan for period 3 with penalized values.....	100
Figure 4.24: Iteration 1- Integer feasible mine plan for period 3 with penalized values.....	101
Figure 4.25: Iteration 1- Integer feasible pits per period. Period 1 pit is shown with yellow, period 2 pit is shown with green and period 3 pits are shown with blue and pink colors	102
Figure 4.26: Iteration 1- Initial set of integer feasible orthogonal partitions. Colors represent the blocks considered in the partition.....	103
Figure 4.27: Iteration 2- Mine plan generated by solving the subproblem shown by yellow color.....	106
Figure 4.28: Iteration 2- Period 2 pit is split into integer feasible mine plans where each color represents a pit.....	107
Figure 4.29: Iteration 2- Subproblem solution column on the left is split into integer feasible columns on the right.....	108

Figure 4.30: Iteration 2- Current state of the partitions after appending the integer feasible columns h to the previous set of partitions v . At this stage the columns h are not orthogonal to v yet	109
Figure 4.31: Iteration 2- Integer feasible orthogonal partitions. Colors represent the blocks considered in the partition.....	110
Figure 4.32: Iteration 2- Integer feasible orthogonal partitions presented as pits per period	111
Figure 4.33: Iteration 3- Mine plan generated by solving the subproblem. Period 1 pit is shown with yellow color and period 2 pit is shown with blue color	114
Figure 4.34: Integer feasible mine plan for period 2 with penalized values.....	115
Figure 4.35: Integer feasible mine plan for period 2 with penalized values.....	115
Figure 4.36: Iteration 2- Subproblem mine plan is split into integer feasible mine plans where each color represents a pit.....	116
Figure 4.37: Iteration 3- Subproblem solution column on the left is split into integer feasible columns on the right.....	117
Figure 4.38: Iteration 3- Current state of the partitions after appending the integer feasible columns h to the previous set of partitions v . At this stage the columns h are not orthogonal to v yet	118
Figure 4.39: Iteration 3- Integer feasible orthogonal partitions. Colors represent the blocks considered in the partition.....	119
Figure 4.40: Iteration 3- Integer feasible orthogonal partitions presented as pits per period	120
Figure 4.41: Schedules of the optimal mine plan	123
Figure 5.1: McLaughlin Mine process flow	125
Figure 5.2: East-11075 cross section taken from the ultimate pit.....	128
Figure 5.3: Plan view of the ultimate pit.....	129
Figure 5.4: North-9800 cross section taken from the ultimate pit	130
Figure 5.5: Yearly processed tonnage and the average grade at the mill	132
Figure 5.6: Yearly processed tonnage and the average grade at the leach pads	133
Figure 5.7: Risk behavior of the mine plan over the production years.....	133
Figure 5.8: Yearly production schedules on East-11075 cross section	134

Figure 5.9: Plan view of the yearly production schedules	135
Figure 5.10: Yearly production schedules on North- 9800 cross section	136
Figure 5.11: Comparison of the traditional ultimate pit vs true ultimate pit on a Plan view	136
Figure 5.12: Projection of true ultimate pit on traditional ultimate pit	137
Figure 5.13: Gold Deposit Mine process flow	138
Figure 5.14: East-28000 cross section taken from the ultimate pit	139
Figure 5.15: Plan view of the ultimate pits	140
Figure 5.16: North- 43000 cross section taken from the ultimate pit	141
Figure 5.17: Yearly processed tonnage and the average grade at the mill	143
Figure 5.18: Yearly processed mill and leach tonnage and the average grade at the leach pad .	144
Figure 5.19: Yearly production schedules shown on a Plan View	145
Figure 5.20: Yearly production schedules shown on East-27700 cross section	146
Figure 5.21: Yearly production schedules shown on North-55200 cross section	146
Figure A.1: Potential impacts of slope steepening (Read & Stacey, 2010)	160
Figure A.2: Block dependencies on a directed graph	160
Figure A.3: 1:5:9 block pattern applied to a cubic revenue block model (Khalokakaie, 1999) .	162
Figure A.4: Cone template generated by a) linear and b) non-linear interpolation (Shishvan and Sattarvand, 2012)	165
Figure A.5: Cone template generated for principal and diagonal directions	174
Figure A.6: Cone template generated for the 1 st Quadrant	176
Figure A.7: Final cone pattern template for 40° pit slope angle, 50x50x35ft blocks	176
Figure A.8: Final cone pattern template for 40° pit slope angle, 20x20x15ft blocks	177
Figure A.9: Cone template with principal, diagonal and azimuth directions	179
Figure A.10: Parameters needed for non-linear interpolation (Gilani and Sattarvand, 2015)	181
Figure A.11: Azimuths and slope angles given for each region on the first quadrant. Azimuths shown with red, principal and diagonal directions shown with black color	187

Figure A.12: Successor and predecessor azimuths with α and β angles	189
Figure A.13: The block levels and j locations on principal direction and z locations on diagonal direction shown with similar colors	193
Figure A.14: Cone template generated for principal and diagonal directions	194
Figure A.15: Successor and predecessor radius for coordinate (1,2) with β and α angles.	194
Figure A.16: Cone template generated for the 1st Quadrant	196

LIST OF TABLES

Table 5.1: Economic parameters.....	125
Table 5.2: Summary of the cutoff grade intervals	126
Table 5.3: McLaughlin deposit characteristics	126
Table 5.4: Risk profile of the deposit.....	126
Table 5.5: Summary of blocks in the ultimate pit.....	127
Table 5.6: Ultimate pit risk profile	128
Table 5.7: Production requirements for the mine plan	131
Table 5.8: Summary of the results for the generated mine plan	132
Table 5.9: Summary of the results	134
Table 5.10: Summary of the blocks in the ultimate pit.....	139
Table 5.11: Mine plan production scheduling requirements	142
Table 5.12: Initial stockpile parameters.....	142
Table 5.13: Summary of the mine production scheduling results	143
Table 5.14: Summary of the results	144
Table 5.15: Summary of the results for scheduling the McLaughlin Deposit with predetermined destinations together with the set of constraints enforced	148
Table 5.16: Summary of the results for scheduling the McLaughlin Deposit with dynamic cutoff grade strategy and the regarding set of constraints.....	149
Table 5.17: Summary of the results for scheduling the multi pit Gold Deposit subject to 17 slope zones and the corresponding set of constraints	149
Table A.1: Summary of the cone radius and the distance from the center of the mass of a block calculated for the principal and diagonal directions for 8 increments	170
Table A.2: Candidates for the end pattern levels and j locations on principal direction shown with similar colors.....	171

Table A.3: The best candidate end pattern selected on principal direction based on the minimum percent deviation.....	171
Table A.4: Candidates for the end pattern levels and z locations on diagonal direction shown with similar colors.....	172
Table A.5: The best candidate end pattern selected on diagonal direction based on the minimum percent deviation.....	172
Table A.6: The block levels and j locations on principal direction shown with similar colors..	173
Table A.7: The block levels and z locations on diagonal direction shown with similar colors..	174
Table A.8: The blocks that are included in the cone template colored with blue.....	175
Table A.9: Cone radius calculated for each region.....	187
Table A.10: Summary of the distance from the center of the mass of a block calculated for the principal and diagonal directions for 8 increments	188
Table A.11: Summary of the radius calculated for each level on each principal and diagonal direction.....	189
Table A.12: Start pattern level selected for each direction. Yellow color shows the radius on the principal directions greater than CMP_j , blue color shows the radius on the diagonal directions greater than CMD_z	190
Table A.13: Candidates for the end pattern levels and j locations on principal and diagonal directions shown with similar colors.....	191
Table A.14: The best candidate end pattern selected on principal and diagonal directions based on the minimum percent deviation.....	192
Table A.15: The blocks that are included in the cone template colored with blue.....	195

ACKNOWLEDGMENTS

I would like to thank all of the significant people that made this PhD a success. First of all, I wish to express my deepest appreciation to my thesis advisor Dr. Kadri Dagdelen who provided continuous encouragement and support throughout this journey. He was not just an academic advisor; he was also a life mentor and has been the biggest influence in my career. Without his support, this endeavor would not have become a success. Next, I would like to thank Dr. Thys Johnson whose brilliance was an inspiration for me to overcome the challenges that I frequently encountered throughout this process. His friendship and his continuous support have been priceless. I sincerely thank Dr. Marcelo Godoy, for making this research possible with his contributions and his financial support. This research was funded by Newmont Mining Corporation and I will be forever grateful for their support.

Special thanks are also given to Dr. Priscilla Nelson for her financial support in difficult times. Many thanks to Dr. Christopher Painter Wakefield for being there for me whenever I needed guidance with coding. I also wish to thank Dr. Hugh Miller for his time and effort in being a part of the committee. I would like to also acknowledge the financial support provided by Larry Allen in the latter semester.

I feel privileged to be surrounded by smart people at Mines. I would like to thank Onur Golbasi, Marion Nicco and Marko Visnjic for their wonderful fellowship and always being there to support me in times of need. I wish to thank my colleague Ady Van Dunem, who has always been a brother to me. We have shared countless adventures during this period.

I would like to take the opportunity to express my gratitude to my parents in Turkey for their motivation, friendship and financial support.

Lastly, I am forever grateful to my wife, Anna, for her love, constant encouragement, support and understanding.

To my parents and my wife

CHAPTER 1.

INTRODUCTION

The strategic open pit mine production scheduling problem is a large-scale integer programming problem which requires a solution on a block by block decision basis to determine which blocks should be extracted, when they should be extracted, and what to do with the blocks once they are extracted. Researchers have been trying to solve this problem since 1960`s but so far, it has not been possible to find an optimal integer solution when it is modeled with mining and mill capacity, blending and stockpile constraints. Since the size of a mining problem makes the exact integer programming solution techniques inapplicable, many attempts have been made to obtain a solution with numerous techniques such as pushback designs, heuristic and aggregation methods which cannot be proven to converge to the true optimal solution. Also, the methods that can find the closest integer optimal solutions to the models with a block by block basis can only solve the problems constrained with upper and lower bound mining capacity constraints. Linear programming (LP) relaxation of real sized mine planning problems can be solved to a proven optimality by applying existing decomposition algorithms. However, the mining decisions obtained from the LP relaxation are usually fractional and cannot be implemented practically. A new integer solution algorithm will be introduced in this dissertation that can solve mine production scheduling problems modeled with multi capacities, grade blending, grade uncertainty, stockpiles, variable pit slopes and multi destinations. An integer solution to this complex scheduling problem has never been achieved or at least never been reported in the literature.

1.1 Need for A Mine Production Scheduling Optimization

Mine production can be defined as a complex system formed by its elements which are the operational mining system and the blocks to be mined and functions with a goal of maximizing the discounted cash flows of the mining operations. The boundary within which the system functions is expressed by an ultimate pit which contains all the blocks that are profitable to be mined considering only pit slope constraints. This assumes all the resource capacities are unlimited, the mills are free from any kind of blending requirements and the time value of money

is ignored. The actual mine production is free of such unrealistic assumptions, therefore understanding the interactions between the mining blocks within this system becomes significant to achieve the objective of maximizing the profit. The factors that delineate these interactions between the blocks and the operational mining system can be commodity price, mining cost, processing cost, grade of the blocks, mineral resource classification categories, cycle times, material type, tonnages mined and processed in a year, total available truck hours, certain blending requirements depending on the material type, multiple processing destinations with different ore recoveries, stockpiles, pit slope angles based on the rock formations, cutoff grades and time value of money. Traditionally, some assumptions are made to reduce the complexity of the system.

One of the common assumptions is, designating the blocks to the most profitable destination which neglects the factors influencing the block interactions. The decision based on the value at the destination will underestimate the potential profit. It can be illustrated as follows. Let us assume there are two possible processing destinations for an ore block, and that only one more ore block can be sent to a destination where the highest recovery can be achieved. If there are two ore blocks where the lower grade ore block is overlaying the higher-grade ore block, sending the lower grade ore to a destination where less ounces will be recovered will create an opportunity for a high-grade ore to be processed at a destination where more ounces could be recovered. Another common assumption is made based on cycle times. If there are multiple waste dumps to send a waste block, the operator makes a choice based on the lowest cycle time. The fact is waste dumps have limited capacities, therefore if a waste block is required to be mined, the optimizer should be able to send it to an available waste dump at the time of extraction. Otherwise, the plan generated by an optimizer will be biased on the preselected destinations which will again underestimate the potential profit.

These kinds of operational decisions always neglect the factors that play a role on the block interactions which may end up costing millions of dollars. The complexity of the mine production system also pushed the operators to come up with plans by aggregating the blocks into benches and scheduling these benches within the pushbacks where the block level resolution is diminished. But this approach will inherit all the disadvantages of the pushback design. Furthermore, homogenizing the blocks on a single bench will lead to a loss of information on a block level. Although some bench aggregation techniques may allow individual blocks to be mined from a

given bench, restrictions on mining a block before mining the whole bench above reduces the operational flexibility to meet with the blending requirements as well as other production requirements. All these assumptions are done with a goal of quickly generating mine production plans by ignoring the critical relationship between the mentioned factors and the blocks; which results with poorly developed mine plans that does not meet with the production and processing requirements and may also end up with low profit margins. The only way to generate a mine production plan that can incorporate all the factors characterizing the interactions between the blocks simultaneously is mathematically modeling the mine production system as an optimization problem shown by Johnson in 1968 and solving for an optimal integer solution. So far, nobody was able to provide an optimal integer solution to a mine production scheduling problem on a block by block level constrained by production and processing capacities, grade blending constraints, risk blending constraints, multi destinations, variable pit slope angles and stockpiles.

1.2 Scope of Work

The linear programming (LP) relaxation of a mine production scheduling problem can be solved fast to a mathematically proven optimality by decomposition algorithms. In this dissertation, the Bienstock-Zuckerberg (BZ) decomposition algorithm is implemented to find the optimal LP solution. Since the BZ algorithm generates fractional block mining decisions, the results cannot be interpreted by any practical means, however the objective function value will always remain as an upper bound to the mine production scheduling problem. If the optimal integer solution is not known, the success of an integer feasible solution can always be determined by solving the LP relaxation of the same problem and measuring the gap from the LP optimal solution.

The proposed integer solution algorithm will benefit from the mathematical theories behind the decomposition algorithms to generate the closest optimal integer solution to the LP optimal solution. Instead of using a linear combination of the extreme solution vectors obtained from the consecutive sub-problem solutions in a decomposition algorithm, the proposed solution algorithm will replace them with the convex combination of the integer feasible solution vectors to the original problem that will narrow the span of a solution space from the sub-problem solution space to the original problem solution space where the optimal integer solution exists. The proposed solution algorithm should be able to provide an optimal integer solution very close to the LP

optimal solution of any large scale open pit mine production scheduling problems modeled with multi capacities, grade blending, grade uncertainty, stockpiles, variable pit slopes, multi destinations and truck hours. It should be highlighted that the blocks will not have any pre-determined destinations based on either grades or cycle times since the best destination selection per block will be done automatically or as a function of the “state of the system” during the optimization process in order to maximize the NPV. The methodology developed within this research will also allow the managers to incorporate their risk tolerances within the mine production scheduling problem. The solution algorithm that can generate an optimal integer solution to a mine production scheduling problem that has never been solved will be a milestone in operations research.

CHAPTER 2.

LITERATURE REVIEW

In this chapter, an extensive literature review of the mathematical modeling methods together with the solution techniques to the open pit mine production scheduling problems that have been under investigation since the 1960s will be presented. The solution techniques proposed by the researchers are presented under categories characterized by the variable types such as linear programming, mixed integer linear programming and integer programming solution techniques. The proposed solution techniques will be critiqued within each category. The reason for collecting the assessment of the techniques under a separate section is to avoid repeating similar evaluations since most of the techniques share resembling pitfalls. It is also worth mentioning that none of the techniques proposed by the researchers were able to provide an optimal integer solution to the mine production scheduling problems. This is the leading motivation behind the work in this thesis.

2.1 Ultimate Pit Limit Problem

Traditionally, the ultimate pit limit problem only investigates the most profitable blocks to be mined as if all the resource capacities were unlimited and the mills were free from any kind of blending requirements. In other words, the value of the pit is maximized subject only to the extraction sequence of the blocks governed by the pit slopes. Since all the resource capacities are assumed to be unlimited and the mills are assumed to have no blending requirements, the need for a scheduling vanishes as well as the impact of time value of money on block extraction. It is usually accepted as the maximum limits of mining or as the ultimate shape of the pit at the end of mine life. There are various heuristic and exact algorithms that can solve the “ultimate pit problem” in a reasonable amount of time.

The moving cone algorithm of Pana (1965) was a widely accepted and implemented heuristic algorithm due to its fast solution time. Dagdelen (1985) shows that the algorithm generates cones with a vertex positioned on a positive block and moving from one positive block to another. The side walls of the cone obey the pit slope and the profitable cones are selected for mining. Gauthier and Gray (1971), Barnes (1982) and Underwood (1996) mention that the moving

cone method has shortcomings since the cones are only targeting a single positive block and ignoring the combination of the other cones. Hence, if a single cone is not profitable, it will not be mined although it would generate profit once combined with the other cones. Since the moving cone method cannot promise the optimal answer, it falls into the heuristic category.

In 1965, an algorithm based on graph theory concepts was introduced by Lerchs and Grossman which is currently known as the LG algorithm. The pit is initially represented with a graph by converting each block to a node with a node mass equal to the block value and connecting all the positive nodes (ore blocks) and the negative nodes (waste blocks) to the root node with the arcs. In other words, an initial spanning tree is formed. Spanning trees will be partitioned into strong and weak node groups. If the group of nodes represented with an arc coming from a root node have a cumulative node value or mass that is positive, the group of nodes is called strong otherwise they are called weak nodes. If the weak group of nodes overlies a strong group of nodes, they are combined and reclassified into strong and weak group of nodes based on their cumulative mass. The collection of strong nodes forms the maximum closure which indeed represents the set of blocks that maximizes the profit and defines the ultimate pit limit.

Johnson (1968) is the first to show that when the ultimate pit limit problem is formulated as a LP, the dual of the LP model can be transformed to a max flow network model. The author also shows that the LP representation of an ultimate pit limit problem has a totally unimodular structure where the entries of the coefficient matrix are integral, and the determinant of the coefficient matrix is either 1 or -1, hence the problem can be converted to a bipartite graph. The author became a pioneer in the field by showing that an ultimate pit limit problem is a maximum flow problem which can be solved by any algorithm that can solve max flow network models.

Zhao and Kim (1992) modified the LG algorithm and extended its application to the large block models. The pit is represented with a directed graph where the arcs are formed between the positive (ore) and negative (waste) nodes (blocks) to imply the sequencing relationships. The positive nodes are connected to negative nodes which is called as full support. Also, if an ore block cannot support the waste block, then an arc is directed from waste block to the ore block called as partial support. The value of the root node which is at the starting point of the last generated arc is checked to determine if the tree should be categorized as strong or weak. If a weak group of nodes overlies a strong group of nodes and only non-root nodes of the strong and weak trees have a

precedence relation, the root node of strong tree is shifted to the non-root node of the strong tree and the root node of weak tree is shifted to the non-root node of the weak tree since the arcs can only be drawn between the root nodes. This approach solves the jointly support and reallocation problems. Also, Zhao (1992) explains in his thesis that the proposed approach avoids the normalization step of the LG algorithm which will significantly reduce the computation time. However, a direct comparison with the original LG algorithm was not provided.

Yegulalp and Aries (1992) applied the excess-scaling algorithm of Ahuja and Orlin (1989) to solve the ultimate pit limit problem as a max flow problem. However, the authors showed that the LG algorithm implemented by Whittle Programming Pty. Ltd solves the ultimate pit limit problem faster than their implementation of the excess-scaling algorithm.

Underwood and Tolwinski (1996) interpreted the graph theoretic methodology of LG algorithm from a mathematic programming point of view by examining the similarity of the steps of LG algorithm with the steps of dual simplex algorithm. The authors show that the dual simplex algorithm maintains the same logic as the LG algorithm by removing the strong nodes and leaving the weak nodes at every stage and once the optimality is achieved, the strong nodes will represent the ultimate pit limit solution. The authors claim that the only difference between the two algorithms is that the dual simplex algorithm updates the value of the arc between strong and weak node by maximizing its value while maintaining the dual feasibility which improves the computing performance. The authors show that dual simplex algorithm solves the ultimate pit problem a few minutes faster than the LG algorithm for the cases they tested with up to 2.5 million blocks.

Hochbaum (2008) introduced the pseudoflow algorithm which is the fastest known algorithm that can solve the ultimate pit limit problem or multi period sequencing problem as a maximum flow or minimum s-t cut problem. The algorithm may start with a simple initialization by saturating the sink adjacent and the source adjacent arcs and keeping the rest of the arcs with zero flow (Chandran and Hochbaum, 2009). Hence, excesses are created at the source adjacent nodes and deficits are created at the sink adjacent nodes. The method tries to reroute flow through all the arcs going from a subset of nodes that has excess to a subset of nodes that has a deficit, so that in the end a provable minimum cut in the graph will be achieved (Hochbaum, 2008). According to the computational study of the pseudoflow and push-relabel algorithms for the maximum flow problem conducted by Chandran and Hochbaum in 2009, the pseudoflow

algorithm is faster than the push-relabel algorithm which was widely accepted as the fastest algorithm to solve the maximum flow problem at that time. It has been also shown that the pseudoflow algorithm solving the max flow formulation of the ultimate pit limit problem is faster than the Lerchs and Grossman algorithm.

2.2 Pushback Design

The pushback design method determining a production schedule to an open pit mine should start with determining the ultimate pit limits. Once the ultimate pit limit is determined, mine planning continues with the goal of finding the optimal extraction sequence of the blocks, which in the end, results in incremental pits called as pushbacks. The need for dividing the pit into sub pits arises due to the scale of a block by block production scheduling problem hence, sub-pits or pushbacks are used to schedule the blocks quarterly or yearly. Many heuristic techniques were developed in order to sequence the pushbacks so that the schedules from the pushbacks will be in compliance with the resource capacity and mill blending requirements.

Dagdelen and Francois-Bongarcon (1982) determined a series of pushbacks by varying the price of the commodity, cutoff grade, mining or processing costs.

Gershon (1987) proposed an algorithm which generates cones with the shape of a pit expanding towards the bottom of the pit with the vertex positioned on each block. The total quality of the ore within the cone determines the positional weight of the block. The blocks are ranked based on their positional weight and the highest rank block is scheduled first. Then, the positional weights of the blocks are re-initialized with the remaining blocks. This allows scheduler to reach the high-grade ore at the bottom of the pit quicker than the traditional approaches.

Whittle's approach (1988) is a pit parametrization technique which varies the block values incrementally and generates nested pits by implementing the LG algorithm.

Seymour (1995) proposed a parameterization algorithm that abandons the traditional pit parametrization techniques which generate pits as a function of pit value. Instead, the method selects both the pit volume and pit value as the parameters of this function. The algorithm is a modified version of the LG algorithm by adding the parametrized variables and allowing to form

sub trees that represent the small pits (Meagher *et al.*, 2014). The branches are categorized as strong or weak based on the threshold value of their strength which is calculated by dividing the cumulative value of the nodes in the branch by the cumulative mass of the nodes in the branch. Strong branches form the sub tress (small pits) which are sequenced based on their strength value.

Ramazan and Dagdelen (1998) developed a minimum stripping ratio pushback design algorithm which is a modified version of the Seymour`s algorithm in 1995. The authors apply break-even cutoff grade to differentiate ore and waste with an indicator value “1” assigned to ore and indicator value “0” assigned to waste. The authors replaced the cumulative value approach in Seymour`s algorithm with the cumulative indicator value approach to calculate the strength of the branch. The goal of the algorithm is to find pits with the minimum stripping ratio which will lead to a schedule where the ore blocks are mined as soon as possible. Authors also demonstrated that Whittle`s pit parametrization method may produce the same nested pits as the minimum stripping ratio pushback design algorithm if all the ore blocks are assigned the value of the highest-grade ore in the block model.

Somrit (2011) introduced a phase design algorithm which uses Lagrange parameters to determine the size of the pits in compliance with the annual resource capacity requirements. The usage of Lagrange parameters is similar to the Whittle`s pit parametrization technique. Although the relationship between the Lagrange parameters and pit size is not linear, the author uses linear interpolation to determine the parameters since it is impossible to predict the relationship with an equation. This kind of approach may result in gap problems. The pits are determined in a reverse order compared to the commonly adopted approaches. The algorithm first tries to find a pit that meets the resource requirements of “t-1” periods. The unmined portion of the pit becomes a phase for the period “t”. Then, the algorithm looks for the next pit that obeys the resource requirements of “t-2” periods. The remaining blocks create the next phase for the period “t-1”. Hence, the algorithm proceeds backwards in time till it determines the phase corresponding to the first period.

2.2.1 Shortcomings of the Pushback Design Methods

Due to the large scale of block by block mine production scheduling problems, the pushback concept became attractive since it allows scheduling to be done from the sub-pits or

nested pits which results in less variables in the optimization model. Dagdelen and Francois-Bongarcon (1982), Gershon (1986), Whittle (1988), Seymour (1995), Ramazan and Dagdelen (1998), Somrit (2011) proposed various techniques to obtain the pushbacks. The methods follow the incremental fashion of obtaining pushbacks that fail to incorporate the blending requirements as well as the uncertainty of the grade of ore. Moreover, the size of the pit has a non-linear relationship with the value of the pit which makes it extremely hard to determine. But the common approach attempts to find the pit size with an assumption of a linear relationship which in the end results with a gap problem. The resulting gap problem will prevent pushbacks to meet with the capacity requirements. So far, researchers have not found an approach to overcome the stated problems which will in the end produce poorly designed pushbacks. Since many production schedule plans highly depend on the design of pushbacks, poor designs will prevent the schedules from achieving a maximum NPV or even obtaining a feasible solution.

2.3 Mathematical Programming and Integer Solution Techniques for Production Scheduling

The mine production scheduling problem was originally formulated by Johnson (1968) with a goal of finding the optimal block schedule that will maximize the NPV throughout the life of a mine. This generic model incorporates the dynamic cutoff grade strategy by allowing the blocks to be sent to the most attractive destination determined by the pit slope, capacity limitations and average grade requirements at each destination and availability of the equipment. Johnson's model will generate a higher NPV since the destinations of the scheduled blocks will not be pre-determined by the mine planner, instead the optimal destinations will be selected by the scheduler to maximize the NPV. The best schedule would be obtained if the model could be solved with integer decision variables. The complexity of the problem increases as the number of the blocks, which are the decision variables in the model increases. A true integer optimal solution of the proposed model still cannot be determined with the current solution techniques. The past and most recent research in the area is geared towards developing solution techniques which can find such feasible solutions that are close to the optimal integer solutions within a reasonable time frame. Since the optimal integer solution to the proposed model is unknown, the solution techniques developed cannot identify how far one is from the integer optimal solution, however the quality of

the solutions are measured from the optimal linear programming solution since it provides an upper bound (Dagdelen, 1985).

2.3.1 Linear Programming Solution Techniques

Linear Programming (LP) models are often called linear relaxation models, if one decides to relax the integrality of the decision variables by presenting them as continuous variables in the model. As mentioned before, the theoretical upper bound provided by the optimal LP solution is a benchmark to measure the “optimality gap” of the integer solution to the same problem which allows the evaluation of the success of the integer solutions.

G. Dantzig proposed the simplex algorithm in 1947, which is an exact optimization technique to solve LP models. The simplex algorithm still remains one of the most popular algorithm, adapted by commonly used optimization engines such as CPLEX and Gurobi. Once the LP model is generated, the solution space bounded with the constraints creates a polytope where the optimal solution may exist on one of its vertices which can be also called extreme points. Given a mine production scheduling problem with an objective of maximizing NPV, the simplex algorithm will solve the linear relaxation of the model by starting the search of an optimal solution on an extreme point and moving to another extreme point until the objective function value cannot be improved. This kind of a search is not practical for mining problems since the number of variables and constraints may result with too many extreme points to be searched which results in inefficient solution times.

Johnson (1965) proposed the Dantzig Wolfe (DW) decomposition algorithm to solve the LP relaxation of the mine production scheduling model by decomposing the original model into subproblems. The subproblem can be called a Lagrange relaxation of a model where the resource capacity and blending constraints are moved to the objective function and penalized with the associated dual values. The Lagrange relaxation problem is similar to the ultimate pit problem except the blocks are sequenced in multi time period. Since the subproblem or Lagrange relaxation problem has the totally unimodular structure, Dagdelen (1985) showed that the multi period subproblem is a maximum flow network problem which is faster than solving the problem as a LP. The DW algorithm uses a convex combination of the extreme points of the Lagrange relaxation

problem to find the optimal solution to the original problem. The variables associated with the extreme point solution vectors of the subproblem are used to solve the master problem subject to the constraints of the original problem and a constraint that assures the convex combination of extreme points. The optimal solution which may be in a fractional form can be used as a guidance for scheduling the blocks since the optimal LP solution is an upper bound for the optimal integer programming (IP) solution.

Bienstock and Zuckerberg (2009) improved the DW decomposition algorithm and the BZ is presently recognized for being the fastest LP solving decomposition algorithm for the precedence constrained production scheduling problems such as mine production scheduling problems. The BZ algorithm can solve the LP relaxation of the mine production scheduling problems constrained with upper and lower bound resource and blending constraints to a proven optimality. The algorithm decomposes the original problem into a subproblem and master problem. Each subproblem is solved likewise in the DW algorithm except the extreme point solution vectors of the subproblem are orthogonalized in the BZ algorithm which will increase the dimension of the solution space by adding more axes to the system which eventually reduces the number of vectors linearly combined to represent the optimal solution in the master (original) problem. In other words, since there is a variable associated with each one of the vectors generated from the subproblem solution, a decrease in the number of vectors will result with less variables to solve in the master problem. Detailed investigation of the BZ algorithm is presented in Chapter 4.

Van Dunem (2016) integrated the uncertainty concept in the form of risk constraints to the mine production scheduling problem. The author developed a model which allows the user to reflect his/her risk tolerance within the production scheduling problem. The model consists of mill capacity, mill blending and risk constraints where the mill blending, and risk constraints have both minimum and maximum requirements. The resource modeling is done in order to estimate the grade of a block with an associated level of uncertainty. Mineral resources are classified into the categories as inferred, indicated and measured and defined as follows (CIM, 2006):

(i) An Inferred mineral resource is that part of a Mineral Resource for which quantity and grade or quality are estimated on the basis of limited geological evidence and sampling. Geological evidence is sufficient to imply but not verify geological and grade or quality continuity.

(ii) An Indicated mineral resource is that part of a mineral resource for which quantity, grade or quality, densities, shape and physical characteristics are estimated with sufficient confidence to allow the application of modifying factors in sufficient detail to support mine planning and evaluation of the economic viability of the deposit.

(iii) A Measured mineral resource is that part of a mineral resource for which quantity, grade or quality, densities, shape, and physical characteristics are estimated with confidence sufficient to allow the application of modifying factors to support detailed mine planning and final evaluation of the economic viability of the deposit.

The risk constraints are developed in order to control the uncertainty by applying an upper bound on the number of the inferred blocks sent to the mill and lower bounds on the amount of indicated and measured blocks. With the incorporation of the risk constraints the author aims to minimize the impact of uncertainty on meeting the operational or production targets. Traditional stochastic production scheduling problems incorporate the grade uncertainty by varying the grade of the ore blocks from one scenario to another which eventually increases the number variables with the magnitude of the number of scenarios and results in a NP-hard problem with a huge variable space. In contrary, the author proposed a novel approach by integrating the grade simulations once the production scheduling is completed. In other words, the initial schedule is generated in order to meet with the risk tolerance of the manager. Then, the simulated grades are integrated to check whether the given schedule will meet the blending requirements. For example, if 9 out of 10 integrated simulations are within -15% of the lower bound and +15% of the upper bound blending requirements, then the optimal production schedule is obtained under the grade uncertainty. If the integrated simulations fail to meet with the blending requirements, the manager can lower the number of the inferred blocks requested by the mill or increase the number of measured blocks which will potentially lower the grade variability at the mill. Also, the management can drill more drill holes to increase the level of confidence within the inferred group of blocks. The NPV difference between the riskiest scenario and the considerably less risky scenario can give a hint to

the management about the expected NPV that can be generated if more drilling is conducted on the inferred group of blocks. Although modeling the production scheduling problem as a deterministic model reduces the number variables significantly, the problem size is still considerably large to solve as an integer programming model. Moreover, a solution for the LP relaxation of the model might not be obtained by using CPLEX. Thus, the author proposed a decomposition method developed by Bienstock and Zuckerberg in 2009 to solve the LP relaxation of the model. The integration of the Pseudoflow algorithm to the BZ decomposition algorithm made the LP relaxation of the large-scale open pit production scheduling problem under the metal uncertainty solvable with fast solution times.

2.3.2 Mixed Integer Linear Programming Solution Techniques

Mixed integer linear programming techniques gained importance due to the size of the mine production scheduling problems. Since the optimal integer solution of the problem has not been achieved, a lot of research has moved towards solving the mining problems with the mix of continuous and integer variables.

Gershon (1983) modified the mine production scheduling model proposed by Johnson (1968) by adding continuous variables that will mine the blocks partially if all the preceding blocks have been removed. The author claims that forcing the integrality of the decision variables will not reflect the practical approach, hence by allowing partial mining, the schedules can meet with the blending constraints. Osanloo *et al.* (2008) mentioned that Gershon`s approach will not be able to handle large problems due to the large number of the binary variables and also since the size of the problem is the main concern, dynamic cutoff grade strategy cannot be implemented.

Ramazan and Dimitrakopoulos (2004) improved Gershon`s (1983) model by setting the ore blocks as binary variables and leaving the waste blocks as continuous variables. The decision to partially mine the waste blocks still exist in the model. The authors claim that since the ore blocks must be mined fully, the waste blocks will also be mined fully in order to satisfy the sequencing requirements. Partially mined blocks may exist in the last period. The case study lead to a conclusion that the number of binary variables is significantly decreased as well as the solution

time by setting only the ore blocks as binary and also the partially mined blocks can be minimized if the reserve constraints are equalities.

Kawahata (2006) took a unique approach to solve the MILP problems. The author divided the pit into series of pushbacks and divided the pushbacks into increments (benches). The blocks within the increments aggregated based on the similarities of the grades of the blocks and rock properties. The original MILP formulation consists of binary variables that preserve the sequencing between the pushbacks and increments and continuous variables which allow the scheduler to pick the destination for the portion of the increment mined under dynamic cutoff grade policy. The author reduces the solution space that the MILP model works within by splitting the original problem into two Lagrange relaxation problems where the aggressive case relaxes the process capacity constraints while keeping the mining capacity constraints and the conservative case relaxes the mining capacity constraints while maintaining the process capacity constraints in the model. Since the Lagrange relaxation model consists of only small number of pushback variables achieved by aggregating the benches, the binary nature of the variables does not pose any computational disadvantage. Since the solution of the subproblems will provide an upper and lower bound to the original MILP problem, the variables which are not considered in the solution set of the subproblems can be eliminated from the variable set of the MILP problem. This unique approach will decrease the solution time of the MILP problem significantly.

Boland *et al.* (2008) formulated the MILP problem with the same aggregation technique presented by Kawahata (2006). The author used binary variables to sequence the aggregated blocks, however the model has two continuous variables, one for the fraction of the aggregate excavated and the other variable represents the fraction of the aggregate processed. The author claims that the blocks in any aggregate can have multiple attributes based on the different metal content. The model does not have any blending constraints and the blocks in the aggregates are already classified as ore and waste with a predetermined cutoff grade.

Goycoolea *et al.* (2015) proposed a MILP model which works with pre-defined pushbacks. As in Kawahata's approach (2006), the author aggregated the blocks within each pushback into benches (increments). The model considers upper bound capacity on both production and processing constraints. Also, a dynamic cutoff grade strategy was incorporated by allowing the

scheduler to choose a destination for the mined blocks. Continuous variables were used to mine the increments and the blocks from each increment partially. The author added additional volume-flow constraints to limit the variance on the production from one period to another. Also, instead of using binary variables to sequence the increments, the author achieved the same outcome with a constraint that required all predecessor increments of any increment which is partially mined to be mined completely.

King (2016) formulated the open pit to underground transition model with pre-determined block destinations by varying crown and sill pillar locations. The author took a phase design approach with the benches binned into blocks characterized by grades. The author combined the surface mine production scheduling model with the underground mine production scheduling model and called it an enhanced transition model. Binary variables were used to schedule the underground activities such as extraction, backfilling and development as well as the sequencing relationship between the benches for the surface mining. Also, continuous variables were used to partially mine the benches and partially send the blocks to the mill. However, the partially bench mining constraints were written in such a fashion that once the bench is mined it should be mined completely so that the sequencing can take place. The author also allowed the stockpiles in the model, however the stockpile rehandling costs are not considered. In order to solve the model, the author used the BZ algorithm to solve the LP relaxation of the model and then a rounding heuristic method is applied in order to get an integer solution. The author solved the model many times by changing the locations of the sill and crown pillars in order to find the best location that will maximize the NPV.

2.3.2.1 Shortcomings of the Mixed Integer Linear Programming Solution Techniques

Mixed integer linear programming (MILP) techniques gained importance since the optimal integer solution of the mine production scheduling problem has not been achieved due to the size of the problems. Gershon (1983), Ramazan and Dimitrikapolus (2004), Kawahata (2006), Boland (2008), Goycoolea *et al.* (2015), King (2016) solved the mine production scheduling problems with the various forms of MILP techniques. Kawahata (2006) and Boland *et al.* (2008) further aggregated the blocks into the benches in their MILP models in order to reduce the number of

variables. MILP models are solved for binary variables and continuous variables. Most of the authors use binary variables for the sequencing relations between the benches or blocks, and continuous variables to partially mine the benches or the waste blocks. The main disadvantage of aggregation techniques is the loss of resolution. In other words, once the blocks are aggregated into benches, individual block grades are replaced with the average grade of the blocks like if the block grades on a bench are distributed homogeneously. This assumption might generate schedules which can easily mislead the operations in terms of meeting blending requirements. However, the MILP model proposed by Goycoolea *et al.* (2015) may overcome this issue since the model allows blocks to be scheduled individually from the benches but the sequencing relations are still preserved between the benches which will create a dependency between the block and the bench above. The plans generated with this approach might lead the scheduler to mine blocks which do not have any spatial correlation with blocks on the next level. This might lead to extra stripping for a particular period in certain situations. For the MILP models where the scheduling is done block by block and no bench aggregation is considered, the number of binary variables may be too large which will prevent the solver to find an optimal solution in a reasonable amount of time (Less than 8 hours). Also, there is no real mining case study presented in the literature which can be solved by modeling as MILP that takes into account mining and processing capacity and blending requirements with scheduling on a block by block basis.

2.3.3 Integer Solution Techniques

True open pit mine planning schedules can only be achieved if one can solve Johnson's (1968) production scheduling model with integer variables.

Dagdelen (1985) was the first to show that the Lagrange relaxation technique can be applied to solve Johnson's model to generate integer solutions feasible to the original problem. The Lagrange relaxation model has the same objective function as the original model with an addition of side constraints penalized with the Lagrange multipliers and the model is subject to the sequencing constraints. Hence, the Lagrange relaxation problem solution always provides an upper bound to the optimal solution of the original problem. The goal is to find the best multipliers that will generate solutions as close as possible to the original problem. Dagdelen proposed the subgradient optimization method to generate Lagrange multipliers. If one cannot find such

Lagrange multipliers that will produce feasible solution, to the original problem, it can be concluded that the condition of gap exists (Everett, 1963). Dagdelen identified the totally unimodular structure of the Lagrange relaxation problem and represented the multi-time period sequencing problem on a network structure. The author solved the multi-time period sequencing problem as a combination of a single time period maximum flow network problem.

Tachefine and Sumois (1996) also applied the Lagrange relaxation technique to solve the open pit mine production scheduling problem which does not have any blending constraints. The author uses the Bundle method to obtain the Lagrange multipliers. In order to make the solution of the Lagrange relaxation feasible to the original problem, the author proposed a greedy heuristic algorithm. The author solves the Lagrange relaxation problem as a maximum closure graph problem. The greedy algorithm tries to find a set of frontier nodes which are candidates for elimination. The nodes are selected for elimination in such a fashion that the violation of the capacity constraints is minimized while keeping the impact on the value of the closure minimum.

Akaike (1999) extended Dagdelen's Lagrange relaxation approach with the 4D network relaxation method. The proposed method solves a Lagrange relaxation problem as a multi time period maximum flow problem. Moreover, the method can handle the cutoff grade strategy and the stockpiles can be incorporated in the model. Since the method tries to find the best Lagrange multipliers with the subgradient method, the gap problem is inevitable if one exists.

Ramazan (2001) proposed the fundamental tree algorithm to solve the problem within the pushbacks. The author removes the side constraints of the original model and represents the problem as a network problem. The network model is formulated as a LP problem. At each iteration, the arcs that have 0 flow are eliminated and a new tree(s) is formed. The iterations terminate if the number of trees at the current iteration is equal to the previous iteration or if all the trees have only 1 positive node. Each fundamental tree can be treated as aggregated blocks that contain certain ore tonnage, waste tonnage and grade. Once the fundamental trees are obtained, the production scheduling model for the pushback can be solved as an IP problem with the resource capacity and blending constraints with a binary variable assigned for each fundamental tree. Although the number of the binary variables in the model can be reduced significantly, large deposits can still have large number of fundamental trees that will make model inefficient.

Amaya *et al.* (2009) implemented a local search heuristic method to improve the initial integer feasible solution of a production scheduling model constrained with an upper bound on mining capacity constraint. Given an initial integer feasible solution, the method iteratively fixes parts of the schedule and the remaining unfixed parts are solved within a binary context to find out if a local improvement exists. If the solution provides an improvement, then the initial solution space is updated. The author presented three strategies to search for an improvement. The cone above strategy is used by first picking a block randomly from the set of blocks which are not considered in the initial integer feasible solution set. Then, the target block and its predecessors are solved as an integer while the rest of the initial feasible solution space is fixed. The periods strategy can be applied by randomly selecting two-time intervals from a given integer feasible solution space and solved for the best solution. The transversal strategy is applied by first defining a distance and height and then searching for the blocks that are not considered in the initial integer feasible solution.

Cullenbine (2011) proposed a sliding time window heuristic algorithm to approximately solve the production scheduling problems with the minimum and maximum resource capacity constraints. The algorithm initially fixes the initial integer feasible solutions for the T1 periods. Then, the exact window T2 is determined where the integer solution will be obtained. Then, for the remaining periods T3, Lagrange relaxation of the model is solved by penalizing the minimum and the maximum resource capacity constraints with the duals obtained from the optimal solution of the LP relaxation of the original IP model.

Chicoisne (2012) proposed a toposort heuristic algorithm which uses the optimal LP solution as a guide to round the fractional solutions to an integer result. Expected extraction time of a block is determined by multiplying the proportion of the block mined with its extraction period. Once the integer feasible solution is found, Amaya's local heuristic technique is applied in order to improve the initial integer feasible solution. The algorithm can only be applied for models with single or two capacity constraints in an upper bound form.

Lamghari and Dimitrakopoulos (2012) proposed the Tabu search meta-heuristic method integrated with two diversification strategies that are long term memory search and variable neighborhood search, to improve the feasible solution domain of the stochastic mine production

scheduling problem. The authors modeled the two-stage stochastic problem by penalizing the upper and lower bound mining capacity constraints in the objective function while preserving the scenario dependent mill capacity and metal content constraints. In order to solve the modified model, the initial integer feasible solution (x_i) is obtained by randomly selecting the blocks from a mineable set of blocks meaning that the blocks should not have any predecessors, or the predecessors are already mined. The random selection continues until the amount of the chosen blocks for each period will be greater than or equal to the average of the upper and lower bounds for the mining capacity constraint. The solution honors the non-stochastic constraints such as mining capacity, reserve and sequencing constraints. The tabu search procedure begins once the initial solution is obtained. For each block in the solution set, schedule times of the closest successor and predecessor blocks are identified. Then, the latest schedule time of the predecessor $e(x_i)$ and the earliest schedule time of the successor $l(x_i)$ is selected. For any block i in x_i , the blocks can be shifted from one period to another between the time intervals of $e(x_i)$ and $l(x_i)$ at each iteration. The block which is scheduled minimum amount of times is selected as a candidate in order to improve the searching activity in the less explored areas. If the solutions from the consecutive iterations stop improving for a certain amount of pre-determined time, then the Tabu search terminates. Then, the author applies diversification strategies to the Tabu search solution space in order to explore the areas which are rarely considered. The first strategy is named as a long-term memory diversification strategy. Since each block may be scheduled multiple times during the Tabu search, the time period that a block has been scheduled for less frequently is stored in a set. Then, a block randomly drawn from the set is scheduled to that time period. The second proposed diversification strategy is the variable neighborhood diversification strategy. Given the solution set obtained by Tabu search, a block which is scheduled the least number of times is selected from the set and rescheduled to a time period between $e(x_i)$ and $l(x_i)$ which gives the best NPV. The iterations terminate if the Tabu search solution cannot be improved or whenever the pre-defined number of iterations is reached.

Lambert (2012) proposed the Tailored Lagrange relaxation algorithm to solve the open pit mine production scheduling problems with minimum and maximum resource capacity constraints. The author proposed a maximum value feasible pit (MVFP) algorithm which will generate initial integer feasible solutions to the Tailored Lagrange relaxation model. The MVFP is processed in 3

phases. In Phase 1, a pit parametrization technique is used in order to find ore blocks that will satisfy total processing requirements over the entire time horizon. If the Phase 1 cannot find any pit that satisfies both the minimum and maximum ore production requirements over the time horizon, the well-known gap problem exists. In Phase 2, if the minimum production requirements are not satisfied, an integer program is solved to expand the pit. For the phase 3, an ultimate pit limit approach can be conducted or sliding time window algorithm can be applied to generate an initial integer feasible solution. Once the three phases are completed, the information collected during the MVFP stage will determine the dualization strategy of the Tailored Lagrange relaxation model.

Lamghari et al. (2014) proposed a method to generate an initial integer feasible solution under mining and mill capacity constraints with an upper bound and to improve the generated initial integer feasible solution. The author presented an exact approach and greedy heuristics to get an initial integer feasible solution. For the exact approach, the production scheduling model with mining and mill capacity constraints is solved within a binary context per period as a single time period problem. The author also benefits from the variable elimination techniques by pre-checking the mining capacity violations and generating a valid inequality to strengthen the formulation. For the sequential greedy heuristics, a cone is generated with a vertex on each block with a purpose of finding the set of blocks which will minimize the deviations from the mining and mill capacity requirements while maximizing the NPV. One can make a similar analogy by applying the floating cone algorithm to a multi time production scheduling problem to generate initial integer feasible solutions. Then, the author implemented the variable neighborhood descent (VND) search method of Hansen and Maladenovic (2001) to improve the initial integer feasible solutions. The algorithm works in three steps. Given the initial integer feasible solution, the algorithm first trades the ore and waste blocks between the consecutive time periods. Then, the blocks together with the predecessors are shifted forward and backward in time until no more NPV improvement is achieved.

Lamghari and Dimitrakopoulos (2015) proposed three heuristic algorithms, one for generating an initial integer feasible solution called look-ahead heuristic (LAH) and the others for improving the generated integer feasible solution to maximize the NPV of the two-stage stochastic mine production scheduling problem named a network-flow based heuristic (NF) and a diversified

local search heuristic (DLS). In order to apply the LAH, the multi-time period stochastic model is re-written as a single time period stochastic model. LAH is an improved version of the greedy heuristics presented in Lamghari (2014). LAH generates cones only with the blocks if the objective function value can be improved while obeying the mining constraints. The iterations terminate when there is no such cone that can improve the latest objective function value. Once the initial integer feasible solution is obtained the author applied the NF heuristic to improve the generated solution. The idea of the NF heuristic is to search the initially generated solution space extensively by shifting the blocks from one period to another in order to delay and advance the extraction of the blocks with a purpose of improving the NPV. Also, the NF heuristic has an advantage over the variable neighborhood descent (VND) heuristic introduced by Lamghari (2014) by allowing a search in the solution space instead of just considering the feasible solution space. Since the neighborhoods generated with the forward and backward processes of the NF are large compared to the Tabu Search (TS) presented by Lamghari (2012) and VND neighborhoods, the author converted the problem to a network model and solved the longest path problem (LPP) to find the solution that gives the best improvement in NPV. The author proposed the pulling algorithm of Ahuja (1993) to solve the longest path problem as a network problem. If the solution obtained is not better than the previous one, then the algorithm terminates. Secondly, the author also proposed diversified the local search heuristic (DLS) to improve the initial integer feasible solution. The heuristic DLS is basically the combination of the VND and NF heuristics. Instead of applying the NF heuristic directly to the initial integer feasible solution obtained by LAH, the author first applies VND to the solution so that the NF heuristic will begin with an improved version of the initial integer feasible solution which will also avoid the local optima of the VND heuristic. If the consecutive application of the VND and NF heuristics do not improve the solution, the algorithm terminates, and the current solution is accepted as the final best solution to the problem.

2.3.3.1 Shortcomings of the Integer Solution Techniques

Mining operations can be truly optimized if an optimal integer solution can be found to the production scheduling model proposed by Johnson in 1968. The generic model encompasses block by block scheduling subject to mining and mill capacity and blending constraints while obeying the pit slope requirements. The existing solution algorithms to solve the integer programming models to a proven optimality such as branch and bound (A. H. Land and A. G. Doing, 1960) and

branch and cut (M. Padberg and G.Rinaldi, 1983) are not efficient enough to solve large scale mine planning production scheduling problems within a binary context.

2.3.3.1.1 Lagrange Relaxation Technique

Researchers investigated new solution techniques to solve the mine production scheduling problems with the integer variables. Dagdelen (1985), Tachefine and Sumois (1996), Akaike (1999), Kawahata 2006, Cullenbine (2011), Lambert (2013) implemented various forms of Lagrange relaxation techniques with a goal of achieving an integer solution. If the optimal solution of a Lagrange relaxation problem is feasible to the original problem, it provides an upper bound solution to the original problem. On the other hand, there is no guarantee that a feasible solution to the original problem by adjusting the Lagrange multipliers can be obtained. As shown by Everett in 1963, Lagrange multipliers measure the change in the objective function value when the resource constraints (mining and mill capacity or blending constraints) are expended. Since the relationship between the resource expenditure and the objective function is non-linear, whenever the function has a non-concave region, the hyperplane defined by the Lagrange multiplier cannot be tangent to the accessible points on the region which constitutes the “gap” problem (Everett, 1963). The presence of a gap problem will prevent Lagrange relaxation solution from providing a feasible solution to the original problem. So far, researchers have not identified a method to overcome the gap problems. This will make mine production scheduling problems solved with Lagrange relaxation techniques less attractive since gap problems are not avoidable which appears to be due to the non-homogeneity of a mineral deposit.

2.3.3.1.2 Fundamental Tree

Ramazan (2001) aggregated the blocks into fundamental trees based on only considering the values of the blocks and the precedence relationships. Then, the author treated each tree as an integer variable and solved the mine production scheduling problem under mining and mill capacity and blending constraints. Since, the complexity of the integer programming models is directly related with the number of variables, the author first generated the pushbacks and then scheduled the trees (grouped blocks) from the pushbacks. Unfortunately, since the author uses the aggregation technique together with pushbacks, the weaknesses of those two approaches presented

above will result in poor quality solutions. Moreover, for the large scale open pit mine production scheduling problems, the model will have large number of fundamental trees which will increase the complexity of the integer programming model which might prevent obtaining an integer solution.

2.3.3.1.3 Heuristic Techniques

2.3.3.1.3.1 Heuristic Techniques to Obtain Initial Integer Feasible Solution

Since the scale of mine production scheduling problems restrict the usage of exact solution algorithms, researchers implemented heuristic techniques to find the integer feasible solutions fast and as close as possible to the optimal LP solution. Chicoisne (2012), Lamghari and Dimitrakopoulos (2012, 2015), Lambert (2013), Lamghari *et al.* (2014) proposed methods to obtain initial integer feasible solutions. Among these methods, Lamghari criticized the random heuristic (RH) method proposed by Lamghari and Dimitrakopoulos in 2012 for its myopic approach. Also, the solution times of the improvement heuristics applied to the initial integer feasible solutions generated by the random heuristics are highest according to the cases presented. The author also criticized the greedy heuristic proposed by Lamghari *et al.* (2014) for considering all the unmined blocks as a candidate for the cones generated and selecting cones that will only minimize the deviations from the upper bound mining capacity requirements even if the selected cone does not improve the current solution. The author proposed a look ahead heuristic (LAH) approach which is an improved version of the greedy heuristic where the candidate blocks for the cone are selected if it leads to an improvement of the objective function value for the time period t , while the upper bound mining capacity constraints are not violated. Although the author didn't present the actual solution times that it takes to obtain an integer feasible solution by applying RH and LAH, it is mentioned that LAH is much faster. Also, since the quality of the solutions generated by the improvement heuristics after obtaining an initial integer feasible solution are highly dependent on the initial solution, the case studies show that regardless of which improvement heuristic method is used, the initial integer feasible solutions generated by LAH always results in a better solution compared to the RH and the combined solution times are faster. Lambert proposed a maximum valued feasible pit (MVFP) method to generate an initial integer feasible solution to the model which considers lower and upper bounds on the mining and mill

capacity constraints. According to the case study presented by the author, it takes 5000 seconds (~1hr 20 min) for the MVFP method to generate an initial integer feasible solution for the models with 10000 blocks and a time horizon of 15 years. The solution time is quite high for a fairly small size model. Chicoisne *et al.* (2012) proposed a TopoSort heuristic which uses LP relaxation solutions to assign weights to each block based on the expected mining time and then schedules the blocks by ranking the weights. The authors implemented the TopoSort heuristic to a case which has more than 4 million blocks with a time horizon of 15 years. According to the results presented, the initial feasible solution to the model which has an upper bound on mining and mill capacity constraints is achieved in less than a second. So far, none of the heuristic methods that generate initial integer feasible solutions are able to handle the models with blending constraints. Moreover, MVFP is the only one that can handle lower bound mining and mill capacity constraints but requires approximately 1h 20 minutes to generate a solution to a model with 10000 blocks which is too long for such a small size model. Also, the comparison between MVFP, LAH and TopoSort heuristic methods in terms of the quality of the solutions they generate has not presented in the literature.

2.3.3.1.3.2 Solution Improvement Heuristic Techniques

An initial integer feasible solution serves as a guide to the improvement heuristic techniques developed by Amaya *et al.* (2009), Cullenbine (2011), Lamghari and Dimitrakopoulos (2012, 2015), Lambert (2013), Lamghari *et al.* (2014). The case study presented by Amaya *et al.* (2009) shows that near optimal solutions were achieved in 4 hours by implementing Local search heuristic (LS) to the Marvin (53668 blocks), AmaricaMine (19320 blocks), AsiaMine (772800 blocks) datasets where the models have only upper bound mining and mill capacity constraints. Similar solutions are also presented by Chicoisne *et al.* (2012), when the authors ran the LS heuristic in about 4 hours once they obtained an initial integer feasible solution with the TopoSort heuristic. Moreover, the authors show that once the number of the resource capacity constraints increase, the optimality gap achieved in 4 hours also increased. The applicability of LS to the real mine models that include upper and lower bounds on resource and blending constraints still remains elusive. The Sliding time window heuristic (STWH) proposed by Cullenbine (2011) can solve the models consisting of upper and lower bounds on the mining and mill capacity constraints. The results presented by the author shows that it may take up to 2hr 40 min to solve a model with

25000 blocks and 15 time periods and produce a solution with an optimality gap of 4.3%. This is so far the best-known solution obtained in the models with lower bounds on the resource constraints. Moreover, since the approach utilizes Lagrange relaxation techniques, it will be vulnerable to potential gap problems. The Tailored Lagrange relaxation method (TLR) proposed by Lambert (2013) can be also applied to the models with upper and lower bounds on resource constraints. The author applied the TLR to the same case studies presented by Cullenbine in 2011 which makes the comparison of the STWH with TLR easier. Although STWH was able to generate a solution for the model with 25000 blocks, TLR could not obtain a solution with an optimality gap less than 6.4% in 10 hours. By looking at the solution times of the remaining models where the number of blocks vary from 10000 blocks to 18000 blocks, it can be concluded that the STWH has a better performance than TLR. Like STWH, the TLR method may also suffer from the weaknesses of the Lagrange relaxation approach since gap problems may exist. Lamghari and Dimitrakopoulos (2015) presented a case study where the Tabu Search (TS), variable neighborhood descent (VND), network flow (NF) and diversified local search (DLS) improvement heuristic methods are compared in terms of the solution times and the optimality gap they produce. The authors concluded that the TS has the worst performance and its solution time can increase significantly depending on the quality of the initial integer feasible solution it starts with. Although VND performs better than NF for the small size models, NF can outperform VND for the relatively larger size problems but may take a longer time than VND to find a solution. The authors also concluded that DLS which is the combination of NF and VND methods performs the best in terms of the quality of the solution it generates and also it does not depend on the quality of the initial integer feasible solution. These 4 methods are implemented on the stochastic models which consist of at most 48000 blocks with upper bound mining and mill capacity constraints. A performance comparison of these methods with the LS, STWH and TLR has not been presented. So far, it can be concluded that among the existing improvement heuristic methods LS can solve the models that has at most 4 million blocks in a block model and 15 time periods subject to upper bound mining and mill capacity constraints within a 4% of optimality gap in 4 hours. Also, STWH remains the only local improvement heuristic that can solve the model with 25000 blocks and 15 time periods in 2hr 40min within an optimality gap of 4.3%. If one considers solving a large case mine production scheduling problem that has 5 to 10 million blocks, multiple destinations subject to

mining and mill capacity, blending and varying slope constraints within a binary context, the currently developed heuristic algorithms will not be able to produce an integer optimal solution.

2.4 Summary of the Current State of Open Pit Mine Production Scheduling

The researchers have been trying to solve this problem since 1960's unfortunately, the proposed solution techniques are not capable of providing an optimal integer solution to the mine production scheduling problems modeled with mining and mill capacity, blending and stockpile constraints under grade uncertainty. There are numerous pushback design, heuristic and aggregation methods investigated in the literature and the closest integer optimal solutions are accomplished for the problems modeled with upper bound mining capacity constraints. Since, the size of a mining problem makes the exact integer programming solution techniques inapplicable, many attempts are made to solve the problem with pushback design, heuristic and aggregation methods which cannot be proven to converge to the true optimal solution. So far, TopoSort heuristic algorithm together with the Local search heuristic algorithm can generate an integer solution for a block model consists of 4 million blocks with a 4% optimality gap in 4hr when modeled with up to two upper bound capacity constraints and the sliding time window heuristic can provide an integer solution within a 4.3% optimality gap in 2hr 40min when modeled with an upper and lower bound capacity constraints for 25,000 blocks. Although these methods can generate close integer optimal solutions, their limited applicability results with a preclusion of their implementation on mine production scheduling problems. Contrarily, the new integer solution algorithm proposed in this dissertation that can solve the mine production scheduling problems modeled with multi capacities, grade blending, grade uncertainty, stockpiles, variable pit slopes and multi destinations as an integer programming problem will realize the optimization process on real life mining problems.

It is important to highlight the most influential developments closely related to the new integer solution algorithm proposed in this thesis as shown in Figure 2.1. The major contributions of each researcher are listed in the figure. Of course, there have been considerably more research conducted on this topic as discussed throughout this Chapter, but the ones presented in Figure 2.1 set the stage for the takeoff platform of the developments presented in this thesis.

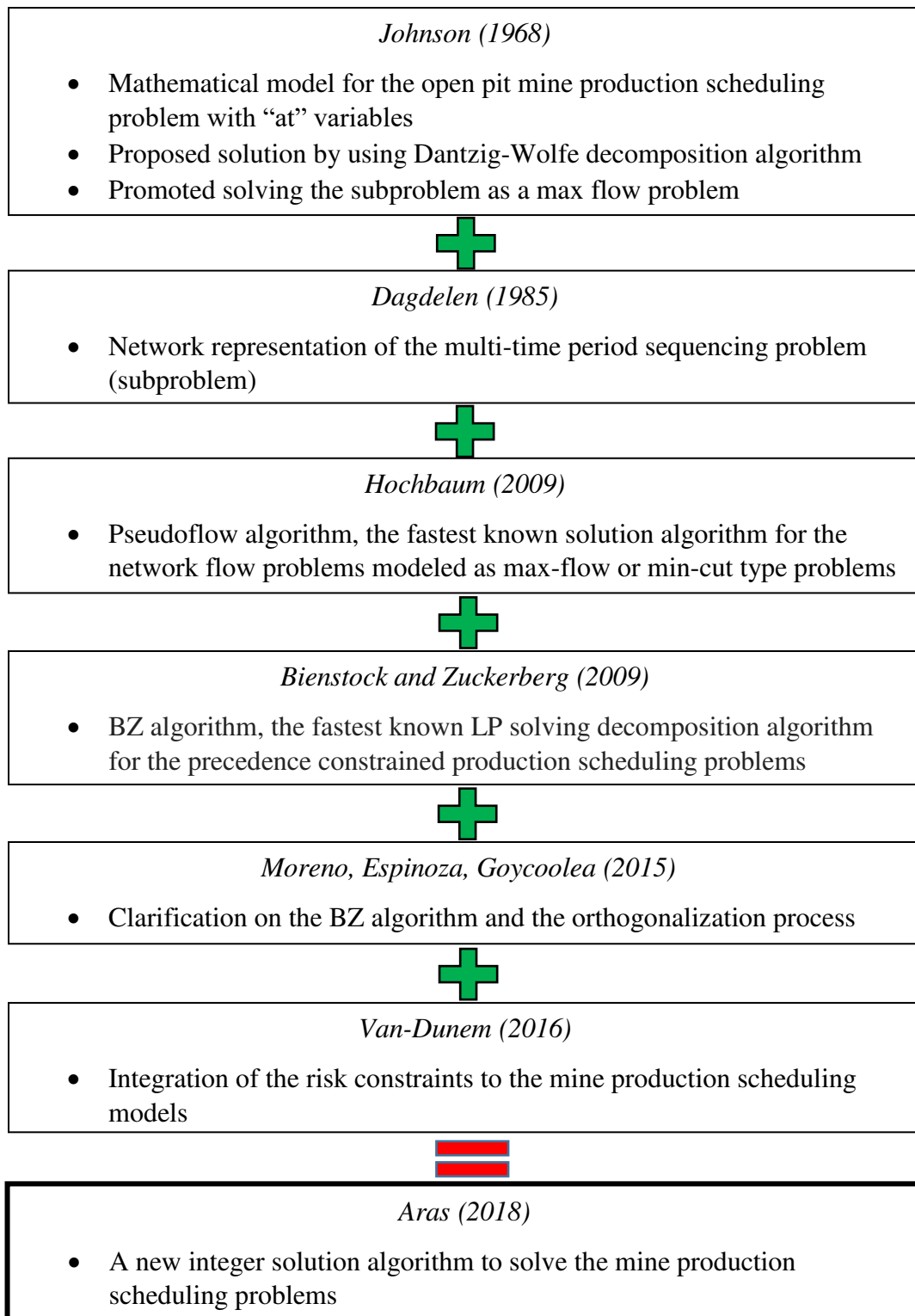


Figure 2.1: Most influential researchers on the development of the new integer solution algorithm

CHAPTER 3.

MATHEMATICAL MODEL FOR OPEN PIT MINE PRODUCTION SCHEDULING PROBLEM UNDER GRADE UNCERTAINTY

The mine production scheduling problem is an optimization problem which requires a mathematical model that incorporates the factors delineating the interactions between the elements of the complex mining system. The proposed integer solution algorithm in this thesis can solve the new mathematical model presented in this chapter that incorporates production and processing capacities, grade blending constraints, risk blending constraints, multi destinations, stockpiles and variable pit slope angles on a block by block level. The new integer solution algorithm benefits from the Bienstock-Zuckerberg decomposition algorithm. Thus, a new mathematical model will be derived by introducing new variables in order to transform the model to a representative form of the master problem of the Bienstock-Zuckerberg decomposition algorithm. Also, the original mathematical model presented by Bienstock Zuckerberg (2009,2010 and 2015) will be derived from the traditionally adapted mathematical formulation introduced by Johnson (1968). The mathematical models presented in this chapter are in generic form that will make it easier for the operator to adjust the models based on the needs of the operation. The proposed models differ from the traditional mine production scheduling models with the integration of two important concepts. The risk constraints which are first implemented by Van-Dunem (2016), are incorporated in order to control the uncertainty by applying an upper bound on the number of the inferred blocks sent to the mill and lower bounds on the amount of indicated and measured blocks. With the incorporation of the risk constraints, the impact of uncertainty on meeting the operational or production targets will be minimized. Moreover, stockpiles based on risk categories are also integrated to the mathematical model. Stockpiles are utilized in mining operations for various purposes. In the event of an equipment breakdown, instead of halting the production, the plant may continue to receive materials from the stockpiles. Moreover, the upside potential of the commodity price is also accounted by keeping the marginal ore in the stockpile for a potential recovery instead of sending to the waste dump. Also, during the optimization process if the capacity of the plant can be filled by mining a very valuable ore block, any marginal ore block preceding this nugget block will be

sent to the waste dump if the stockpiles are excluded. Hence, the stockpiles can account for the capacity limitations. Traditionally, the mine production scheduling problems are modeled with stockpiles based on grade intervals or material types. The proposed model will further split the stockpiles into risk categories, in order to prevent the blend of high risk ore with the low risk ore. For example, if an inferred ore block is designated to a low-grade stockpile, it will be sent to a low-grade stockpile that holds the inferred blocks. This approach will maximize the risk control in the presence of stockpiles. Furthermore, the average grade of the material coming from the stockpile was assumed to be equal to the lowest bound of the stockpile grade interval to prevent overestimating the stockpile average grade. The average grade could also be used. The choice is left to the user. The process flow representing the proposed mine production scheduling mathematical model is shown in Figure 4.1.

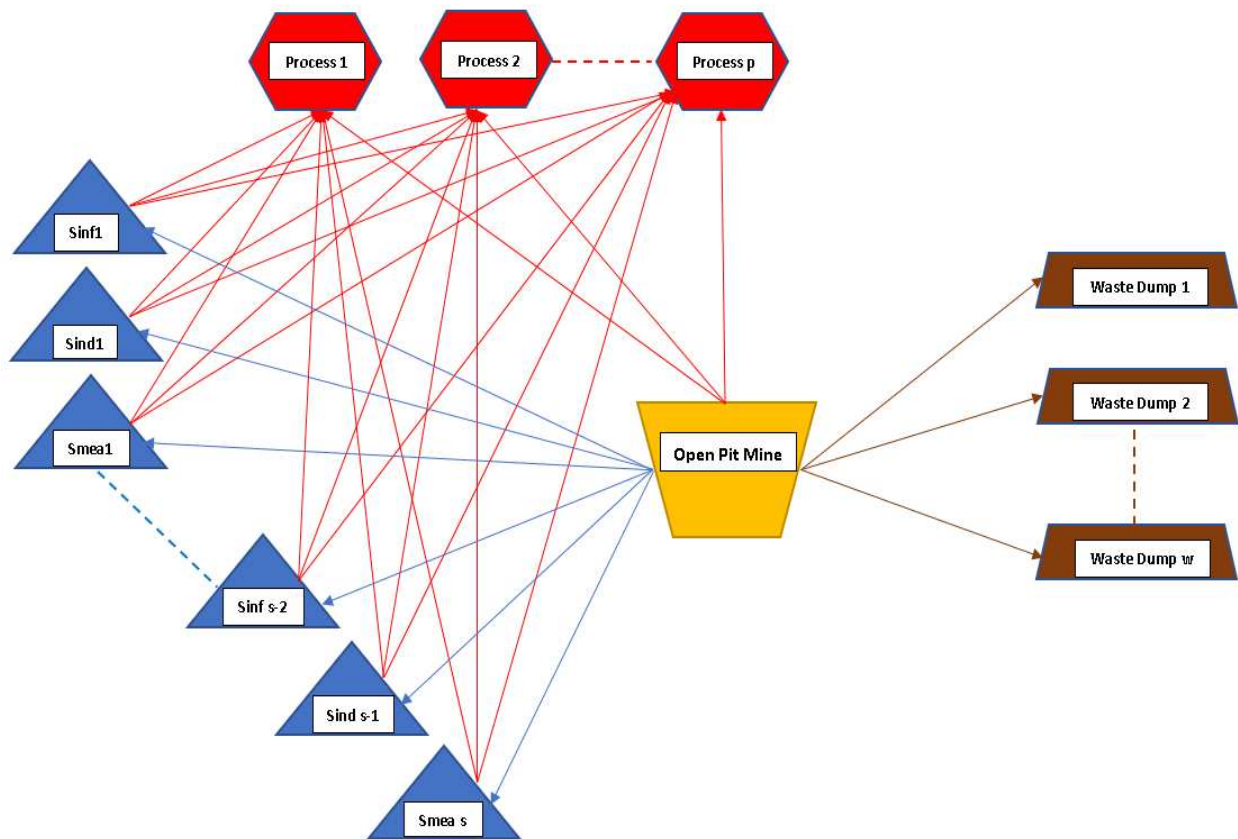


Figure 3.1: Process Flow of the Mine Production Scheduling Problem

3.1 Mathematical Model Incorporated with Dynamic Cutoff Grade Strategy and Stockpiles Formulated with “At” Variables

Indices and Sets:

$b \in B$: set of all blocks b

$b' \in B_b \subseteq B$: set of blocks b' that must be extrated before block b (block b' precedes block b)

$b_1 \in B^1 \subseteq B$: set of all blocks b_1 classified into the Inferred resource category

$b_2 \in B^2 \subseteq B$: set of all blocks b_2 classified into the Indicated resource category

$b_3 \in B^3 \subseteq B$: set of all blocks b_3 classified into the Measured resource category

$d \in D$: set of all destinations d

$w \in W \subseteq D$: set of all waste dump destinations w

$p \in P \subseteq D$: set of all process destinations p

$s \in S \subseteq D$: set of all stockpile destinations s

$s_1 \in S_{R_1} \subseteq S$: set of all inferred stockpile destinations s_1

$s_2 \in S_{R_2} \subseteq S$: set of all indicated stockpile destinations s_2

$s_3 \in S_{R_3} \subseteq S$: set of all measured stockpile destinations s_3

$t \in T$: set of all time periods t

Data:

c_{bd}^t : net present value (NPV) generated by sending a ton of block b to destination d in period t (\$/ton)

v_{sp}^t : net present value (NPV) generated by sending a ton of material from stockpile s to process destination d in period t (\$/ton)

p_b : tonnage of block b

g_b : ore grade of block b

r_{b1} : 1 if the block b belongs to an inferred category, 0 otherwise

r_{b2} : 1 if the block b belongs to an indicated category, 0 otherwise

r_{b3} : 1 if the block b belongs to a measured category, 0 otherwise

\overline{M}^t : maximum tonnage of the mining capacity in time period t

\overline{W}_w^t : maximum tonnage of the waste dump capacity in time period t

\overline{C}_p^t : maximum tonnage of the process capacity at process destination p , in time period t

\underline{G}_p^t : minimum average grade required at process destination p , in time period t

\overline{R}_1^t : maximum proportion of the inferred material processed in time period t (percent)

R_2^t : minimum proportion of the indicated material processed in time period t (percent)

R_3^t : minimum proportion of the measured material processed in time period t (percent)

Decision Variables:

x_{bd}^t = the fraction of block b , that is extracted in time period t , and sent to destination d

y_{sp}^t = the tonnage of material, sent from stockpile s to the procesing destination p , in time period t

C_s^t = remaining material tonnage at stockpile s at the end of period t

Objective Function:

$$\text{Max } Z = \sum_{b \in B} \sum_{d \in D} \sum_{t \in T} c_{bd}^t x_{bd}^t p_b + \sum_{s \in S} \sum_{p \in P} v_{sp}^t y_{sp}^t \quad (3.1)$$

Subject To:

Reserve Constraint:

$$\sum_{d \in D} \sum_{t \in T} x_{bd}^t \leq 1 \quad \forall b \in B, \forall t \in T \quad (3.2)$$

Block Sequencing Constraint:

$$\sum_{t' \leq t} \sum_{d \in D} x_{bd}^{t'} \leq \sum_{t' \leq t} \sum_{d \in D} x_{b'd}^{t'} \quad \forall b \in B, b' \in B_b, t \in T \quad (3.3)$$

Maximum Mining Capacity:

$$\sum_{b \in B} \sum_{d \in D} p_b x_{bd}^t \leq \bar{M}^t \quad \forall t \in T \quad (3.4)$$

Maximum Capacity at Process Destination:

$$\sum_{b \in B} p_b x_{bp}^t + \sum_{s \in S} y_{sp}^t \leq \bar{C}_p \quad \forall t \in T, p \in P \quad (3.5)$$

Maximum Capacity at Waste Dump Destination:

$$\sum_{b \in B} p_b x_{bw}^t \leq \bar{W}_w \quad w \in W, \forall t \in T \quad (3.6)$$

Minimum Grade Blending Requirement at Process Destination:

$$\sum_{b \in B} (G_p^t - g_b) p_b x_{bp}^t \leq 0 \quad \forall t \in T, p \in P \quad (3.7)$$

Maximum Proportion of Inferred Material at Process Destination:

$$\sum_{b \in B} \sum_{p \in P} (r_{b1} - \bar{R}_1^t) p_b x_{bp}^t + \sum_{s_1 \in S_{R_1}} \sum_{p \in P} y_{s_1 p}^t (1 - \bar{R}_1^t) - \sum_{s \neq s_1 \in S} \sum_{p \in P} \bar{R}_1^t y_{sp}^t \leq 0 \quad (3.8)$$

$\forall t \in T$

Minimum Proportion of Indicated Material at Process Destination:

$$\sum_{b \in B} \sum_{p \in P} (\underline{R}_2^t - r_{b2}) p_b x_{bp}^t - \sum_{s_2 \in S_{R_2}} \sum_{p \in P} y_{s_2 p}^t (\underline{R}_2^t - 1) + \sum_{s \neq s_2 \in S} \sum_{p \in P} \underline{R}_2^t y_{sp}^t \leq 0 \quad (3.9)$$

$\forall t \in T$

Minimum Proportion of Measured Material at Process Destination:

$$\sum_{b \in B} \sum_{p \in P} (\underline{R}_3^t - r_{b3}) p_b x_{bp}^t - \sum_{s_3 \in S_{R_3}} \sum_{p \in P} y_{s_3 p}^t (\underline{R}_3^t - 1) + \sum_{s \neq s_3 \in S} \sum_{p \in P} \underline{R}_3^t y_{sp}^t \leq 0 \quad (3.10)$$

$\forall t \in T$

Stockpile Balance Constraint:

$$\sum_{b \in B} p_b x_{bs}^t - \sum_{p \in P} y_{sp}^t + C_s^{t-1} \leq C_s^t \quad \forall s \in S, t \in T \quad (3.11)$$

Variable Restrictions:

$$y_{sp}^t \geq 0 \quad \forall s \in S, t \in T, p \in D \quad (3.12)$$

$$C_s^t \geq 0 \quad \forall s \in S, t \in T \quad (3.13)$$

$$x_{bd}^t \in [0, 1] \quad \forall b \in B, t \in T, d \in D \quad (3.14)$$

The objective function (3.1) maximize the discounted value of all extracted blocks across all destination and periods. Constraints (3.2) ensure the total mineable proportion of any block sent to any destination at any period cannot exceed the totality of the block. Constraints (3.3) represent the sequencing relationship between the target block b and the dependent block b' . Constraints (3.4) enforce the upper bound on the maximum mining capacity. Constraints (3.5) enforce the upper bound on the maximum processing capacity. Constraints (3.6) enforce upper bound on the capacity of the waste dump. Constraints (3.7) impose the lower bound on the average grade

requirement at process destination. Constraints (3.8) restrict the proportion of the inferred material to be processed by enforcing an upper bound. Constraints (3.9) demand a minimum proportion of the processed material to be indicated by enforcing a lower bound. Constraints (3.10) demand a minimum proportion of the processed material to be measured by enforcing a lower bound. Constraints (3.11) are stockpile material balance constraints. Constraints (3.12) ensure the tonnage of the material sent from the stockpile is a continuous nonnegative variable. Constraints (3.13) ensure the tonnage of the material remaining in the stockpile is a continuous nonnegative variable. Constraints (3.14) ensure that all the decision variables are continuous between zero and one. There could be other constraints such as minimum capacity and maximum blending constraints, but such conditions seem unlikely therefore we specifically did not include those constraints to the mathematical model.

3.2 Variable Substitution for Single Path “By” Variables

The new mathematical model adapted to the new integer solution algorithm will be derived in 2 stages. The first stage of the variable substitution process will be demonstrated by initially introducing the “by” variable. Transforming the “at” variables to single path “by” variables will ensure interactions between the time periods as shown in Figure 3.2. All the rest of the indices, sets, data and the stockpile variables will be kept the same as the ones used in the “at” formulation, hence a new definition is not required. The variable substitution process will proceed with such fashion that the ultimate goal is to maintain the “at” meaning for all the constraints.

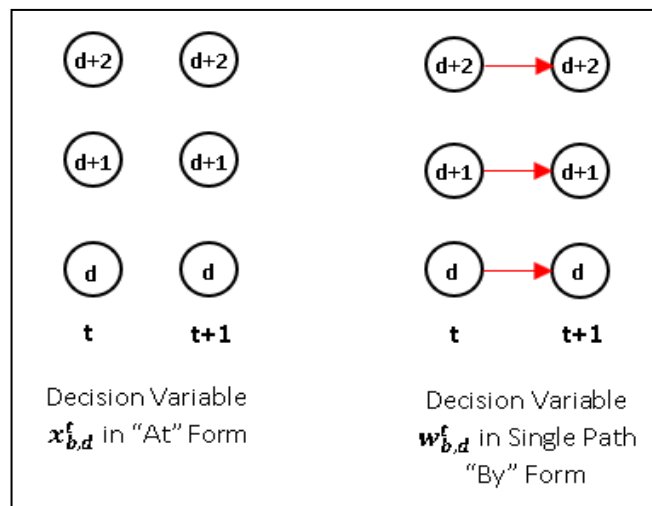


Figure 3.2 : Node structure of the “at” variables and the “by” variables

Decision Variables:

w_{bd}^t = the fraction of block b , that is extracted by time period t , and sent to destination d

The following shows the relationship between the at variable “ x_{bd}^t ” and the by variable “ w_{bd}^t ”.

$$w_{bd}^t = \sum_{t' \leq t} x_{bd}^{t'} \quad \forall b \in B, t \in T, d \in D \quad (3.15)$$

$$x_{bd}^1 = w_{bd}^1 \quad \forall b \in B, d \in D, \text{ iff } t = 1 \quad (3.16)$$

$$x_{bd}^t = w_{bd}^t - w_{bd}^{t-1} \quad \forall b \in B, d \in D, t \in T \text{ iff } t \geq 2 \quad (3.17)$$

The **objective function** of the mine production scheduling problem without stockpiles when formulated with “at” variables is:

$$\text{Max } Z = \sum_{b \in B} \sum_{d \in D} \sum_{t \in T} c_{bd}^t x_{bd}^t$$

Variable substitution process will proceed with replacing the at variable “ x_{bd}^t ” with the “by” variable “ w_{bd}^t ” as shown below:

Let's rewrite the objective function $\text{Max } Z = \sum_{b \in B} \sum_{d \in D} \sum_{t \in T} c_{bd}^t x_{bd}^t$ as:

$$\text{Max } Z = \sum_{b \in B} \sum_{d \in D} (c_{bd}^1 x_{bd}^1 + c_{bd}^2 x_{bd}^2 + c_{bd}^3 x_{bd}^3 \dots \dots c_{bd}^{T-1} x_{bd}^{T-1} + c_{bd}^T x_{bd}^T)$$

Substitute “ x_{bd}^1 ” with “ w_{bd}^1 ” for $t = 1$ and “ $w_{bd}^{t+1} - w_{bd}^t$ ” for $t \rightarrow t + 1$ to T :

$$\text{Max } Z = \sum_{b \in B} \sum_{d \in D} (c_{bd}^1 w_{bd}^1 + c_{bd}^2 (w_{bd}^2 - w_{bd}^1) + c_{bd}^3 (w_{bd}^3 - w_{bd}^2) + \dots \dots + c_{bd}^{T-1} (w_{bd}^{T-1} - w_{bd}^{T-2}) + c_{bd}^T (w_{bd}^T - w_{bd}^{T-1}))$$

$$\text{Max } Z = \sum_{b \in B} \sum_{d \in D} (c_{bd}^1 w_{bd}^1 + c_{bd}^2 w_{bd}^2 - c_{bd}^2 w_{bd}^1 + c_{bd}^3 w_{bd}^3 - c_{bd}^3 w_{bd}^2 + \dots \dots + c_{bd}^{T-1} w_{bd}^{T-1} - c_{bd}^{T-1} w_{bd}^{T-2} + c_{bd}^T w_{bd}^T - c_{bd}^T w_{bd}^{T-1})$$

Rearrange the w_{bd}^t variables:

$$\text{Max } Z = \sum_{b \in B} \sum_{d \in D} (w_{bd}^1 (c_{bd}^1 - c_{bd}^2) + w_{bd}^2 (c_{bd}^2 - c_{bd}^3) + w_{bd}^3 (c_{bd}^3 - c_{bd}^4) + \dots \dots + w_{bd}^{T-2} (c_{bd}^{T-2} - c_{bd}^{T-1}) + w_{bd}^{T-1} (c_{bd}^{T-1} - c_{bd}^T) + w_{bd}^T c_{bd}^T)$$

This can be simplified as:

$$\text{Max } Z = \sum_{b \in B} \sum_{d \in D} \sum_{t < T} w_{bd}^t (c_{bd}^t - c_{bd}^{t+1}) + \sum_{b \in B} \sum_{d \in D} w_{bd}^T c_{bd}^T \quad (3.18)$$

The *sequencing constraints* of the mine production scheduling problem when formulated with “at” variables are:

$$\sum_{t' \leq t} \sum_{d \in D} x_{bd}^{t'} \leq \sum_{t' \leq t} \sum_{d \in D} x_{b'd}^{t'} \quad \forall b \in B, b' \in B_b, t \in T$$

The constraints can be also written as:

$$\sum_{d \in D} x_{bd}^1 + x_{bd}^2 \dots x_{bd}^t \leq \sum_{d \in D} x_{b'd}^1 + x_{b'd}^2 \dots x_{b'd}^t \quad \text{for } t \in T$$

We know that $w_{bd}^t = \sum_{t' \leq t} x_{bd}^{t'}$ which can be written as $w_{bd}^t = x_{bd}^1 + x_{bd}^2 \dots x_{bd}^t$

By making the variable substitution, the sequencing constraints can be replaced with the following constraint in “by” form:

$$\sum_{d \in D} w_{bd}^t \leq \sum_{d \in D} w_{b'd}^t \quad \forall b \in B, b' \in B_b, t \in T \quad (3.19)$$

For the *capacity, grade blending, and risk constraints*, “at” to “by” variable conversion can be done straightforward as shown below:

Given mining capacity constraints formulated with at variables such as;

$$\sum_{b \in B} \sum_{d \in D} p_b x_{bd}^t \leq \bar{M}^t \quad \forall t \in T,$$

simply replacing " x_{bd}^1 " with " w_{bd}^1 " for $t = 1$ and " $w_{bd}^t - w_{bd}^{t-1}$ " for $t \rightarrow t + 1$ to T

$$\sum_{b \in B} \sum_{d \in D} p_b w_{bd}^1 \leq \bar{M}^1 \quad \text{iff } t = 1 \quad (3.20)$$

$$\sum_{b \in B} \sum_{d \in D} p_b (w_{bd}^t - w_{bd}^{t-1}) \leq \bar{M}^t \quad \forall t \in T \text{ iff } t \geq 2 \quad (3.21)$$

capacity constraints are represented with by variables.

Likewise, for the blending constraints:

$$\text{At formulation} \rightarrow \sum_{b \in B} (G_p^t - g_b) p_b x_{bp}^t \leq 0 \quad \forall t \in T, p \in P$$

$$\text{By formulation(1)} \rightarrow \sum_{b \in B} (G_p^1 - g_b) p_b w_{bp}^1 \leq 0 \quad \forall p \in P, \text{ iff } t = 1 \quad (3.22)$$

$$\text{By formulation(2)} \rightarrow \sum_{b \in B} (G_p^t - g_b)(w_{bp}^t - w_{bp}^{t-1}) p_b \leq 0 \quad (3.23)$$

$$\forall p \in P, t \in T \text{ iff } t \geq 2$$

For the risk constraints:

$$\text{At formulation} \rightarrow \sum_{b \in B} \sum_{p \in P} (r_{b1} - \bar{R}_1) p_b x_{bp}^t \leq 0 \quad \forall t \in T$$

$$\text{By formulation(1)} \rightarrow \sum_{p \in P} \sum_{b \in B} (r_{b1} - \bar{R}_1) p_b w_{bp}^1 \leq 0 \quad \text{iff } t = 1 \quad (3.24)$$

$$\text{By formulation(2)} \rightarrow \sum_{p \in P} \sum_{b \in B} (r_{b1} - \bar{R}_1) p_b (w_{bp}^t - w_{bp}^{t-1}) \leq 0 \quad \forall t \in T \text{ iff } t \geq 2 \quad (3.25)$$

Mathematical model for the mine production scheduling problem formulated with “by” variables is given as:

Objective Function:

$$\text{Max } Z = \sum_{b \in B} \sum_{d \in D} \sum_{t < T} w_{bd}^t (c_{bd}^t - c_{bd}^{t+1}) + \sum_{b \in B} \sum_{d \in D} w_{bd}^T c_{bd}^T + \sum_{s \in S} \sum_{p \in P} v_{sp}^t y_{sp}^t \quad (3.26)$$

Subject To:

By Variable Constraint:

$$w_{bd}^t \leq w_{bd}^{t+1} \quad \forall b \in B, t < T, d \in D \quad (3.27)$$

Block Sequencing Constraint:

$$\sum_{d \in D} w_{bd}^t \leq \sum_{d \in D} w_{b'd}^t \quad \forall b \in B, b' \in B_b, t \in T \quad (3.28)$$

Maximum Mining Capacity:

$$\sum_{b \in B} \sum_{d \in D} p_b w_{bd}^1 \leq \bar{M}^1 \quad \text{iff } t = 1 \quad (3.29)$$

$$\sum_{b \in B} \sum_{d \in D} p_b (w_{bd}^t - w_{bd}^{t-1}) \leq \bar{M}^t \quad \forall t \in T \text{ iff } t \geq 2 \quad (3.30)$$

Maximum Capacity at Process Destination:

$$\sum_{b \in B} p_b w_{bp}^1 + \sum_{s \in S} y_{sp}^1 \leq \bar{C}_p \quad \forall p \in P, \text{ iff } t = 1 \quad (3.31)$$

$$\sum_{b \in B} p_b (w_{bp}^t - w_{bp}^{t-1}) + \sum_{s \in S} y_{sp}^t \leq \bar{C}_p^t \quad \forall p \in P, t \in T \text{ iff } t \geq 2 \quad (3.32)$$

Maximum Capacity at Waste Dump Destination:

$$\sum_{b \in B} p_b w_{bw}^1 \leq \bar{W}_w^1 \quad \forall w \in W, \text{ iff } t = 1 \quad (3.33)$$

$$\sum_{b \in B} p_b (w_{bw}^t - w_{bw}^{t-1}) \leq \bar{W}_w^t \quad \forall w \in W, t \in T \text{ iff } t \geq 2 \quad (3.34)$$

Minimum Grade Blending Requirement at Process Destination:

$$\sum_{b \in B} (\underline{G}_p^1 - g_b) p_b w_{bp}^1 + \sum_{s \in S} (\underline{G}_p^1 - g_s) y_{sp}^1 \leq 0 \quad \forall p \in P, \text{ iff } t = 1 \quad (3.35)$$

$$\sum_{b \in B} (\underline{G}_p^t - g_b) p_b (w_{bp}^t - w_{bp}^{t-1}) + \sum_{s \in S} (\underline{G}_p^t - g_s) y_{sp}^t \leq 0 \quad \forall p \in P, t \in T \text{ iff } t \geq 2 \quad (3.36)$$

Maximum Proportion of Inferred Material at Process Destination:

$$\sum_{p \in P} \sum_{b \in B} (r_{b1} - \bar{R}_1^1) p_b w_{bp}^1 + \sum_{p \in P} \sum_{s_1 \in S_{R_1}} y_{s_1 p}^1 (1 - \bar{R}_1^1) - \sum_{p \in P} \sum_{s \neq s_1 \in S} \bar{R}_1^1 y_{sp}^1 \leq 0 \quad (3.37)$$

iff $t = 1$

$$\sum_{p \in P} \sum_{b \in B} (r_{b1} - \bar{R}_1^t) p_b (w_{bp}^t - w_{bp}^{t-1}) + \sum_{p \in P} \sum_{s_1 \in S_{R_1}} y_{s_1 p}^t (1 - \bar{R}_1^t) - \sum_{p \in P} \sum_{s \neq s_1 \in S} \bar{R}_1^t y_{sp}^t \leq 0 \quad \forall t \in T \text{ iff } t \geq 2 \quad (3.38)$$

Minimum Proportion of Indicated Material at Process Destination:

$$\sum_{p \in P} \sum_{b \in B} (\underline{R}_2^1 - r_{b2}) p_b w_{bp}^1 - \sum_{p \in P} \sum_{s_2 \in S_{R_2}} y_{s_2 p}^1 (\underline{R}_2^1 - r_{b2}) + \sum_{p \in P} \sum_{s \neq s_2 \in S} \underline{R}_2^1 y_{sp}^1 \leq 0 \quad (3.39)$$

iff $t = 1$

$$\sum_{p \in P} \sum_{b \in B} (\underline{R}_2^t - r_{b2}) p_b (w_{bp}^t - w_{bp}^{t-1}) - \sum_{p \in P} \sum_{s_2 \in S_{R_2}} y_{s_2 p}^t (\underline{R}_2^t - r_{b2}) + \sum_{p \in P} \sum_{s \neq s_2 \in S} \underline{R}_2^t y_{sp}^t \leq 0 \quad \forall t \in T \text{ iff } t \geq 2 \quad (3.40)$$

Minimum Proportion of Measured Material at Process Destination:

$$\sum_{p \in P} \sum_{b \in B} (\underline{R}_3^1 - r_{b3}) p_b w_{bp}^1 - \sum_{p \in P} \sum_{s_3 \in S_{R_3}} y_{s_3 p}^1 (\underline{R}_3^1 - r_{b3}) + \sum_{p \in P} \sum_{s \neq s_3 \in S} \underline{R}_3^1 y_{sp}^1 \leq 0 \quad (3.41)$$

iff $t = 1$

$$\begin{aligned} \sum_{p \in P} \sum_{b \in B} (\underline{R}_3^t - r_{b3}) p_b (w_{bp}^t - w_{bp}^{t-1}) - \sum_{p \in P} \sum_{s_3 \in S_{R_3}} y_{s_3 p}^t (\underline{R}_3^t - r_{b3}) \\ + \sum_{p \in P} \sum_{s \neq s_3 \in S} \underline{R}_3^t y_{sp}^t \leq 0 \end{aligned} \quad \forall t \in T \text{ iff } t \geq 2 \quad (3.42)$$

Stockpile Balance Constraint:

$$\sum_{b \in B} p_b (w_{bs}^t - w_{bs}^{t-1}) - \sum_{p \in P} y_{sp}^t + C_s^{t-1} \leq C_s^t \quad \forall s \in S, t \in T \quad (3.43)$$

Variable Restrictions:

$$y_{sp}^t \geq 0 \quad \forall s \in S, t \in T, p \in D \quad (3.44)$$

$$C_s^t \geq 0 \quad \forall s \in S, t \in T \quad (3.45)$$

$$w_{bd}^t \in [0, 1] \quad \forall b \in B, t \in T, d \in D \quad (3.46)$$

3.3 Variable Substitution for New Dual Path Double “By” Variables

This section will demonstrate the second stage of the variable substitution process which will lead to a new mathematical formulation introduced in this thesis. In the first stage, the single path “by” variable “ w_{bd}^t ” was derived by summing the at variable “ x_{bd}^t ” across the time periods. Intuitively, the second stage will proceed by summing the single path “by” variable “ w_{bd}^t ” across the destinations; which will translate into a node structure of a new dual path double “by” variable “ u_{bd}^t ” as shown in Figure 3.3. The underlying reason for proceeding with this complicated process is eventually preparing the mathematical model to be ready to implement the proposed integer solution algorithm. Since the proposed algorithm uses the same master problem defined by Bienstock Zuckerberg algorithm, the mathematical model generated here needs to honor the format of the master problem.

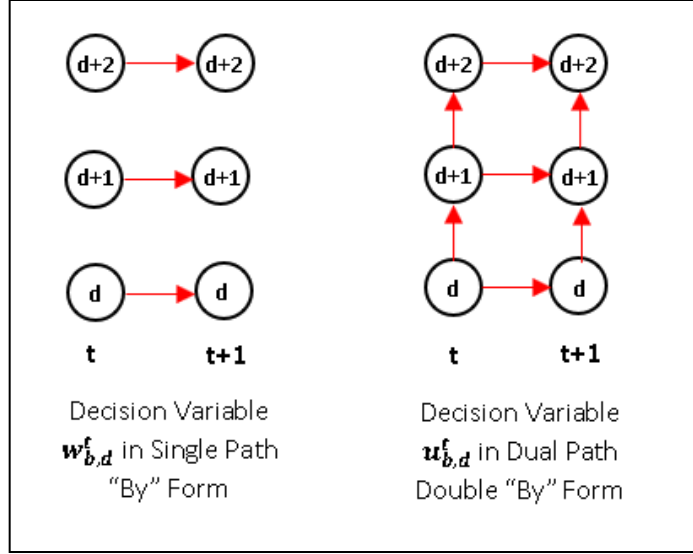


Figure 3.3 : Node structure of the single path “by” variables and the dual path double “by” variables

Decision Variables:

The following shows the relationship between the single path “by” variable “ w_{bd}^t ” and the new dual path double “by” variable “ u_{bd}^t ”.

$$u_{bd}^t = \sum_{d' \leq d} w_{bd'}^t \quad \forall b \in B, t \in T, d \in D \quad (3.47)$$

$$w_{b1}^t = u_{b1}^t \quad \forall b \in B, t \in T, \text{ iff } d = 1 \quad (3.48)$$

$$w_{bd}^t = u_{bd}^t - u_{bd-1}^t \quad \forall b \in B, t \in T, d \in D \text{ iff } d \geq 2 \quad (3.49)$$

The **objective function** of the mine production scheduling problem without stockpiles when formulated with single path “by” variable is:

$$\sum_{b \in B} \sum_{d \in D} \sum_{t < T} w_{bd}^t (c_{bd}^t - c_{bd}^{t+1}) + \sum_{b \in B} \sum_{d \in D} w_{bd}^T c_{bd}^T$$

The following demonstrates the process of replacing the single path “by” variable “ w_{bd}^t ” with the new dual path double “by” variable “ u_{bd}^t ”:

Let's rewrite the first part of the objective function formulated with the "old" by variable

P1: $\sum_{b \in B} \sum_{d \in D} \sum_{t < T} w_{bd}^t (c_{bd}^t - c_{bd}^{t+1})$ as:

$$\begin{aligned}
& \sum_{b \in B} \sum_{t < T} (w_{b_1}^t (c_{b_1}^t - c_{b_1}^{t+1}) + w_{b_2}^t (c_{b_2}^t - c_{b_2}^{t+1}) + w_{b_3}^t (c_{b_3}^t - c_{b_3}^{t+1}) \\
& \quad + \dots \dots + w_{b_{D-1}}^t (c_{b_{D-1}}^t - c_{b_{D-1}}^{t+1}) + w_{b_D}^t (c_{b_D}^t - c_{b_D}^{t+1})) \\
& \sum_{b \in B} \sum_{t < T} (w_{b_1}^t c_{b_1}^t - w_{b_1}^t c_{b_1}^{t+1} + w_{b_2}^t c_{b_2}^t - w_{b_2}^t c_{b_2}^{t+1} + w_{b_3}^t c_{b_3}^t - w_{b_3}^t c_{b_3}^{t+1} \\
& \quad + \dots \dots \dots + w_{b_{D-1}}^t c_{b_{D-1}}^t - w_{b_{D-1}}^t c_{b_{D-1}}^{t+1} + w_{b_D}^t c_{b_D}^t - w_{b_D}^t c_{b_D}^{t+1}) \\
& \text{Substitute "w}_{b_1}^t" \text{ with "u}_{b_1}^t" \text{ for } d = 1 \text{ and "u}_{b_{d+1}}^t - u_{b_d}^t" \text{ for } d \rightarrow d + 1 \text{ to } D
\end{aligned}$$

$$\begin{aligned}
& \sum_{b \in B} \sum_{t < T} (u_{b_1}^t c_{b_1}^t - u_{b_1}^t c_{b_1}^{t+1} + (u_{b_2}^t - u_{b_1}^t) c_{b_2}^t - (u_{b_2}^t - u_{b_1}^t) c_{b_2}^{t+1} \\
& \quad + (u_{b_3}^t - u_{b_2}^t) c_{b_3}^t - (u_{b_3}^t - u_{b_2}^t) c_{b_3}^{t+1} \\
& \quad + \dots \dots \dots + (u_{b_{D-1}}^t - u_{b_{D-2}}^t) c_{b_{D-1}}^t - (u_{b_{D-1}}^t - u_{b_{D-2}}^t) c_{b_{D-1}}^{t+1} \\
& \quad + (u_{b_D}^t - u_{b_{D-1}}^t) c_{b_D}^t - (u_{b_D}^t - u_{b_{D-1}}^t) c_{b_D}^{t+1}) \\
& \sum_{b \in B} \sum_{t < T} (u_{b_1}^t c_{b_1}^t - u_{b_1}^t c_{b_1}^{t+1} + u_{b_2}^t c_{b_2}^t - u_{b_1}^t c_{b_2}^t - u_{b_2}^t c_{b_2}^{t+1} + u_{b_1}^t c_{b_2}^{t+1} \\
& \quad + u_{b_3}^t c_{b_3}^t - u_{b_2}^t c_{b_3}^t - u_{b_3}^t c_{b_3}^{t+1} + u_{b_2}^t c_{b_3}^{t+1} \\
& \quad + \dots \dots \dots + u_{b_{D-1}}^t c_{b_{D-1}}^t - u_{b_{D-2}}^t c_{b_{D-1}}^t - u_{b_{D-1}}^t c_{b_{D-1}}^{t+1} + u_{b_{D-2}}^t c_{b_{D-1}}^{t+1} \\
& \quad + u_{b_D}^t c_{b_D}^t - u_{b_{D-1}}^t c_{b_D}^t - u_{b_D}^t c_{b_D}^{t+1} + u_{b_{D-1}}^t c_{b_D}^{t+1}) \\
& \sum_{b \in B} \sum_{t < T} (u_{b_1}^t ((c_{b_1}^t - c_{b_2}^t) - (c_{b_1}^{t+1} - c_{b_2}^{t+1})) + u_{b_2}^t ((c_{b_2}^t - c_{b_3}^t) - (c_{b_2}^{t+1} - c_{b_3}^{t+1})) \\
& \quad + u_{b_3}^t ((c_{b_3}^t - c_{b_4}^t) - (c_{b_3}^{t+1} - c_{b_4}^{t+1})) \\
& \quad + \dots \dots + u_{b_{D-1}}^t ((c_{b_{D-1}}^t - c_{b_D}^t) - (c_{b_{D-1}}^{t+1} - c_{b_D}^{t+1})) + u_{b_D}^t (c_{b_D}^t - c_{b_D}^{t+1}))
\end{aligned}$$

This can be simplified as:

$$\begin{aligned}
\mathbf{P1}^*: \text{Max } Z &= \sum_{b \in B} \sum_{d < D} \sum_{t < T} u_{bd}^t ((c_{bd}^t - c_{bd+1}^t) - (c_{bd}^{t+1} - c_{bd+1}^{t+1})) \\
& \quad + \sum_{b \in B} \sum_{t < T} u_{bD}^t (c_{bD}^t - c_{bD}^{t+1})
\end{aligned} \tag{3.50}$$

Let's rewrite the second part of the objective function formulated with the "old" by variable

$$\mathbf{P2}: \sum_{b \in B} \sum_{d \in D} w_{bd}^T c_{bd}^T \text{ as:}$$

$$\sum_{b \in B} w_{b1}^T c_{b1}^T + w_{b2}^T c_{b2}^T + w_{b3}^T c_{b3}^T + \dots + w_{bD-1}^T c_{bD-1}^T + w_{bD}^T c_{bD}^T$$

Substitute " w_{b1}^T " with " u_{b1}^T " for $d = 1$ and " $u_{bd+1}^T - u_{bd}^T$ " for $d \rightarrow d + 1$ to D

$$\sum_{b \in B} u_{b1}^T c_{b1}^T + (u_{b2}^T - u_{b1}^T) c_{b2}^T + (u_{b3}^T - u_{b2}^T) c_{b3}^T + \dots + (u_{bD-1}^T - u_{bD-2}^T) c_{bD-1}^T$$

$$+ (u_{bD}^T - u_{bD-1}^T) c_{bD}^T$$

$$\sum_{b \in B} u_{b1}^T c_{b1}^T + u_{b2}^T c_{b2}^T - u_{b1}^T c_{b2}^T + u_{b3}^T c_{b3}^T - u_{b2}^T c_{b3}^T$$

$$+ \dots + u_{bD-1}^T c_{bD-1}^T - u_{bD-2}^T c_{bD-1}^T + u_{bD}^T c_{bD}^T - u_{bD-1}^T c_{bD}^T$$

$$\sum_{b \in B} u_{b1}^T (c_{b1}^T - c_{b2}^T) + u_{b2}^T (c_{b2}^T - c_{b3}^T) + u_{b3}^T (c_{b3}^T - c_{b4}^T)$$

$$+ \dots + u_{bD-1}^T (c_{bD-1}^T - c_{bD}^T) c_{bD-1}^T + u_{bD}^T c_{bD}^T$$

This can be simplified as:

$$\mathbf{P2}^*: \text{Max } Z = \sum_{b \in B} \sum_{d < D} u_{bd}^T (c_{bd}^T - c_{bd+1}^T) + \sum_{b \in B} u_{bD}^T c_{bD}^T \quad (3.51)$$

Finally, the new objective function can be obtained by combining $\mathbf{P1}^*$ with $\mathbf{P2}^*$ such as:

$$\begin{aligned} \text{Max } Z = & \sum_{b \in B} \sum_{d < D} \sum_{t < T} u_{bd}^t ((c_{bd}^t - c_{bd+1}^t) - (c_{bd}^{t+1} - c_{bd+1}^{t+1})) \\ & + \sum_{b \in B} \sum_{t < T} u_{bD}^t (c_{bD}^t - c_{bD}^{t+1}) + \sum_{b \in B} \sum_{d < D} u_{bd}^T (c_{bd}^T - c_{bd+1}^T) + \sum_{b \in B} u_{bD}^T c_{bD}^T \end{aligned} \quad (3.52)$$

The **by constraint** of the mine production scheduling problem that creates a dependency between the variables across the consecutive time periods is written as:

$$w_{bd}^t \leq w_{bd}^{t+1} \quad \forall b \in B, t < T, d \in D$$

After the variable substitution, the new dual path double by constraint can be written as:

$$u_{bd}^t - u_{bd-1}^t \leq u_{bd}^{t+1} - u_{bd-1}^{t+1} \quad \forall b \in B, t < T, d \in D \quad (3.53)$$

The **sequencing constraint** of the mine production scheduling problem when formulated with by variable is:

$$\sum_{d \in D} w_{bd}^t \leq \sum_{d \in D} w_{b'd}^t \quad \forall b \in B, b' \in B_b$$

Since $u_{bd}^t = \sum_{d' \leq d} w_{bd'}^t$, the new sequencing constraint can be written as:

$$u_{bd}^t \leq u_{b'd}^t \quad \forall b \in B, b' \in B_b, t \in T \quad (3.54)$$

For the *capacity, grade blending, and risk constraints*, the variable substitution process will proceed with such a fashion that will construe “at” meaning to the mining decision variables under given conditions.

Substitute $w_{bd}^t = x_{bd}^t$ with:

- (1) u_{b1}^1 $\forall b \in B, \text{ iff } t = 1, d = 1$ as shown in eq (3.48)
(2) $u_{bd}^1 - u_{bd-1}^1$ $\forall b \in B, \text{ iff } t = 1, \forall d \in D \text{ iff } d \geq 2$ as shown in eq (3.49)

Substitute x_{bd}^t with:

- (3) $u_{bd}^t - u_{bd-1}^t - (u_{bd}^{t-1} - u_{bd-1}^{t-1})$ $\forall b \in B, \forall t \in T \text{ iff } t \geq 2, d \in D \text{ iff } d \geq 2$

Proof for:

- (3) $u_{bd}^t - u_{bd-1}^t - (u_{bd}^{t-1} - u_{bd-1}^{t-1}) = x_{bd}^t \quad \forall b \in B, \forall t \in T \text{ iff } t \geq 2, d \in D \text{ iff } d \geq 2$

Given $u_{bd}^t = \sum_{d' \leq d} w_{bd'}^t$ and $w_{bd}^t = \sum_{t' \leq t} x_{bd}^{t'}$;

Decompose " $u_{bd}^t - u_{bd-1}^t - (u_{bd}^{t-1} - u_{bd-1}^{t-1})$ " and write it in terms of " w_{bd}^t ":

$$\begin{aligned} u_{bd}^t - u_{bd-1}^t - (u_{bd}^{t-1} - u_{bd-1}^{t-1}) &= w_{bd}^t + w_{bd-1}^t - (w_{bd-1}^t + w_{bd-2}^t) \\ &\quad - (w_{bd}^{t-1} + w_{bd-1}^{t-1} - (w_{bd-1}^{t-1} + w_{bd-2}^{t-2})) \\ u_{bd}^t - u_{bd-1}^t - (u_{bd}^{t-1} - u_{bd-1}^{t-1}) &= w_{bd}^t + w_{bd-1}^t - w_{bd-1}^t - w_{bd-2}^t - w_{bd}^{t-1} - w_{bd-1}^{t-1} \\ &\quad + w_{bd-1}^{t-1} + w_{bd-2}^{t-2} \end{aligned}$$

$$u_{bd}^t - u_{bd-1}^t - (u_{bd}^{t-1} - u_{bd-1}^{t-1}) = w_{bd}^t - w_{bd}^{t-1}$$

Decompose " $w_{bd}^t - w_{bd}^{t-1}$ " and write it in terms of " x_{bd}^t ":

$$\begin{aligned} u_{bd}^t - u_{bd-1}^t - (u_{bd}^{t-1} - u_{bd-1}^{t-1}) &= x_{bd}^t + x_{bd}^{t-1} - x_{bd}^{t-1} \\ u_{bd}^t - u_{bd-1}^t - (u_{bd}^{t-1} - u_{bd-1}^{t-1}) &= x_{bd}^t \quad \forall t \in T \text{ iff } t \geq 2, d \in D \text{ iff } d \geq 2 \end{aligned}$$

which completes the proof.

- (4) $u_{b1}^t - u_{b1}^{t-1}$ $\forall t \in T \text{ iff } t \geq 2, d = 1$

Proof for:

- (4) $u_{b1}^t - u_{b1}^{t-1} = x_{b1}^t \quad \forall t \in T \text{ iff } t \geq 2, d = 1$

Since $w_{b1}^t = u_{b1}^t \quad \forall b \in B, t \in T, \text{ iff } d = 1$ as shown in eq (3.48);

$$u_{b1}^t - u_{b1}^{t-1} = w_{b1}^t - w_{b1}^{t-1} \quad \forall t \in T \text{ iff } t \geq 2, d = 1$$

We also know that $x_{bd}^t = w_{bd}^t - w_{bd}^{t-1} \quad \forall b \in B, d \in D, t \in T \text{ iff } t \geq 2$

as shown in eq (3.17); hence

$$u_{b1}^t - u_{b1}^{t-1} = x_{b1}^t \quad \forall b \in B, \forall t \in T \text{ iff } t \geq 2, d = 1$$

which completes the proof.

Given **mining capacity** constraints formulated with by variable w_{bd}^t such as:

$$\sum_{b \in B} \sum_{d \in D} p_b w_{bd}^1 \leq \bar{M}^1 \quad \text{iff } t = 1$$

$$\sum_{b \in B} \sum_{d \in D} p_b (w_{bd}^t - w_{bd}^{t-1}) \leq \bar{M}^t \quad \forall t \in T \text{ iff } t \geq 2$$

After the variable substitution, new capacity constraints can be written as:

$$\sum_{b \in B} p_b u_{b1}^1 + \sum_{b \in B} \sum_{2 \leq d \in D} p_b (u_{bd}^1 - u_{bd-1}^1) \leq \bar{M}^1 \quad \text{iff } t = 1 \quad (3.55)$$

$$\sum_{b \in B} p_b (u_{b1}^t - u_{b1}^{t-1}) + \sum_{b \in B} \sum_{2 \leq d \in D} p_b ((u_{bd}^t - u_{bd-1}^t) - (u_{bd}^{t-1} - u_{bd-1}^{t-1})) \leq \bar{M}^t \quad (3.56)$$

$\forall t \in T \text{ iff } t \geq 2$

Given **blending** constraints formulated with by variable w_{bd}^t such as:

$$\sum_{b \in B} (G_p^1 - g_b) p_b w_{bp}^1 \leq 0 \quad \forall p \in P, \text{ iff } t = 1$$

$$\sum_{b \in B} (G_p^t - g_b) p_b (w_{bp}^t - w_{bp}^{t-1}) \leq 0 \quad \forall p \in P, t \in T \text{ iff } t \geq 2$$

After the variable substitution, new blending constraints can be written as:

$$\sum_{b \in B} (G_1^1 - g_b) p_b u_{b1}^1 \leq 0 \quad \text{iff } p = 1 \in P, t = 1 \quad (3.57)$$

$$\sum_{b \in B} (G_p^1 - g_b) p_b (u_{bp}^1 - u_{bp-1}^1) \leq 0 \quad \forall p \in P \text{ iff } p \geq 2, t = 1 \quad (3.58)$$

$$\sum_{b \in B} (G_p^t - g_b) p_b ((u_{bp}^t - u_{bp-1}^t) - (u_{bp}^{t-1} - u_{bp-1}^{t-1})) \leq 0 \quad (3.59)$$

$\forall p \in P \text{ iff } p \geq 2, t \in T \text{ iff } t \geq 2$

$$\sum_{b \in B} (G_1^t - g_b) p_b (u_{b1}^t - u_{b1}^{t-1}) \leq 0 \quad \text{iff } p = 1 \in P, \forall t \in T \text{ iff } t \geq 2 \quad (3.60)$$

Given **risk** constraints formulated with by variable w_{bd}^t such as:

$$\sum_{b \in B} (r_{b1} - \bar{R}_1) p_b w_{b1}^1 \leq 0 \quad \text{iff } p = 1 \in P, t = 1$$

$$\sum_{p \in P} \sum_{b \in B} (r_{b1} - \bar{R}_1^t) p_b (w_{bp}^t - w_{bp}^{t-1}) \leq 0 \quad \forall t \in T \text{ iff } t \geq 2$$

After the variable substitution, new risk constraints can be written as

$$\sum_{b \in B} (r_{b1} - \bar{R}_1^1) p_b u_{b1}^1 \leq 0 \quad \text{iff } p = 1 \in P, t = 1 \quad (3.61)$$

$$\sum_{2 \leq p \in P} \sum_{b \in B} (r_{b1} - \bar{R}_1^1) p_b (u_{bp}^1 - u_{bp-1}^1) \leq 0 \quad \text{iff } t = 1, p \geq 2 \quad (3.62)$$

$$\sum_{2 \leq p \in P} \sum_{b \in B} (r_{b1} - \bar{R}_1^t) p_b ((u_{bp}^t - u_{bp-1}^t) - (u_{bp}^{t-1} - u_{bp-1}^{t-1})) \leq 0 \quad (3.63)$$

$\forall t \in T \text{ iff } t \geq 2, p \geq 2$

$$\sum_{b \in B} (r_{b1} - \bar{R}_1^t) p_b (u_{b1}^t - u_{b1}^{t-1}) \leq 0 \quad \text{iff } p = 1 \in P, \forall t \in T \text{ iff } t \geq 2 \quad (3.64)$$

A new mathematical model for the mine production scheduling problem formulated with new by variables is given as:

Objective Function:

$$\begin{aligned} \text{Max } Z = & \sum_{b \in B} \sum_{d < D} \sum_{t < T} u_{bd}^t ((c_{bd}^t - c_{bd+1}^t) - (c_{bd}^{t+1} - c_{bd+1}^{t+1})) \\ & + \sum_{b \in B} \sum_{t < T} u_{bd}^t (c_{bd}^t - c_{bd}^{t+1}) + \sum_{b \in B} \sum_{d < D} u_{bd}^T (c_{bd}^T - c_{bd+1}^T) + \sum_{b \in B} u_{bd}^T c_{bd}^T \\ & + \sum_{s \in S} \sum_{p \in P} v_{sp}^t y_{sp}^t \end{aligned} \quad (3.65)$$

Subject To:

By Destination Variable Constraint:

$$u_{bd}^t \leq u_{bd+1}^t \quad \forall b \in B, t \in T, d < D \quad (3.66)$$

Relationship Across the Time Periods:

$$u_{b1}^t \leq u_{b1}^{t+1} \quad \forall b \in B, t < T, d \in D \text{ iff } d = 1 \quad (3.67)$$

$$u_{bd}^t - u_{bd-1}^t \leq u_{bd}^{t+1} - u_{bd-1}^{t+1} \quad \forall b \in B, t < T, d \in D \text{ iff } d \geq 2 \quad (3.68)$$

Block Sequencing Constraint:

$$u_{bd}^t \leq u_{b'd}^t \quad \forall b \in B, b' \in B_b, t \in T, d \in D \quad (3.69)$$

Maximum Mining Capacity:

$$\sum_{b \in B} p_b u_{b1}^1 + \sum_{b \in B} \sum_{2 \leq d \in D} p_b (u_{bd}^1 - u_{bd-1}^1) \leq \bar{M}^1 \quad \text{iff } t = 1 \quad (3.70)$$

$$\sum_{b \in B} p_b (u_{b1}^t - u_{b1}^{t-1}) + \sum_{b \in B} \sum_{2 \leq d \in D} p_b ((u_{bd}^t - u_{bd-1}^t) - (u_{bd}^{t-1} - u_{bd-1}^{t-1})) \leq \bar{M}^t \quad (3.71)$$

$\forall t \in T \text{ iff } t \geq 2$

Maximum Capacity at Process Destination:

$$\sum_{b \in B} p_b u_{b1}^1 + \sum_{s \in S} y_{s1}^1 \leq \bar{C}_1 \quad \text{iff } t = 1, \text{ iff } p = 1 \in P \quad (3.72)$$

$$\sum_{b \in B} p_b (u_{bp}^1 - u_{bp-1}^1) + \sum_{s \in S} y_{sp}^1 \leq \bar{C}_p \quad \text{iff } t = 1, \forall p \in P \text{ iff } p \geq 2 \quad (3.73)$$

$$\sum_{b \in B} p_b ((u_{bp}^t - u_{bp-1}^t) - (u_{bp}^{t-1} - u_{bp-1}^{t-1})) + \sum_{s \in S} y_{sp}^t \leq \bar{C}_p \quad (3.74)$$

$\forall t \in T \text{ iff } t \geq 2, p \in P \text{ iff } p \geq 2$

$$\sum_{b \in B} p_b (u_{b1}^t - u_{b1}^{t-1}) + \sum_{s \in S} y_{s1}^t \leq \bar{C}_1 \quad \forall t \in T \text{ iff } t \geq 2, \text{ iff } p = 1 \in P \quad (3.75)$$

Maximum Capacity at Waste Dump Destination:

$$\sum_{b \in B} p_b u_{b1}^1 \leq \bar{W}_1 \quad \text{iff } t = 1, \text{ iff } w = 1 \in W \quad (3.76)$$

$$\sum_{b \in B} p_b (u_{bw}^1 - u_{bw-1}^1) \leq \bar{W}_w \quad \text{iff } t = 1, \forall w \in W \text{ iff } w \geq 2 \quad (3.77)$$

$$\sum_{b \in B} p_b ((u_{bw}^t - u_{bw-1}^t) - (u_{bw}^{t-1} - u_{bw-1}^{t-1})) \leq \bar{W}_w^t \quad (3.78)$$

$\forall t \in T \text{ iff } t \geq 2, w \in W \text{ iff } w \geq 2$

$$\sum_{b \in B} p_b (u_{b1}^t - u_{b1}^{t-1}) \leq \bar{W}_1^t \quad \forall t \in T \text{ iff } t \geq 2, \text{ iff } w = 1 \in W \quad (3.79)$$

Minimum Grade Blending Requirement at Process Destination:

$$\sum_{b \in B} (\underline{G}_1^1 - g_b) p_b u_{b1}^1 + \sum_{s \in S} (\underline{G}_1^1 - g_s) y_{s1}^1 \leq 0 \quad \text{iff } p = 1 \in P, \text{ iff } t = 1 \quad (3.80)$$

$$\sum_{b \in B} (\underline{G}_p^1 - g_b) p_b (u_{bp}^1 - u_{bp-1}^1) + \sum_{s \in S} (\underline{G}_p^1 - g_s) y_{sp}^1 \leq 0 \quad \forall p \in P \text{ iff } p \geq 2, t = 1 \quad (3.81)$$

$$\sum_{b \in B} (\underline{G}_p^t - g_b) p_b \left((u_{bp}^t - u_{bp-1}^t) - (u_{bp}^{t-1} - u_{bp-1}^{t-1}) \right) + \sum_{s \in S} (\underline{G}_p^t - g_s) y_{sp}^t \leq 0 \quad (3.82)$$

$$\forall p \in P \text{ iff } p \geq 2, t \in T \text{ iff } t \geq 2$$

$$\sum_{b \in B} (\underline{G}_1^t - g_b) p_b (u_{b1}^t - u_{b1}^{t-1}) + \sum_{s \in S} (\underline{G}_1^t - g_s) y_{s1}^t \leq 0 \quad (3.83)$$

$$\text{iff } p = 1 \in P, \forall t \in T \text{ iff } t \geq 2$$

Maximum Proportion of Inferred Material at Process Destination:

$$\sum_{b \in B} (r_{b1} - \bar{R}_1^1) p_b u_{b1}^1 + \sum_{s_1 \in S_{R_1}} y_{s_1 1}^1 (1 - \bar{R}_1^1) - \sum_{s \neq s_1 \in S} \bar{R}_1^1 y_{s1}^1 \leq 0 \quad (3.84)$$

$$\text{iff } p = 1 \in P, \text{ iff } t = 1$$

$$\sum_{2 \leq p \in P} \sum_{b \in B} (r_{b1} - \bar{R}_1^1) p_b (u_{bp}^1 - u_{bp-1}^1) + \sum_{2 \leq p \in P} \sum_{s_1 \in S_{R_1}} y_{s_1 p}^1 (1 - \bar{R}_1^1) \quad (3.85)$$

$$- \sum_{2 \leq p \in P} \sum_{s \neq s_1 \in S} \bar{R}_1^1 y_{sp}^1 \leq 0 \quad \text{iff } t = 1, p \geq 2$$

$$\sum_{2 \leq p \in P} \sum_{b \in B} (r_{b1} - \bar{R}_1^t) p_b \left((u_{bp}^t - u_{bp-1}^t) - (u_{bp}^{t-1} - u_{bp-1}^{t-1}) \right) \quad (3.86)$$

$$+ \sum_{2 \leq p \in P} \sum_{s_1 \in S_{R_1}} y_{s_1 p}^t (1 - \bar{R}_1^t) - \sum_{2 \leq p \in P} \sum_{s \neq s_1 \in S} \bar{R}_1^t y_{sp}^t \leq 0 \quad \forall t \in T \text{ iff } t \geq 2, p \geq 2$$

$$\sum_{b \in B} (r_{b1} - \bar{R}_1^t) p_b (u_{b1}^t - u_{b1}^{t-1}) + \sum_{s_1 \in S_{R_1}} y_{s_1 1}^t (1 - \bar{R}_1^t) - \sum_{s \neq s_1 \in S} \bar{R}_1^t y_{s1}^t \leq 0 \quad (3.87)$$

$$\text{iff } p = 1 \in P, \forall t \in T \text{ iff } t \geq 2$$

Minimum Proportion of Indicated Material at Process Destination:

$$\sum_{b \in B} (\underline{R}_2^1 - r_{b2}) p_b u_{b1}^1 - \sum_{s_2 \in S_{R_2}} y_{s_2 1}^1 (\underline{R}_2^1 - 1) + \sum_{s \neq s_2 \in S} \underline{R}_2^1 y_{s1}^1 \leq 0 \quad (3.88)$$

$$\text{iff } p = 1 \in P, \text{ iff } t = 1$$

$$\begin{aligned}
& \sum_{2 \leq p \in P} \sum_{b \in B} (\underline{R}_2^1 - r_{b2}) p_b (u_{bp}^1 - u_{bp-1}^1) - \sum_{2 \leq p \in P} \sum_{s_2 \in S_{R_2}} y_{s_2 p}^1 (\underline{R}_2^1 - 1) \\
& \quad + \sum_{2 \leq p \in P} \sum_{s \neq s_2 \in S} \underline{R}_2^1 y_{sp}^1 \leq 0 \quad \text{iff } p \geq 2, t = 1
\end{aligned} \tag{3.89}$$

$$\begin{aligned}
& \sum_{2 \leq p \in P} \sum_{b \in B} (\underline{R}_2^t - r_{b2}) p_b \left((u_{bp}^t - u_{bp-1}^t) - (u_{bp}^{t-1} - u_{bp-1}^{t-1}) \right) \\
& \quad - \sum_{2 \leq p \in P} \sum_{s_2 \in S_{R_2}} y_{s_2 p}^t (\underline{R}_2^t - 1) + \sum_{2 \leq p \in P} \sum_{s \neq s_2 \in S} \underline{R}_2^t y_{sp}^t \leq 0 \quad \forall t \in T \text{ iff } t \geq 2, p \geq 2
\end{aligned} \tag{3.90}$$

$$\begin{aligned}
& \sum_{b \in B} (\underline{R}_2^t - r_{b2}) p_b (u_{b1}^t - u_{b1}^{t-1}) - \sum_{s_2 \in S_{R_2}} y_{s_2 1}^t (\underline{R}_2^t - 1) + \sum_{s \neq s_2 \in S} \underline{R}_2^t y_{s1}^t \leq 0 \\
& \quad \text{iff } p = 1 \in P, \forall t \in T \text{ iff } t \geq 2
\end{aligned} \tag{3.91}$$

Minimum Proportion of Measured Material at Process Destination:

$$\begin{aligned}
& \sum_{b \in B} (\underline{R}_3^1 - r_{b3}) p_b u_{bp}^1 - \sum_{s_3 \in S_{R_3}} y_{s_3 1}^1 (\underline{R}_3^1 - 1) + \sum_{s \neq s_3 \in S} \underline{R}_3^1 y_{s1}^1 \leq 0 \\
& \quad \text{iff } p = 1 \in P, \text{ iff } t = 1
\end{aligned} \tag{3.92}$$

$$\begin{aligned}
& \sum_{2 \leq p \in P} \sum_{b \in B} (\underline{R}_3^1 - r_{b3}) p_b (u_{bp}^1 - u_{bp-1}^1) - \sum_{2 \leq p \in P} \sum_{s_3 \in S_{R_3}} y_{s_3 p}^1 (\underline{R}_3^1 - 1) \\
& \quad + \sum_{2 \leq p \in P} \sum_{s \neq s_3 \in S} \underline{R}_3^1 y_{sp}^1 \leq 0 \quad \text{iff } p \geq 2, t = 1
\end{aligned} \tag{3.93}$$

$$\begin{aligned}
& \sum_{2 \leq p \in P} \sum_{b \in B} (\bar{R}_3^t - r_{b3}) p_b \left((u_{bp}^t - u_{bp-1}^t) - (u_{bp}^{t-1} - u_{bp-1}^{t-1}) \right) \\
& \quad - \sum_{2 \leq p \in P} \sum_{s_3 \in S_{R_3}} y_{s_3 p}^t (\bar{R}_3^t - 1) + \sum_{2 \leq p \in P} \sum_{s \neq s_3 \in S} \bar{R}_3^t y_{sp}^t \leq 0 \quad \forall t \in T \text{ iff } t \geq 2, p \geq 2
\end{aligned} \tag{3.94}$$

$$\begin{aligned}
& \sum_{b \in B} (\bar{R}_3^t - r_{b3}) p_b (u_{b1}^t - u_{b1}^{t-1}) - \sum_{s_3 \in S_{R_3}} y_{s_3 1}^t (\bar{R}_3^t - 1) + \sum_{s \neq s_3 \in S} \bar{R}_3^t y_{s1}^t \leq 0 \\
& \quad \text{iff } p = 1 \in P, \forall t \in T \text{ iff } t \geq 2
\end{aligned} \tag{3.95}$$

Stockpile Balance Constraint:

$$\sum_{b \in B} p_b u_{b1}^1 - \sum_{p \in P} y_{1p}^1 + C_s^0 \leq C_1^1 \quad \text{iff } s = 1 \in S, \text{iff } t = 1 \quad (3.96)$$

$$\sum_{b \in B} p_b (u_{bs}^1 - u_{bs-1}^1) - \sum_{p \in P} y_{1p}^1 + C_s^0 \leq C_s^1 \quad \forall s \in S \text{ iff } s \geq 2, \text{iff } t = 1 \quad (3.97)$$

$$\sum_{b \in B} p_b \left((u_{bs}^t - u_{bs-1}^t) - (u_{bs}^{t-1} - u_{bs-1}^{t-1}) \right) - \sum_{p \in P} y_{sp}^t + C_s^{t-1} \leq C_s^t \quad (3.98)$$

$$\forall s \in S \text{ iff } s \geq 2, t \in T \text{ iff } t \geq 2$$

$$\sum_{b \in B} p_b (u_{b1}^t - u_{b1}^{t-1}) - \sum_{p \in P} y_{1p}^t + C_1^{t-1} \leq C_1^t \quad \text{iff } s = 1 \in S, \forall t \in T \text{ iff } t \geq 2 \quad (3.99)$$

Variable Restrictions:

$$y_{sp}^t \geq 0 \quad \forall s \in S, t \in T, p \in D \quad (3.100)$$

$$C_s^t \geq 0 \quad \forall s \in S, t \in T \quad (3.101)$$

$$u_{bd}^t \in [0, 1] \quad \forall b \in B, t \in T, d \in D \quad (3.102)$$

3.4 Variable Substitution for Single Path Double “By” Variables

In this section, it will be shown how to derive the single path double “by” variable z_{bd}^t which was originally introduced by Bienstock Zuckerberg (2009,2010 and 2015), by making a variable substitution for the at variable x_{bd}^t presented in section 3.1. It is also important to highlight the similarities and the differences between the new dual path double “by” variable u_{bd}^t and the single path double “by” variable z_{bd}^t since the new integer solution algorithm can be modeled by using either one of the variables. In Figure 3.4, the node structure of the variables clearly shows that at time period t, both variables have the consecutive destinations connected on a single path. The differences become apparent across the time periods where the new variable u_{bd}^t connects the individual destination nodes from period t to t+1 whereas BZ variable z_{bd}^t ensures that each consecutive destination node is aligned on a single path from period t to t+1. In other words, the variable z_{bd}^t accomplishes the interaction across the time periods by connecting the final destination D of period t to the first destination of period t+1. This property reduces the complexity of the mathematical model in comparison with the model defined by the new variable u_{bd}^t and also

results with less number of sequencing and reserve constraints in the master problem. The mathematical formulation derived using the set of variables z_{bd}^t to model the master problem of the BZ algorithm and the new integer solution algorithm will be demonstrated on a small 2D example in the next Chapter.

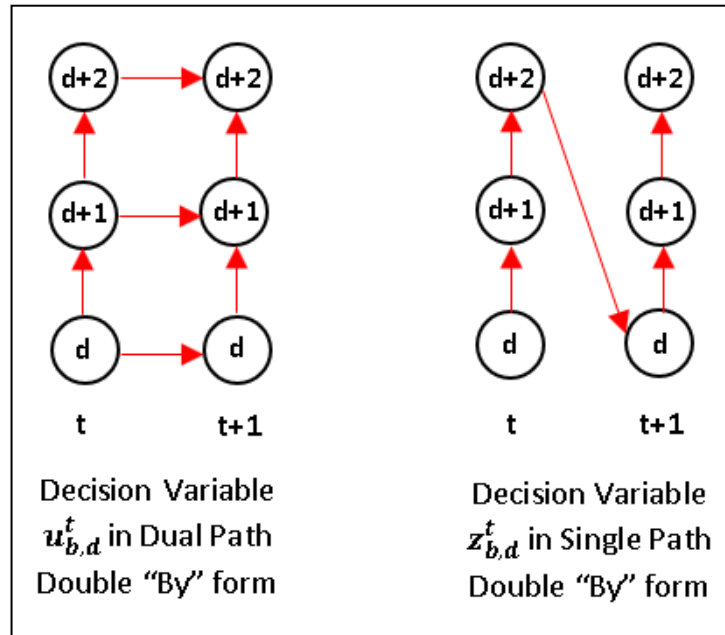


Figure 3.4 : Node structure of the dual path double "by" variables and the single path double "by" variables

In other words, if there are two blocks to be sequenced for three time periods and three destinations, as shown in Figure 3.5a, one would need 9 variables to be used in sequencing equations shown in 3.5b. If one does variable substitution as shown in 3.5c, the new sequencing equations as shown in 3.5d are obtained. It is clear that the new set of sequencing constraints are much simplified as the number of variables per constraint are significantly reduced compared to the sequencing constraints shown in 3.5b.

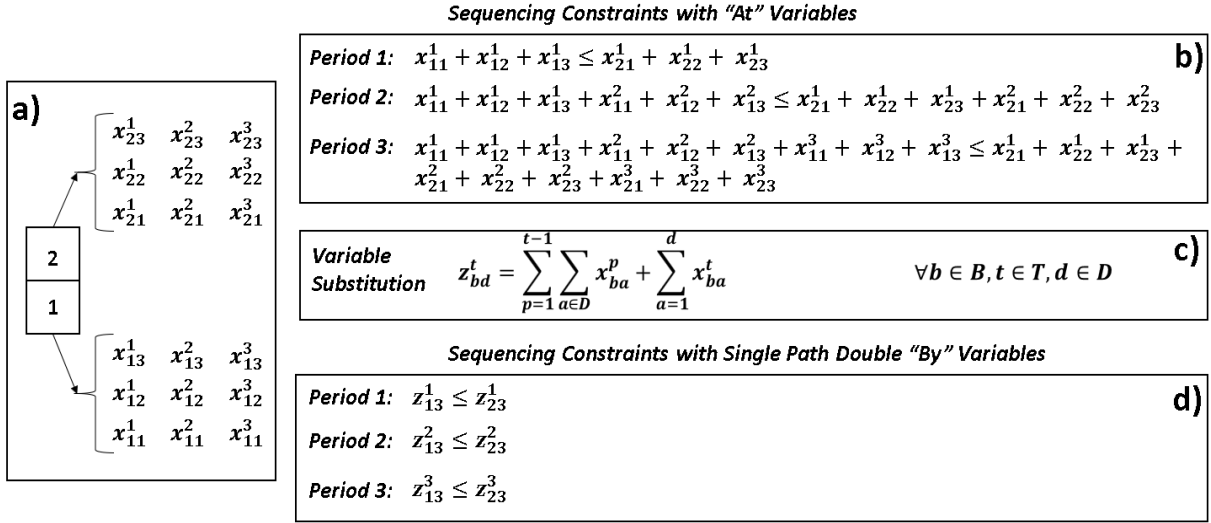


Figure 3.5 : Example showing variable reduction process using “ z_{bd}^t ” variable substitution

Decision Variables:

z_{bd}^t = the fraction of block b , that is sent by destination d , by time period t

The following shows the relationship between the at variable “ x_{bd}^t ” and the single path double “by” variable “ z_{bd}^t ”.

$$z_{bd}^t = \sum_{p=1}^{t-1} \sum_{a \in D} x_{ba}^p + \sum_{a=1}^d x_{ba}^t \quad \forall b \in B, t \in T, d \in D \quad (3.103)$$

$$x_{b1}^1 = z_{b1}^1 \quad \forall b \in B, \text{ iff } d = 1, t = 1 \quad (3.104)$$

$$x_{bd}^t = z_{bd}^t - z_{bd}^{t-1} \quad \forall b \in B, t \in T, d \in D \text{ iff } d \geq 2 \quad (3.105)$$

$$x_{b1}^t = z_{b1}^t - z_{bD}^{t-1} \quad \forall b \in B, \text{ iff } d = 1, t \in T \text{ iff } t \geq 2 \quad (3.106)$$

As shown before the **objective function** of the mine production scheduling problem excluding the stockpiles when formulated with “at” variable is:

$$\text{Max } Z = \sum_{b \in B} \sum_{d \in D} \sum_{t \in T} c_{bd}^t x_{bd}^t$$

Next, the at variable “ x_{bd}^t ” will be replaced with single path double “by” variable “ z_{bd}^t ” as shown below:

Let's rewrite the objective function $\text{Max } Z = \sum_{b \in B} \sum_{d \in D} \sum_{t \in T} c_{bd}^t x_{bd}^t$ as:

$$\begin{aligned}
Max Z = & \sum_{b \in B} c_{b1}^1 x_{b1}^1 + c_{b2}^1 x_{b2}^1 + \dots \dots + c_{bD-1}^1 x_{bD-1}^1 + c_{bD}^1 x_{bD}^1 + c_{b1}^2 x_{b1}^2 + c_{b2}^2 x_{b2}^2 \\
& + \dots \dots + c_{bD-1}^2 x_{bD-1}^2 + c_{bD}^2 x_{bD}^2 + \dots \dots + c_{b1}^T x_{b1}^T + c_{b2}^T x_{b2}^T \\
& + \dots \dots + c_{bD-1}^T x_{bD-1}^T + c_{bD}^T x_{bD}^T
\end{aligned}$$

Substitute x_{bd}^t with z_{bd}^t according to the rules given by eq 3.104, 3.105, 3.106:

$$\begin{aligned}
Max Z = & \sum_{b \in B} c_{b1}^1 z_{b1}^1 + c_{b2}^1 (z_{b2}^1 - z_{b1}^1) + \dots \dots + c_{bD-1}^1 (z_{bD-1}^1 - z_{bD-2}^1) + c_{bD}^1 (z_{bD}^1 - z_{bD-1}^1) \\
& + c_{b1}^2 (z_{b1}^2 - z_{bD}^1) + c_{b2}^2 (z_{b2}^2 - z_{b1}^2) + \dots \dots + c_{bD-1}^2 (z_{bD-1}^2 - z_{bD-2}^2) \\
& + c_{bD}^2 (z_{bD}^2 - z_{bD-1}^2) + \dots \dots + c_{b1}^T (z_{b1}^T - z_{bD}^{T-1}) + c_{b2}^T (z_{b2}^T - z_{b1}^T) \\
& + \dots \dots + c_{bD-1}^T (z_{bD-1}^T - z_{bD-2}^T) + c_{bD}^T (z_{bD}^T - z_{bD-1}^T)
\end{aligned}$$

$$\begin{aligned}
Max Z = & \sum_{b \in B} c_{b1}^1 z_{b1}^1 + c_{b2}^1 z_{b2}^1 - c_{b2}^1 z_{b1}^1 + \dots \dots + c_{bD-1}^1 z_{bD-1}^1 - c_{bD-1}^1 z_{bD-2}^1 \\
& + c_{bD}^1 z_{bD}^1 - c_{bD}^1 z_{bD-1}^1 + c_{b1}^2 z_{b1}^2 - c_{b1}^2 z_{bD}^1 + c_{b2}^2 z_{b2}^2 - c_{b2}^2 z_{b1}^2 \\
& + \dots \dots + c_{bD-1}^2 z_{bD-1}^2 - c_{bD-1}^2 z_{bD-2}^2 + c_{bD}^2 z_{bD}^2 - c_{bD}^2 z_{bD-1}^2 \\
& + \dots \dots + c_{b1}^T z_{b1}^T - c_{b1}^T z_{bD}^{T-1} + c_{b2}^T z_{b2}^T - c_{b2}^T z_{b1}^T \\
& + \dots \dots + c_{bD-1}^T z_{bD-1}^T - c_{bD-1}^T z_{bD-2}^T + c_{bD}^T z_{bD}^T - c_{bD}^T z_{bD-1}^T
\end{aligned}$$

Rearrange the z_{bd}^t variables:

$$\begin{aligned}
Max Z = & \sum_{b \in B} z_{b1}^1 (c_{b1}^1 - c_{b2}^1) + z_{b2}^1 (c_{b2}^1 - c_{b3}^1) + \dots \dots + z_{bD-1}^1 (c_{bD-1}^1 - c_{bD}^1) \\
& + z_{bD}^1 (c_{bD}^1 - c_{b1}^2) + z_{b1}^2 (c_{b1}^2 - c_{b2}^2) + z_{b2}^2 (c_{b2}^2 - c_{b3}^2) \\
& + \dots \dots + z_{bD-1}^2 (c_{bD-1}^2 - c_{bD}^2) + z_{bD}^2 (c_{bD}^2 - c_{b1}^3) \\
& + \dots \dots + z_{bD}^{T-1} (c_{bD}^{T-1} - c_{b1}^T) + z_{b1}^T (c_{b1}^T - c_{b2}^T) + z_{b2}^T (c_{b2}^T - c_{b3}^T) \\
& + \dots \dots + z_{bD-1}^T (c_{bD-1}^T - c_{bD}^T) + z_{bD}^T c_{bD}^T
\end{aligned}$$

This can be simplified as:

$$Max Z = \sum_{d < D} \sum_{t \in T} z_{bd}^t (c_{bd}^t - c_{bd+1}^t) + \sum_{t < T} z_{bD}^t (c_{bD}^t - c_{b1}^{t+1}) + z_{bD}^T c_{bD}^T \quad (3.107)$$

The **sequencing constraint** of the mine production scheduling problem when formulated with “at” variable is:

$$\sum_{t' \leq t} \sum_{d \in D} x_{bd}^{t'} \leq \sum_{t' \leq t} \sum_{d \in D} x_{b'd}^{t'} \quad \forall b \in B, b' \in B_b, t \in T$$

We know that:

$$z_{bD}^t = \sum_{t' \leq t} \sum_{a \in D} x_{ba}^{t'} \text{ therefore, the new sequencing constraint can be written as:}$$

$$z_{bD}^t \leq z_{b'D}^{t'} \quad \forall b \in B, b' \in B_b, t \in T \quad (3.108)$$

For the **capacity, grade blending, and risk constraints**, the variable conversion can be done straightforward as shown below:

Given mining capacity constraints formulated with at variables such as;

$$\sum_{b \in B} \sum_{d \in D} p_b x_{bd}^t \leq \bar{M}^t \quad \forall t \in T,$$

the mining capacity constraints after the variable substitution will be:

$$\sum_{b \in B} p_b z_{b1}^1 + \sum_{b \in B} \sum_{2 \leq d \in D} p_b (z_{bd}^1 - z_{bd-1}^1) \leq \bar{M}^1 \quad \text{iff } t = 1 \quad (3.109)$$

$$\sum_{b \in B} \sum_{2 \leq d \in D} p_b (z_{bd}^t - z_{bd-1}^t) + \sum_{b \in B} p_b (z_{b1}^t - z_{bD}^{t-1}) \leq \bar{M}^t \quad \forall t \in T \text{ iff } t \geq 2 \quad (3.110)$$

Likewise, for the blending constraints:

$$\text{At formulation } \rightarrow \sum_{b \in B} (G_p^t - g_b) p_b x_{bd}^t \leq 0 \quad \forall t \in T, p \in P$$

$$\sum_{b \in B} (G_1^1 - g_b) p_b z_{b1}^1 \leq 0 \quad \text{iff } p = 1 \in P, t = 1 \quad (3.111)$$

$$\sum_{b \in B} (G_p^t - g_b) (z_{bp}^t - z_{bp-1}^t) p_b \leq 0 \quad \forall t \in T, p \in P \text{ iff } p \geq 2 \quad (3.112)$$

$$\sum_{b \in B} (G_1^t - g_b) (z_{b1}^t - z_{bD}^{t-1}) p_b \leq 0 \quad \text{iff } p = 1, \forall t \in T \text{ iff } t \geq 2 \quad (3.113)$$

For the risk constraints:

$$\text{At formulation } \rightarrow \sum_{b \in B} \sum_{p \in P} (r_{b1} - \bar{R}_1^t) p_b x_{bp}^t \leq 0 \quad \forall t \in T$$

$$\sum_{b \in B} (r_{b1} - \bar{R}_1^1) p_b z_{b1}^1 \leq 0 \quad \text{iff } p = 1 \in P, t = 1 \quad (3.114)$$

$$\sum_{2 \leq p \in P} \sum_{b \in B} (r_{b1} - \bar{R}_1^t) p_b (z_{bp}^t - z_{bp-1}^t) \leq 0 \quad \forall t \in T, p \geq 2 \quad (3.115)$$

$$\sum_{b \in B} (r_{b1} - \bar{R}_1) p_b (z_{b1}^t - z_{bD}^{t-1}) \leq 0 \quad \text{iff } p = 1, \forall t \in T \text{ iff } t \geq 2 \quad (3.116)$$

Mathematical model for the mine production scheduling problem formulated with single path double “by” variables is given as:

Objective Function:

$$\begin{aligned} \text{Max } Z = & \sum_{d < D} \sum_{t \in T} z_{bd}^t (c_{bd}^t - c_{bd+1}^t) + \sum_{t < T} z_{bD}^t (c_{bD}^t - c_{b1}^{t+1}) + z_{bD}^T c_{bD}^T \\ & + \sum_{s \in S} \sum_{p \in P} v_{sp}^t y_{sp}^t \end{aligned} \quad (3.117)$$

Subject To:

By Destination Variable Constraint:

$$z_{bd}^t \leq z_{bd+1}^t \quad \forall b \in B, t \in T, d < D \quad (3.118)$$

$$z_{bD}^t \leq z_{b1}^{t+1} \quad \forall b \in B, t < T, d = D \quad (3.119)$$

Block Sequencing Constraint:

$$z_{bD}^t \leq z_{b'D}^t \quad \forall b \in B, b' \in B_b, t \in T \quad (3.120)$$

Maximum Mining Capacity:

$$\sum_{b \in B} p_b z_{b1}^1 + \sum_{b \in B} \sum_{2 \leq d \in D} p_b (z_{bd}^1 - z_{bd-1}^1) \leq \bar{M}^1 \quad \text{iff } t = 1 \quad (3.121)$$

$$\sum_{b \in B} \sum_{2 \leq d \in D} p_b (z_{bd}^t - z_{bd-1}^t) + \sum_{b \in B} p_b (z_{b1}^t - z_{bD}^{t-1}) \leq \bar{M}^t \quad \forall t \in T \text{ iff } t \geq 2 \quad (3.122)$$

Maximum Capacity at Process Destination:

$$\sum_{b \in B} p_b z_{b1}^1 + \sum_{s \in S} y_{s1}^1 \leq \bar{C}_1 \quad \text{iff } t = 1, \text{ iff } p = 1 \in P \quad (3.123)$$

$$\sum_{b \in B} p_b (z_{bp}^t - z_{bp-1}^t) + \sum_{s \in S} y_{sp}^t \leq \bar{C}_p \quad \forall t \in T, \forall p \in P \text{ iff } p \geq 2 \quad (3.124)$$

$$\sum_{b \in B} p_b (z_{b1}^t - z_{bD}^{t-1}) + \sum_{s \in S} y_{s1}^t \leq \bar{C}_1 \quad \forall t \in T \text{ iff } t \geq 2, \text{ iff } p = 1 \in P \quad (3.125)$$

Maximum Capacity at Waste Dump Destination:

$$\sum_{b \in B} p_b z_{b1}^1 + \sum_{s \in S} y_{s1}^1 \leq \bar{W}_1 \quad \text{iff } t = 1, \text{ iff } w = 1 \in W \quad (3.126)$$

$$\sum_{b \in B} p_b(z_{bw}^t - z_{bw-1}^t) + \sum_{s \in S} y_{sw}^t \leq \bar{W}_w^t \quad \forall t \in T, \forall w \in W \text{ iff } w \geq 2 \quad (3.127)$$

$$\sum_{b \in B} p_b(z_{b1}^t - z_{bD}^{t-1}) + \sum_{s \in S} y_{s1}^t \leq \bar{W}_1^t \quad \forall t \in T \text{ iff } t \geq 2, \text{ iff } w = 1 \in W \quad (3.128)$$

Minimum Grade Blending Requirement at Process Destination:

$$\sum_{b \in B} (\underline{G}_1^1 - g_b)p_b z_{b1}^1 + \sum_{s \in S} (\underline{G}_1^1 - g_s)y_{s1}^1 \leq 0 \quad \text{iff } p = 1 \in P, \text{ iff } t = 1 \quad (3.129)$$

$$\sum_{b \in B} (\underline{G}_p^t - g_b)p_b(z_{bp}^t - z_{bp-1}^t) + \sum_{s \in S} (\underline{G}_p^t - g_s)y_{sp}^t \leq 0 \quad \forall t \in T, \forall p \in P \text{ iff } p \geq 2 \quad (3.130)$$

$$\sum_{b \in B} (\underline{G}_1^t - g_b)p_b(z_{b1}^t - z_{bD}^{t-1}) + \sum_{s \in S} (\underline{G}_1^t - g_s)y_{s1}^t \leq 0 \quad \forall t \in T \text{ iff } t \geq 2, \text{ iff } p = 1 \in P \quad (3.131)$$

Maximum Proportion of Inferred Material at Process Destination:

$$\sum_{b \in B} (r_{b1} - \bar{R}_1^1)p_b z_{b1}^1 + \sum_{s_1 \in S_{R_1}} y_{s_1 1}^1 - \sum_{s \neq s_1 \in S} \bar{R}_1^1 y_{s1}^1 \leq 0 \quad \text{iff } p = 1 \in P, t = 1 \quad (3.132)$$

$$\sum_{2 \leq p \in P} \sum_{b \in B} (r_{b1} - \bar{R}_1^t)p_b(z_{bp}^t - z_{bp-1}^t) + \sum_{2 \leq p \in P} \sum_{s_1 \in S_{R_1}} y_{s_1 p}^t - \sum_{2 \leq p \in P} \sum_{s \neq s_1 \in S} \bar{R}_1^t y_{sp}^t \leq 0 \quad \forall t \in T, p \geq 2 \quad (3.133)$$

$$\sum_{b \in B} (r_{b1} - \bar{R}_1^t)p_b(z_{b1}^t - z_{bD}^{t-1}) + \sum_{s_1 \in S_{R_1}} y_{s_1 1}^t - \sum_{s \neq s_1 \in S} \bar{R}_1^t y_{s1}^t \leq 0 \quad \forall t \in T \text{ iff } t \geq 2, \text{ iff } p = 1 \in P \quad (3.134)$$

Minimum Proportion of Indicated Material at Process Destination:

$$\sum_{b \in B} (\underline{R}_2^1 - r_{b2})p_b z_{b1}^1 - \sum_{s_2 \in S_{R_2}} y_{s_2 1}^1 + \sum_{s \neq s_2 \in S} \underline{R}_2^1 y_{s1}^1 \leq 0 \quad \text{iff } p = 1 \in P, t = 1 \quad (3.135)$$

$$\sum_{2 \leq p \in P} \sum_{b \in B} (\underline{R}_2^t - r_{b2})p_b(z_{bp}^t - z_{bp-1}^t) - \sum_{2 \leq p \in P} \sum_{s_2 \in S_{R_2}} y_{s_2 p}^t + \sum_{2 \leq p \in P} \sum_{s \neq s_2 \in S} \underline{R}_2^t y_{sp}^t \leq 0 \quad \forall t \in T, p \geq 2 \quad (3.136)$$

$$\sum_{b \in B} (\underline{R}_2^t - r_{b2}) p_b (z_{b1}^t - z_{bD}^{t-1}) - \sum_{s_2 \in S_{R_2}} y_{s_2 1}^t + \sum_{s \neq s_2 \in S} \bar{R}_2^t y_{s1}^t \leq 0 \quad (3.137)$$

$$\forall t \in T \text{ iff } t \geq 2, \text{ iff } p = 1 \in P$$

Minimum Proportion of Measured Material at Process Destination:

$$\sum_{b \in B} (\underline{R}_3^1 - r_{b3}) p_b z_{b1}^1 - \sum_{s_3 \in S_{R_3}} y_{s_3 1}^1 + \sum_{s \neq s_3 \in S} \underline{R}_3^1 y_{s1}^1 \leq 0 \quad \text{iff } p = 1 \in P, t = 1 \quad (3.138)$$

$$\sum_{2 \leq p \in P} \sum_{b \in B} (\underline{R}_3^t - r_{b3}) p_b (z_{bp}^t - z_{bp-1}^t) - \sum_{2 \leq p \in P} \sum_{s_3 \in S_{R_3}} y_{s_3 p}^t + \sum_{2 \leq p \in P} \sum_{s \neq s_3 \in S} \underline{R}_3^t y_{sp}^t \leq 0 \quad (3.139)$$

$$\forall t \in T, p \geq 2$$

$$\sum_{b \in B} (\underline{R}_3^t - r_{b3}) p_b (z_{b1}^t - z_{bD}^{t-1}) - \sum_{s_3 \in S_{R_3}} y_{s_3 1}^t + \sum_{s \neq s_3 \in S} \underline{R}_3^t y_{s1}^t \leq 0 \quad (3.140)$$

$$\forall t \in T \text{ iff } t \geq 2, \text{ iff } p = 1 \in P$$

Stockpile Balance Constraint:

$$\sum_{b \in B} p_b z_{b1}^1 - \sum_{p \in P} y_{1p}^1 + C_1^0 \leq C_1^1 \quad \text{iff } s = 1 \in S, t = 1 \quad (3.141)$$

$$\sum_{b \in B} p_b (z_{bs}^t - z_{bs-1}^t) - \sum_{p \in P} y_{1p}^t + C_s^0 \leq C_s^t \quad \forall s \in S \text{ iff } s \geq 2, \text{ iff } t = 1 \quad (3.142)$$

$$\sum_{b \in B} p_b (z_{b1}^t - z_{bD}^{t-1}) - \sum_{p \in P} y_{sp}^t + C_1^{t-1} \leq C_1^t \quad \forall t \in T \text{ iff } t \geq 2, \text{ iff } s = 1 \in S \quad (3.143)$$

Variable Restrictions:

$$y_{sp}^t \geq 0 \quad \forall s \in S, t \in T, p \in D \quad (3.144)$$

$$C_s^t \geq 0 \quad \forall s \in S, t \in T \quad (3.145)$$

$$z_{bd}^t \in [0, 1] \quad \forall b \in B, t \in T, d \in D \quad (3.146)$$

CHAPTER 4.

NEW INTEGER SOLUTION ALGORITHM TO SOLVE MULTI-DESTINATION OPEN PIT MINE PRODUCTION SCHEDULING PROBLEM

In this chapter, the new integer solution algorithm that can solve the mine production scheduling problems modeled with multi capacities, grade blending, grade uncertainty, stockpiles, variable pit slopes, multi destinations and truck hours will be covered extensively. It should be stressed that the blocks will not have any pre-determined destinations based on grades, cycle times, material type or some other criteria since the best destination selection per block will be done automatically during the optimization process to maximize the NPV. The mathematical model for the said problem is generated at the end of the previous chapter. A novel solution algorithm will be introduced to solve this complex model. The new solution algorithm will exploit the mathematical theories behind the Bienstock-Zuckerberg decomposition algorithm which is presently the fastest solution algorithm for the precedence constrained production scheduling type problems such as mine production scheduling problems. In order to set the groundwork, the BZ algorithm will be covered in detail. Then, a comprehensive study on the new integer solution algorithm will be presented. Discussions will be given to ensure the working mechanism of the algorithm functions within the frame of mathematical theories.

The mine production scheduling problem is an integer programming problem solved for mining decision variables in a binary form which constitutes the time period when the block is mined and the destination to which the block is sent. So far there is no method that can generate a theoretically proven optimal integer solution to the large-scale integer programming type problems, including mine production scheduling problems due to the exponential computational complexity of the existing exact algorithms. Nevertheless, a theoretical upper bound can be achieved if the linear relaxation of the mine production scheduling problem can be solved to a proven optimality. Henceforth, the knowledge of the optimal solution to a linear relaxation of the mine production scheduling problem will facilitate the comparison of the integer feasible solution to judge the success of the optimization process. In other words, if one can generate an integer

feasible solution to the problem, the degree of success of the integer solution can be measured if the optimal solution to the linear relaxation of the same problem is known. The difference between the integer feasible solution and the optimal linear solution of the same problem is often called an “optimality gap”. The goal of the optimization process is finding an integer feasible solution that will minimize the optimality gap. The exhaustive search of the integer solutions attained by heuristic methods cannot provide the optimality gap since the methods are not fundamentally sophisticated with the mathematical theorems that will lead to a proven optimal solution. On the other hand, the decomposition algorithms are developed based on intricate mathematical theorems that can guarantee the optimality of the linear relaxation solutions for the mine production scheduling problems. Henceforth, the integer solution algorithm developed in this thesis relies heavily on the mathematical theories behind the Bienstock Zuckerberg decomposition algorithm.

4.1 The Bienstock-Zuckerberg Decomposition Algorithm

The Bienstock-Zuckerberg algorithm (BZ) is the most powerful decomposition algorithm that can solve the LP relaxation of the large-scale mine production scheduling problems to a proven optimality. The LP optimality is accomplished by iteratively solving the master and subproblem until the convergence criteria is satisfied. It has been reported by Munoz *et al.* (2017) that the BZ algorithm outperforms the counterpart Dantzig-Wolfe decomposition algorithm. The algorithm derives its strength mainly by exploiting three important mathematical structures. First, the subproblem formulation defines a totally unimodular system that allows the subproblem to be formulated as a max flow problem as first shown by Johnson (1968). The subproblem is a multi-time period sequencing problem which can be represented on a network structure as first shown by Dagdelen (1985). Hochbaum (2008) proposed the Pseudoflow algorithm which is the fastest known algorithm for solving the max flow problems. Henceforth, realizing the underlying network structure of the subproblem and implementing the Pseudoflow algorithm to solve the subproblem as a max flow problem will speed up the convergence process. Secondly, the orthogonalization process that takes place between the subproblem solution and the former partitions increases the dimension of the solution space at each iteration which will allow the LP optimal solution to be captured very fast. The third strength is the contraction operation that leads to the conservation of the original problem structure at the master problem while working with the significantly reduced number of rows and columns.

The generic representation of the original problem adopted in this research is initially shown by Bienstock Zuckerberg (2009,2010 and 2015) and extended by Munoz (2017) as presented below.

Original Problem:

$$Max Z = c^T z \quad (4.1)$$

Subject To:

$$Az \leq b \quad (4.2)$$

$$Hz \leq h \quad (4.3)$$

$$z \in \{0,1\}^n \quad (4.4)$$

(4.1) represents the objective function value of the original problem where c^T is the cost vector. (4.2) encompasses the sequencing and reserve (ensures that the block is mined only once) constraints. (4.3) represents the resource capacity, blending and risk constraints in other words the side constraints. (4.4) ensures that the mining decision variables are in binary form.

The original problem is a well-known NP-hard problem form due to the presence of side constraints and the binary variables. Nevertheless, the LP relaxation of the problem can be solved by decomposing the problem into a master problem and a subproblem. Below is the representation of the subproblem.

Subproblem:

$$Max Z = c^T v - \pi(Hv - h) \quad (4.5)$$

Subject To:

$$Av \leq b \quad (4.6)$$

$$v \in \{0,1\}^n \quad (4.7)$$

The subproblem objective function (4.5) is written by penalizing the side constraints (4.3) of the original problem with the dual values “ π^T ” obtained by solving the master problem. The subproblem is often called a Lagrange relaxation problem. The subproblem is similar to the ultimate pit problem except the blocks are sequenced in multi time period. Since the subproblem or Lagrange relaxation problem has a totally unimodular structure, Johnson (1968) showed that

the single period version of the subproblem can be solved as a maximum flow network problem which is faster than solving the problem as a LP. The idea is extended by Dagdelen (1985) by exposing the mathematical relationship between the dual of the multi time period Lagrange relaxation problem which is identical to this subproblem and the underlying network structure. Currently Hochbaum`s Pseudoflow algorithm is the fastest known algorithm that can solve the ultimate pit limit problem or multi time period sequencing problem as a maximum flow and minimum s-t cut problem. The steps of the Pseudoflow algorithm will be outlined in the next section.

The solution space of the BZ algorithm takes place in Euclidean n-space which encompasses points with n coordinate values if the original problem contains n number of variables. Since all the variables are positive, only the positive orthant of Euclidean n-space will be considered. The constraints of the subproblem are the hyperplanes and the region bounded by their intersections forms a polytope in n-space. Each subproblem solution is essentially a column that represents the coordinate of an extreme point, located on the intersection of the hyperplanes ($Av \leq b$) on the polytope. The Figure 4.1 below represents a cross section taken from a polytope and the blue colored points represent the extreme points where the subproblem solution could potentially take place.

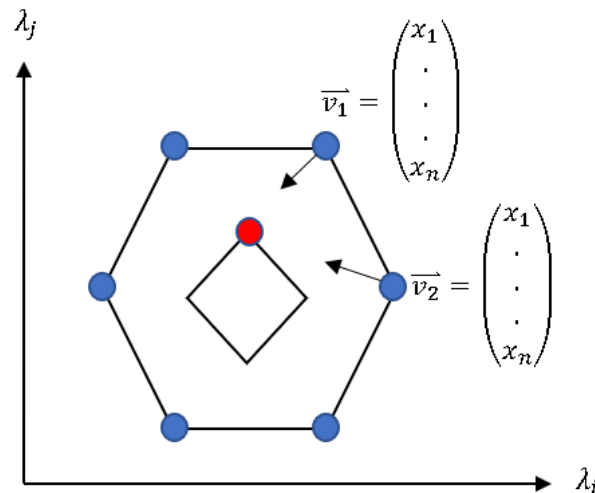


Figure 4.1: Cross section of a polytope illustrating the subproblem solution space (outer polytope) and the original problem solution space (inner polytope). Blue dots are the extreme points of the subproblem solutions, the red dot is the optimum extreme point of the original problem solution

Let's assume that \vec{v}_1 and \vec{v}_2 are the solution vectors obtained from the two consecutive iterations of the subproblem. The next step is the orthogonalization process of the \vec{v}_1 and \vec{v}_2 which will indeed result with a solution space greater than the one defined by the span of \vec{v}_1 and \vec{v}_2 . The idea behind the orthogonalization process is to expand the solution space defined by the partitions from the previous iterations by adding more axis to the solution space. In other words, expanding the set of axes that span the solution space will lead to a higher dimensional solution space where ultimately the optimal solution to the original problem will be captured.

The orthogonalization process is described clearly by Munoz *et al.* (2017), where the authors often call it a refining process. For iteration k , the orthogonalization process proceeds towards obtaining the new set of partitions V^k by performing set intersection and set difference operations on V^{k-1} and the current solution to the subproblem \vec{v}_k . The authors define set intersection operations as such: for two vectors $x, y \in \{0,1\}^n$, define $x \cap y \in \{0,1\}^n$ such that: $(x \cap y)_i = 1$ iff $x_i = 1$ and $y_i = 1$. Additionally, set difference operations are defined as such: for two vectors $x, y \in \{0,1\}^n$, define $x \setminus y \in \{0,1\}^n$ such that $(x \setminus y)_i = 1$ iff $x_i = 1$ and $y_i = 0$. Assuming V^{k-1} is comprised of $\{\vec{v}_1, \dots, \vec{v}_r\}$ then the new partition set V^k will consist of at most $2r + 1$ number of columns at the end of the following orthogonalization process:

$$\{\vec{v}_j \wedge \hat{v} : 1 \leq j \leq r\} \cup \{\vec{v}_j \setminus \hat{v} : 1 \leq j \leq r\} \cup \left\{ \hat{v} \setminus \left(\sum_{j=1}^{k-1} \vec{v}_j \right) \right\} \quad (4.8)$$

Let's assume the new partition set V^k encompasses orthogonal columns $\{\vec{v}_1, \dots, \vec{v}_s\}$. Then, there exists scalars or variables λ_i associated with each $\vec{v}_i \forall i = 1 \dots s$. The solution space of the new partition set is formed by taking the linear combination of the partitions as follows:

$$Span(V) = \{\lambda_1 \vec{v}_1 + \lambda_2 \vec{v}_2 \dots \dots \lambda_s \vec{v}_s\} = V \vec{\lambda} = \vec{x} \quad \text{where } \vec{\lambda} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \cdot \\ \cdot \\ \lambda_s \end{pmatrix} \quad (4.9)$$

After the orthogonalization process, the resulting partitions lead to a contraction operation which will significantly reduce the number of rows and columns in the master problem while preserving the original problem structure. Let's assume that the partition \vec{v}_1 represents the variables $\{x_1, x_2, \dots, x_{1000}\}$ in the original problem. Normally, each variable constitutes an axis in the original problem where the solution space of the x variables will span an immense space in

1000 dimensions. This gigantic space can be contracted by equating the variables in partition $\overline{v_1}$ to each other and replacing them with the variable λ_1 . Henceforth, instead of working with 1000 different axes, there will only be a λ_1 axis passing through the coordinates of $\{x_1, x_2, \dots, x_{1000}\}$. The Theorem 31 in Bienstock Zuckerberg (2009) shows that given the LP optimal solution, if q linearly independent side constraints are binding, then the LP optimal solution will contain no more than q distinct fractional values. An insight to this theorem could be illustrated as follows. Let's assume that the original problem consists of n number of variables and the optimal LP solution results with q number of distinct fractional values where $n \gg q$. Although the solution space of the original problem is governed by n number of axes, q number of distinct fractional values confirm that $n - q$ number of axes are redundant. Henceforth, if we can generate a λ axis for each one of the distinct fractional values, a set of $\{\lambda_1, \lambda_2, \dots, \lambda_q\}$ would be enough to capture the LP optimal solution. Every λ variable is essentially replacing a set of x variables within the original problem. Since the LP optimal solution exists in much lower dimensions than the original problem solution space, the contraction operation replaces the original axes of the problem with λ axes which will lead to a master problem formulation with a considerably lower number of variables and constraints. Below is the representation of the master problem.

Master Problem:

$$Max Z = c^T V\lambda \tag{4.10}$$

Subject To:

$$AV\lambda \leq b \tag{4.11}$$

$$HV\lambda \leq h \tag{4.12}$$

$$\lambda \in [0,1] \tag{4.13}$$

$$u \geq 0 \tag{4.14}$$

The master problem is a reformulation of the original problem constraints from (4.1) to (4.4) by replacing the x variables with the associated λ variables which qualifies as a restricted original problem. Furthermore, the set of constraints in the form of (4.12) are often called side constraints which destroys the underlying network structure which will prevent the implementation

of the max flow solving algorithms. Nevertheless, due to the contraction operation, the number of λ variables and the constraints in the master problem are still small enough to generate fast optimal solutions to the master problem. The polytope corresponding to the feasible region of the original problem takes place inside the subproblem solution space as shown in Figure 4.1. The optimal solution generated for the master problem is a feasible solution to the original problem solution space if not the optimal solution. Once the master problem solution is achieved; the dual vectors corresponding to the side constraints (4.12) are also generated. The significance of the dual vectors in the decomposition mechanism may be realized better if the relevance of the duality in the LP concept is emphasized. Furthermore, the working mechanism of the dual vectors form a strong connection between the simplex algorithm and the BZ decomposition algorithm which will be also illustrated.

Given a LP problem, based on the strong duality theorem; if the feasible solutions of the primal and the dual optimal solutions are equal, then the solution is proven to be the optimal LP solution. Each constraint of the primal problem has an associated dual variable and the right-hand side of the constraints are perceived as available resources. Obviously, the non-zero dual values will only exist if the available resources are fully consumed, in other words if the constraints are binding. The value of the associated dual variable in other words the shadow price will provide a useful measure on the impact of a unit change in the available resource on the objective function value. If the primal problem is a maximization problem, the positive dual variable can be interpreted as such that expanding the available resource will have a positive contribution to the objective function value and the opposite is true if the dual variable has a negative value. Let's also explore the significance of the duals in the simplex algorithm. In each iteration of the simplex algorithm, the non-basic variable which has the highest contribution to the objective function value is identified and included to the basic feasible solution. The variable selection process has an underlying economic interpretation. For each non-basic variable, its net worth is determined by the following equation $\bar{c}_i = c_i - \pi a_k$ where c_i the original coefficient of the variable in the objective function can be interpreted as revenue, dual variable or the reduce cost π represents the cost of consuming a unit of a resource and a_k is the amount of the resources consumed. Thus, \bar{c}_i can be interpreted as the modified value of the blocks.

The relationship between the master and subproblem of the BZ algorithm is apparent in the light of the duality concept. The dual vector π corresponding to the side constraints (4.12) will be used similarly to characterize the net worth of the mining decision variables. To be more precise, the block values will be modified with the corresponding dual values which may either make the blocks favorable or not. In the simplex algorithm we include the decision variable with the highest net worth, \bar{c}_i , into the basis. The pivoting process of the simplex algorithm is analogous to the subproblem solution. The modified block values are the net worth of the blocks based on the duals, and the solution to the subproblem is essentially the best mining plan generated with the modified block values. Henceforth, the λ variable corresponding to the plan enters the basis of the system. If the master problem is constrained as a single destination problem, then the dual vector π will modify the discounted value of the blocks over the time periods which may make the blocks more profitable for the later periods for the subproblem. If the blocks can be sent to multiple destinations as in the original problem that is being solved in this research, then the value of the block at each destination will be modified with the dual values per time period and the destination with the best value will be selected for each time period in the subproblem. This was proven by Johnson (1968). In a sense, the dual values will work similar to the cutoff grades to determine the best destination for the blocks. The relationship between the destination variables across the time periods for a single block was shown in the previous chapter. As it can be seen in Figure 4.2 below, the subproblem variable z_b^t is essentially represented by a single node, when the destination nodes of the $z_{b,d}^t$ variable is collapsed by picking the best destination from the modified block values.

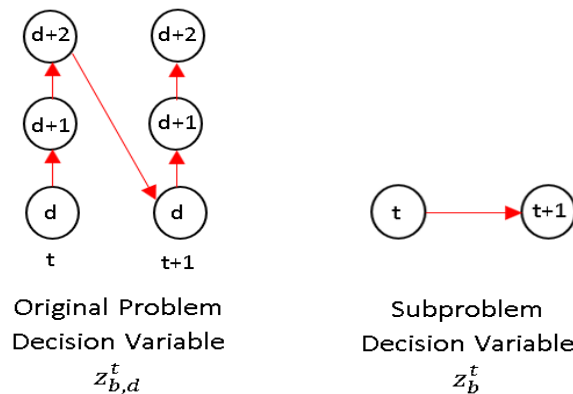


Figure 4.2: Node structure of the original problem variables and the subproblem variables

So far, the working mechanism of the BZ decomposition algorithm is explained in detail to set the groundwork for the new integer solution algorithm. The strengths of the BZ algorithm are mainly driven by the underlying network structure of the subproblem, the orthogonalization process and the contraction operation as illustrated in the example that follows. Also, the analogous relationship between the master and subproblem and the pivoting operation in the simplex algorithm is demonstrated in the light of the duality concept.

4.1.1 Small 2D BZ Example

In this section, the implementation of the BZ algorithm on a multi destination mine production scheduling problem will be demonstrated on a small 2D example. The deposit characteristics together with the decision variables and the economic block model is given in Figure 4.3. The production scheduling requirements are given in Figure 4.4.

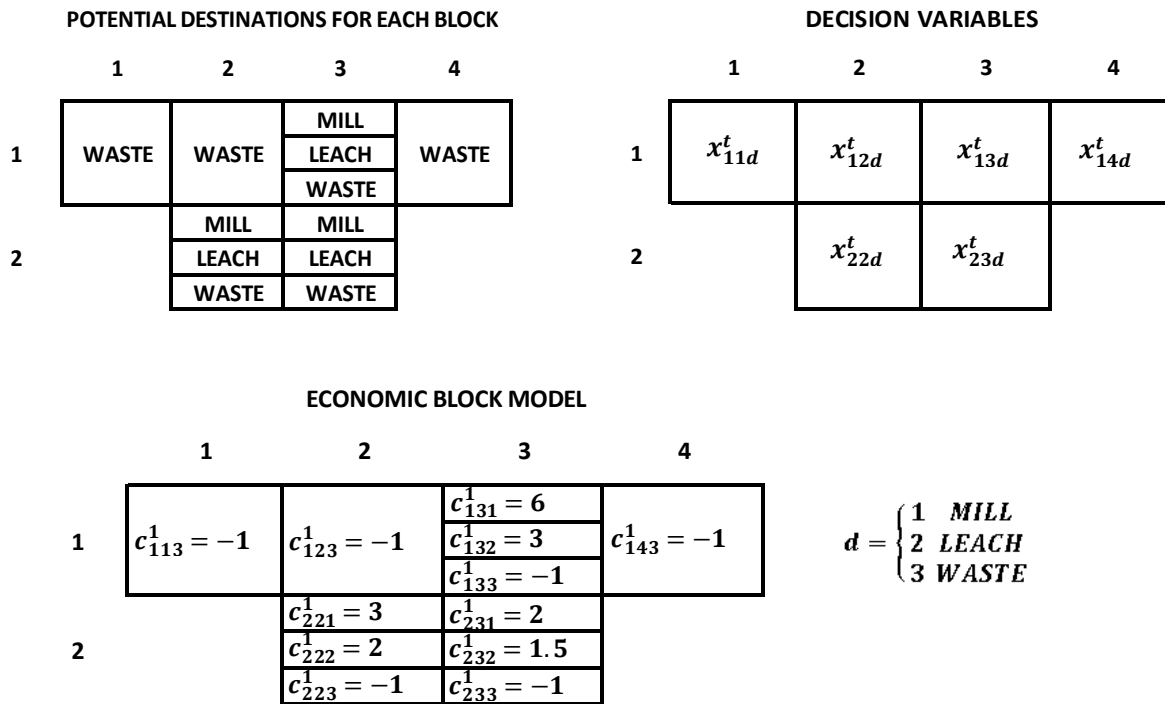


Figure 4.3: Economic block model for the potential destinations of the blocks

$$x_{bd}^t = 1 \text{ if block } b \text{ is mined in time period } t \text{ and sent to destination } d; 0 \text{ otherwise}$$

PERIODS	Total Mining Capacity	Mill Capacity	Leach Capacity
	Blocks	Blocks	Blocks
Period 1	3	1	0
Period 2	3	1	1

Mine Life	2 years
Slope Angle	45 ⁰

Figure 4.4: Production scheduling requirements

In this example, the mine plan consists of three constraints namely total mining capacity, mill capacity and leach capacity. The life of mine is 2 years and the pit walls must satisfy 45⁰ angle requirement. The ore blocks are assumed to be cubes and can be treated at one of the three potential destinations which are mill, leach and waste dump. The optimization process will determine the best destination for the block by realizing the interactions within the mining system. The original model formulation with the variables as derived in the previous chapter is given below.

Original Problem Formulation:

z_{bd}^t = the fraction of block b , that is extracted by time period t and sent by destination d

Objective Function:

$$\begin{aligned}
Max Z = & z_{113}^1 c_{113}^1 + z_{123}^1 c_{123}^1 + z_{143}^1 c_{143}^1 + z_{131}^1 (c_{131}^1 - c_{132}^1) \\
& + z_{131}^1 (c_{131}^1 - c_{132}^1) + z_{132}^1 (c_{132}^1 - c_{133}^1) + z_{133}^1 (c_{133}^1 - c_{131}^2) \\
& + z_{221}^1 (c_{221}^1 - c_{222}^1) + z_{222}^1 (c_{222}^1 - c_{223}^1) + z_{223}^1 (c_{223}^1 - c_{221}^2) \\
& + z_{231}^1 (c_{231}^1 - c_{232}^1) + z_{232}^1 (c_{232}^1 - c_{233}^1) + z_{233}^1 (c_{233}^1 - c_{231}^2) \\
& + z_{131}^2 (c_{131}^2 - c_{132}^2) + z_{132}^2 (c_{132}^2 - c_{133}^2) + z_{133}^2 c_{133}^2 \\
& + z_{221}^2 (c_{221}^2 - c_{222}^2) + z_{222}^2 (c_{222}^2 - c_{223}^2) + z_{223}^2 c_{223}^2 \\
& + z_{231}^2 (c_{231}^2 - c_{232}^2) + z_{232}^2 (c_{232}^2 - c_{233}^2) + z_{233}^2 c_{233}^2 \\
& + z_{113}^2 c_{113}^2 + z_{123}^2 c_{123}^2 + z_{143}^2 c_{143}^2
\end{aligned} \tag{4.15}$$

Subject To:

Ensures Sequencing Across Time Periods

$$z_{113}^1 \leq z_{113}^2 \tag{4.16}$$

$$z_{123}^1 \leq z_{123}^2 \quad (4.17)$$

$$z_{133}^1 \leq z_{131}^2 \quad (4.18)$$

$$z_{143}^1 \leq z_{143}^2 \quad (4.19)$$

$$z_{223}^1 \leq z_{221}^2 \quad (4.20)$$

$$z_{233}^1 \leq z_{1231}^2 \quad (4.21)$$

Ensures Sequencing Between Destinations Within Time Periods:

$$z_{131}^1 \leq z_{132}^1 \quad (4.22)$$

$$z_{132}^1 \leq z_{133}^1 \quad (4.23)$$

$$z_{221}^1 \leq z_{222}^1 \quad (4.24)$$

$$z_{222}^1 \leq z_{223}^1 \quad (4.25)$$

$$z_{231}^1 \leq z_{232}^1 \quad (4.26)$$

$$z_{232}^1 \leq z_{233}^1 \quad (4.27)$$

$$z_{131}^2 \leq z_{132}^2 \quad (4.28)$$

$$z_{132}^2 \leq z_{133}^2 \quad (4.29)$$

$$z_{221}^2 \leq z_{222}^2 \quad (4.30)$$

$$z_{222}^2 \leq z_{223}^2 \quad (4.31)$$

$$z_{231}^2 \leq z_{232}^2 \quad (4.32)$$

$$z_{232}^2 \leq z_{233}^2 \quad (4.33)$$

Block Sequencing Constraints:

$$z_{223}^1 \leq z_{113}^1 \quad (4.34)$$

$$z_{223}^1 \leq z_{123}^1 \quad (4.35)$$

$$z_{223}^1 \leq z_{133}^1 \quad (4.36)$$

$$z_{233}^1 \leq z_{123}^1 \quad (4.37)$$

$$z_{233}^1 \leq z_{133}^1 \quad (4.38)$$

$$z_{233}^1 \leq z_{143}^1 \quad (4.39)$$

$$z_{223}^2 \leq z_{113}^2 \quad (4.40)$$

$$z_{223}^2 \leq z_{123}^2 \quad (4.41)$$

$$z_{223}^2 \leq z_{133}^2 \quad (4.42)$$

$$z_{233}^2 \leq z_{123}^2 \quad (4.43)$$

$$z_{233}^2 \leq z_{133}^2 \quad (4.44)$$

$$z_{233}^2 \leq z_{143}^2 \quad (4.45)$$

Capacity Constraints:

Mill Capacity Constraints:

$$z_{131}^1 + z_{221}^1 + z_{231}^1 \leq 1 \quad (4.46)$$

$$(z_{131}^2 - z_{133}^1) + (z_{221}^2 - z_{223}^1) + (z_{231}^2 - z_{233}^1) \leq 1 \quad (4.47)$$

Leach Capacity Constraints:

$$(z_{132}^2 - z_{131}^1) + (z_{222}^2 - z_{221}^1) + (z_{232}^2 - z_{231}^1) \leq 1 \quad (4.48)$$

Total Mining Capacity Constraints:

$$z_{113}^1 + z_{123}^1 + z_{131}^1 + (z_{132}^1 - z_{131}^1) + (z_{133}^1 - z_{132}^1) + z_{143}^1 + z_{221}^1 + (z_{222}^1 - z_{221}^1) \\ + (z_{223}^1 - z_{222}^1) + z_{231}^1 + (z_{232}^1 - z_{231}^1) + (z_{233}^1 - z_{232}^1) \leq 3 \quad (4.49)$$

$$z_{113}^2 + z_{123}^2 + (z_{131}^2 - z_{133}^1) + (z_{132}^2 - z_{131}^1) + (z_{133}^2 - z_{132}^1) + z_{143}^2 + (z_{221}^2 - z_{223}^1) \\ + (z_{222}^2 - z_{221}^1) + (z_{223}^2 - z_{222}^1) + (z_{231}^2 - z_{233}^1) + (z_{232}^2 - z_{231}^1) + (z_{233}^2 - z_{232}^1) \leq 3 \quad (4.50)$$

Variable Restrictions:

$$0 \leq z_{ijd}^t \leq 1 \quad \forall i \in \{1,2\}, \forall j \in \{1,2,3,4\}, \forall t \in \{1,2\}, \forall d \in \{1,2,3\} \quad (4.51)$$

STEP 1:

The first step of the BZ algorithm will start by determining the initial set of partitions V^0 to formulate the master problem. Each partition is essentially a mine plan and there are no feasibility requirements for the selected mine plan. In other words, the starting mine plan does not have to be feasible to the original problem solution space. Hence, a partition will be created from the set of blocks within the ultimate pit. If the life of mine is t periods, then t number of orthogonal partitions will be generated where each partition has all the blocks within the ultimate pit. Indeed, a number of mine plans are generated where each plan mines the ultimate pit. This plan is most likely infeasible to the original problem space. The orthogonality between the partitions are also preserved by allowing each partition to represent the blocks within the ultimate pit only for a specific time period. Furthermore, generating an ultimate pit is similar to solving the subproblem with the original block values by assuming the dual values are zero.

In Figure 4.5, the economic block model is shown for year 1 and discounted for year 2. The blocks have value depending on the treatment type at the destination. The ultimate pit calculation requires to predetermine the destination of a block. Hence, the destination where the

block gains the highest value will be selected. The selected block values are shown with light blue color in Figure 4.5. The ultimate pit limit will be determined by using Pseudoflow algorithm. The Figure 4.6 shows the resulting orthogonal columns \vec{s}_1 and \vec{s}_2 where \vec{s}_1 represents the blocks mined in the ultimate pit in period 1, and \vec{s}_2 represents the blocks mined in the ultimate pit in period 2. Since each period segment of the solution column can be perceived as a pit, the blocks mined in the period 1 segment of the column \vec{s}_1 is colored with yellow and the period 2 segment of the column \vec{s}_2 is colored with green. This is essentially the representation of the orthogonal columns in terms of mining pits.

	1	2	3	4
1	$c_{113}^1 = -1$	$c_{123}^1 = -1$	$c_{131}^1 = 6$	$c_{143}^1 = -1$
$c_{132}^1 = 3$				
$c_{133}^1 = -1$				
2		$c_{221}^1 = 3$	$c_{231}^1 = 2$	
		$c_{222}^1 = 2$	$c_{232}^1 = 1.5$	
		$c_{223}^1 = -1$	$c_{233}^1 = -1$	

	1	2	3	4
1	$c_{113}^2 = -0.9$	$c_{123}^2 = -0.9$	$c_{131}^2 = 5.3$	$c_{143}^2 = -0.9$
$c_{132}^2 = 2.7$				
$c_{133}^2 = -0.9$				
2		$c_{221}^2 = 2.7$	$c_{231}^2 = 1.8$	
		$c_{222}^2 = 1.8$	$c_{232}^2 = 1.3$	
		$c_{223}^2 = -0.9$	$c_{233}^2 = -0.9$	

Figure 4.5: Fixing the destinations based on the highest block value. The blue colors represent the most valuable destinations for a block in time period 1 and period 2

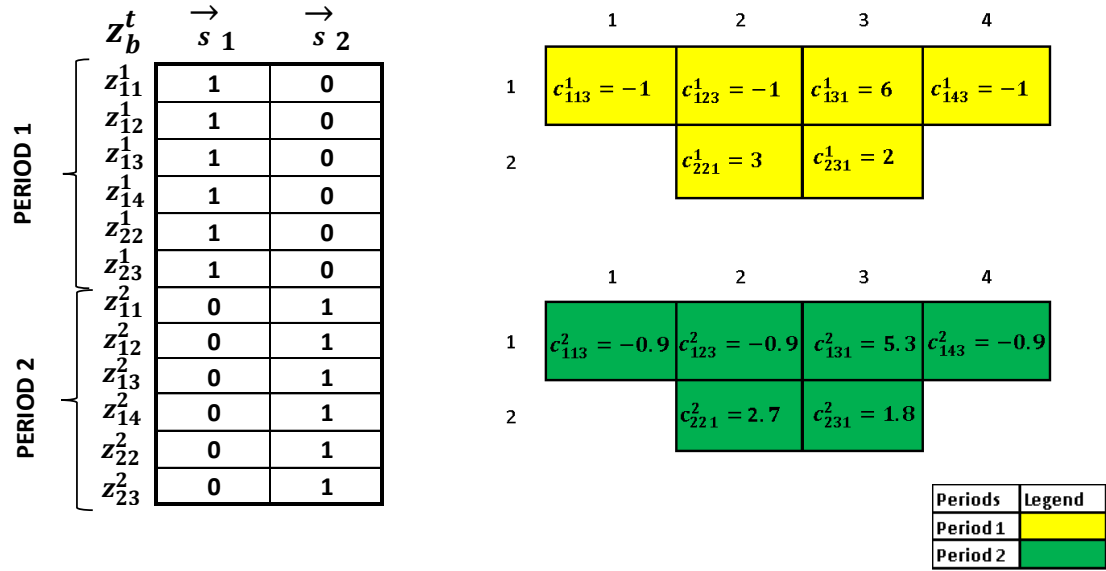


Figure 4.6: Subproblem solution column and the mining plans generated per period

The original variable z_{bd}^t has a destination index, however the subproblem solution does not have the destination index since the destinations are predetermined. The blocks that exist in columns \bar{s}_1 and \bar{s}_2 are assigned to the destinations with the highest value. In order to model the master problem, the solution columns \bar{s}_1 and \bar{s}_2 need to be converted to \bar{v}_1 and \bar{v}_2 as shown in Figure 4.7. The conversion process follows the node structure of the original problem variable z_{bd}^t which is illustrated in the previous chapter. Since the column \bar{v}_1 has the same structure of the original problem variables, the λ variable associated for each column \bar{v} will easily replace the original problem variables z_{bd}^t . The solution column of the subproblem is only split into period segments. For the conversion process, each period segment needs to be split further into destination segments as shown in Figure 4.7. For example, the block z_{13}^1 shows up in column \bar{s}_1 and we know that the block was designated to the mill. Hence, the location of z_{131}^1 in column \bar{v}_1 must be 1. Due to the “by” relationship between the destinations, the location of z_{132}^1 and of z_{133}^1 will also be 1. Eventually, every block variable z_b^t in \bar{s}_1 will be converted to z_{bd}^t and placed in \bar{v}_1 . If the predetermined destination is d in time period t, then “1” will be assigned to the location of the same block for destination d to D and time period t to T.

		Z_b^t					
		\rightarrow	\rightarrow				
		s_1	s_2			\rightarrow	\rightarrow
						v_1	v_2
PERIOD 1	MILL	z_{111}^1	1	0	z_{111}^1	0	0
		z_{121}^1	1	0	z_{121}^1	0	0
		z_{131}^1	1	0	z_{131}^1	1	0
		z_{141}^1	1	0	z_{141}^1	0	0
		z_{221}^1	1	0	z_{221}^1	1	0
		z_{231}^1	1	0	z_{231}^1	1	0
	LEACH	z_{112}^1	0	0	z_{112}^1	0	0
		z_{122}^1	0	0	z_{122}^1	0	0
		z_{132}^1	1	0	z_{132}^1	1	0
		z_{142}^1	0	0	z_{142}^1	0	0
		z_{222}^1	1	0	z_{222}^1	1	0
		z_{232}^1	1	0	z_{232}^1	1	0
	WASTE	z_{113}^1	1	0	z_{113}^1	1	0
		z_{123}^1	1	0	z_{123}^1	1	0
z_{133}^1		1	0	z_{133}^1	1	0	
z_{143}^1		1	0	z_{143}^1	1	0	
z_{223}^1		1	0	z_{223}^1	1	0	
z_{233}^1		1	0	z_{233}^1	1	0	
PERIOD 2	MILL	z_{111}^2	0	1	z_{111}^2	0	0
		z_{121}^2	0	1	z_{121}^2	0	0
		z_{131}^2	0	1	z_{131}^2	0	1
		z_{141}^2	0	1	z_{141}^2	0	0
		z_{221}^2	0	1	z_{221}^2	0	1
		z_{231}^2	0	1	z_{231}^2	0	1
	LEACH	z_{112}^2	0	0	z_{112}^2	0	0
		z_{122}^2	0	0	z_{122}^2	0	0
		z_{132}^2	0	1	z_{132}^2	0	1
		z_{142}^2	0	0	z_{142}^2	0	0
		z_{222}^2	0	1	z_{222}^2	0	1
		z_{232}^2	0	1	z_{232}^2	0	1
	WASTE	z_{113}^2	0	1	z_{113}^2	0	1
		z_{123}^2	0	1	z_{123}^2	0	1
z_{133}^2		0	1	z_{133}^2	0	1	
z_{143}^2		0	1	z_{143}^2	0	1	
z_{223}^2		0	1	z_{223}^2	0	1	
z_{233}^2		0	1	z_{233}^2	0	1	

Figure 4.7: Subproblem solution column on the left is split into destination segments on the right in order to represent the original problem variable z_{bd}^t node structure in the master problem

STEP 2:

Once the orthogonal partitions \vec{v}_1 and \vec{v}_2 are obtained, the master problem is ready to be formulated. The formulation will proceed with the contraction operation where λ_1 and λ_2 will replace the variables that exist in the partitions \vec{v}_1 and \vec{v}_2 . The contraction operation preserves the original problem structure. Intuitively, the master problem formulation is the original problem formulation with the extra equality constraints between the variables that exist in the same partition. Below is the mathematical model for the master problem generated by substituting z_{bd}^t variables of the original problem with the corresponding λ variables.

$$\lambda_1 \rightarrow \vec{v}_1 = \{z_{113}^1, z_{123}^1, z_{133}^1, z_{143}^1, z_{223}^1, z_{233}^1, z_{131}^1, z_{221}^1, z_{231}^1, z_{132}^1, z_{222}^1, z_{232}^1\}$$

$$\lambda_2 \rightarrow \vec{v}_2 = \{z_{113}^2, z_{123}^2, z_{133}^2, z_{143}^2, z_{223}^2, z_{233}^2, z_{131}^2, z_{221}^2, z_{231}^2, z_{132}^2, z_{222}^2, z_{232}^2\}$$

Master Problem Formulation:

Objective Function:

$$\begin{aligned} \text{Max } Z = & \lambda_1 c_{113}^1 + \lambda_1 c_{123}^1 + \lambda_1 c_{143}^1 + \lambda_1 (c_{131}^1 - c_{132}^1) + \lambda_1 (c_{131}^1 - c_{132}^1) \\ & + \lambda_1 (c_{132}^1 - c_{133}^1) + \lambda_1 (c_{133}^1 - c_{231}^2) + \lambda_1 (c_{221}^1 - c_{222}^1) + \lambda_1 (c_{222}^1 - c_{223}^1) \\ & + \lambda_1 (c_{223}^1 - c_{221}^1) + \lambda_1 (c_{231}^1 - c_{232}^1) + \lambda_1 (c_{232}^1 - c_{233}^1) + \lambda_1 (c_{233}^1 - c_{231}^1) \\ & + \lambda_2 (c_{131}^2 - c_{132}^2) + \lambda_2 (c_{132}^2 - c_{133}^2) + \lambda_2 c_{133}^2 + \lambda_2 (c_{221}^2 - c_{222}^2) \\ & + \lambda_2 (c_{222}^2 - c_{223}^2) + \lambda_2 c_{223}^2 + \lambda_2 (c_{231}^2 - c_{232}^2) + \lambda_2 (c_{232}^2 - c_{233}^2) + \lambda_2 c_{233}^2 \\ & + \lambda_2 c_{113}^2 + \lambda_2 c_{123}^2 + \lambda_2 c_{143}^2 \end{aligned} \tag{4.52}$$

Subject To:

Ensures Sequencing Across Time Periods:

$$\lambda_1 \leq \lambda_2 \quad \text{replaces (eq 4.16,4.17, 4.18, 4.19, 4.20, 4.21)} \tag{4.53}$$

Ensures Sequencing Between Destinations Within Time Periods:

$$\lambda_1 \leq \lambda_1 \quad \text{replaces (eq 4.22,4.23, 4.24, 4.25, 4.26, 4.27)} \tag{4.54}$$

$$\lambda_2 \leq \lambda_2 \quad \text{replaces (eq 4.28,4.29, 4.30, 4.31, 4.32, 4.33)} \tag{4.55}$$

Block Sequencing Constraints:

$$\lambda_1 \leq \lambda_1 \quad \text{replaces (eq 4.34,4.35, 4.36, 4.37, 4.38, 4.39)} \tag{4.56}$$

$$\lambda_2 \leq \lambda_2 \quad \text{replaces (eq 4.40,4.41, 4.42, 4.43, 4.44, 4.45)} \tag{4.57}$$

Capacity Constraints:

Mill Capacity Constraints:

$$\lambda_1 + \lambda_1 + \lambda_1 \leq 1 \quad \text{replaces (eq 4.47)} \tag{4.58}$$

$$(\lambda_2 - \lambda_1) + (\lambda_2 - \lambda_1) + (\lambda_2 - \lambda_1) \leq 1 \quad \text{replaces (eq 4.47)} \quad (4.59)$$

Leach Capacity Constraints:

$$(\lambda_2 - \lambda_2) + (\lambda_2 - \lambda_2) + (\lambda_2 - \lambda_2) \leq 1 \quad \text{replaces (eq 4.48)} \quad (4.60)$$

Total Mining Capacity Constraints:

$$\lambda_1 + \lambda_1 + \lambda_1 + (\lambda_1 - \lambda_1) + (\lambda_1 - \lambda_1) + \lambda_1 + \lambda_1 + (\lambda_1 - \lambda_1) + (\lambda_1 - \lambda_1) + \lambda_1 + (\lambda_1 - \lambda_1) + (\lambda_1 - \lambda_1) \leq 3 \quad \text{replaces (eq 4.49)} \quad (4.61)$$

$$\lambda_2 + \lambda_2 + (\lambda_2 - \lambda_1) + (\lambda_2 - \lambda_2) + (\lambda_2 - \lambda_2) + \lambda_2 + (\lambda_2 - \lambda_1) + (\lambda_2 - \lambda_2) + (\lambda_2 - \lambda_1) + (\lambda_2 - \lambda_1) + (\lambda_2 - \lambda_2) + (\lambda_2 - \lambda_2) \leq 3 \quad \text{replaces (eq 4.50)} \quad (4.62)$$

Variable Restrictions:

$$0 \leq \lambda_1 \leq 1 \quad (4.63)$$

$$0 \leq \lambda_2 \leq 1 \quad (4.64)$$

The mathematical model generated by direct variable substitution consists of many redundant constraints. Below is the simplified mathematical model for the master problem.

Objective Function:

$$\text{Max } Z = 7.1\lambda_1 - 4.2\lambda_2 \quad (4.65)$$

Subject To:

Sequencing Constraint:

$$\lambda_1 \leq \lambda_2 \quad (4.66)$$

Capacity Constraints:

Mill Capacity Constraints:

$$3\lambda_1 \leq 1 \quad (4.67)$$

$$3\lambda_2 - 3\lambda_1 \leq 1 \quad (4.68)$$

Total Mining Capacity Constraints:

$$6\lambda_1 \leq 3 \quad (4.69)$$

$$7\lambda_2 - 4\lambda_1 \leq 3 \quad (4.70)$$

Variable Restrictions:

$$0 \leq \lambda_1 \leq 1 \quad (4.71)$$

$$0 \leq \lambda_2 \leq 1 \quad (4.72)$$

When the problem is solved with the simplex algorithm, we get $\lambda_1 = 1/3$, $\lambda_2 = 1/3$, which means that the mining decision variables in set \bar{v}_1 and \bar{v}_2 are also equal to $1/3$.

STEP 3:

Once the master problem is solved, the dual values μ_k^t are generated for the side constraints.

The dual values for this problem are assumed for demonstration purposes.

$$\text{Duals For Mill Capacity Constraints} \quad \rightarrow \mu_1^1 = 2, \quad \mu_1^2 = 2.5$$

$$\text{Duals For Leach Capacity Constraints} \quad \rightarrow \mu_2^1 = 0, \quad \mu_2^2 = 0$$

$$\text{Duals For Total Mining Capacity Constraints} \quad \rightarrow \mu_3^1 = 1, \quad \mu_3^2 = 0.7$$

The dual values will be used to modify the original block values. If the block is a waste block, only the dual from the total mining capacity constraint will be used. For the mill blocks, the duals for the mill capacity and the total mining capacity constraints will be used. Similarly, the leach block values will be penalized with the duals from the leach capacity and the total mining capacity constraints. The modified block values are shown in Figure 4.8. In order to solve the multi time period sequencing subproblem, the destinations of the blocks need to be predetermined. Henceforth, by looking at the modified value of the blocks at each destination, the destination with the highest value will be selected. Once the selection is completed, the subproblem is solved by implementing the Pseudoflow algorithm. In Figure 4.9, the column \vec{s}_3 represents the solution to the subproblem which is essentially the best current mine plan qualified to enter the basis. Since the solution is in “by” form, the blocks mine in period 1 also shows in the period 2 section of column \vec{s}_3 . As it can be seen in Figure 4.9, the first bench is mined in period 1 colored by yellow. The modified block values delay the production of the second bench to the second period which leads to a higher profit as shown by green color. The next step is converting the subproblem solution column \vec{s}_3 to \vec{v}_3 , in order to preserve the node structure of the original problem variables. The Figure 4.10 makes it easier to see how the resolution is increased by splitting each period segment of column \vec{s}_3 into destinations in column \vec{v}_3 . The orthogonal partitions from the previous iteration are \vec{v}_1 and \vec{v}_2 . If \vec{v}_3 is introduced as a new partition to the set of previous partitions, it will be clear \vec{v}_3 is not orthogonal to \vec{v}_1 and \vec{v}_2 as shown in Figure 4.11. Hence, the orthogonalization process must be initiated, in order to expand the solution space defined by the partitions from the previous iterations by adding more axis to the solution space. Figure 4.12 demonstrates the new partitions after the orthogonalization process.

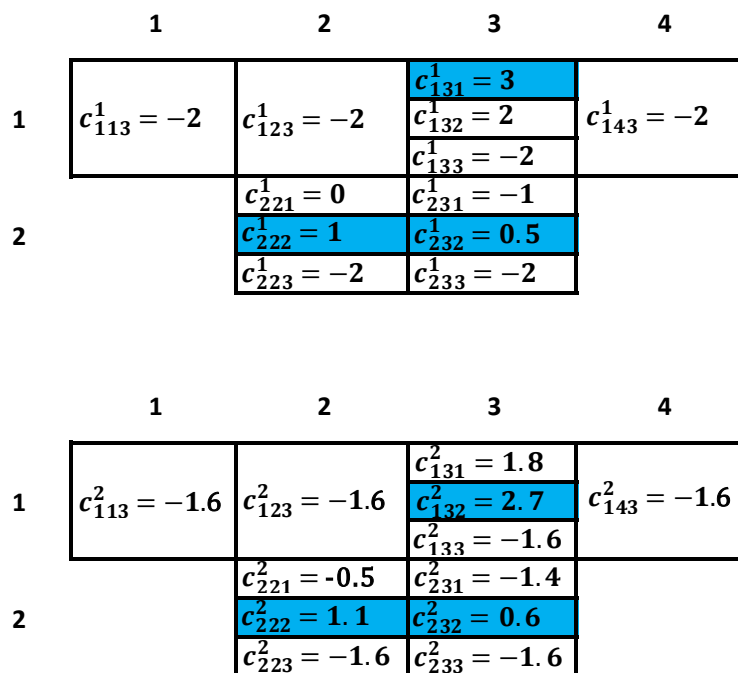


Figure 4.8: Modified block values by the duals for period 1 and period 2. The blue colors represent the most valuable destinations for a block in time period 1 and period 2

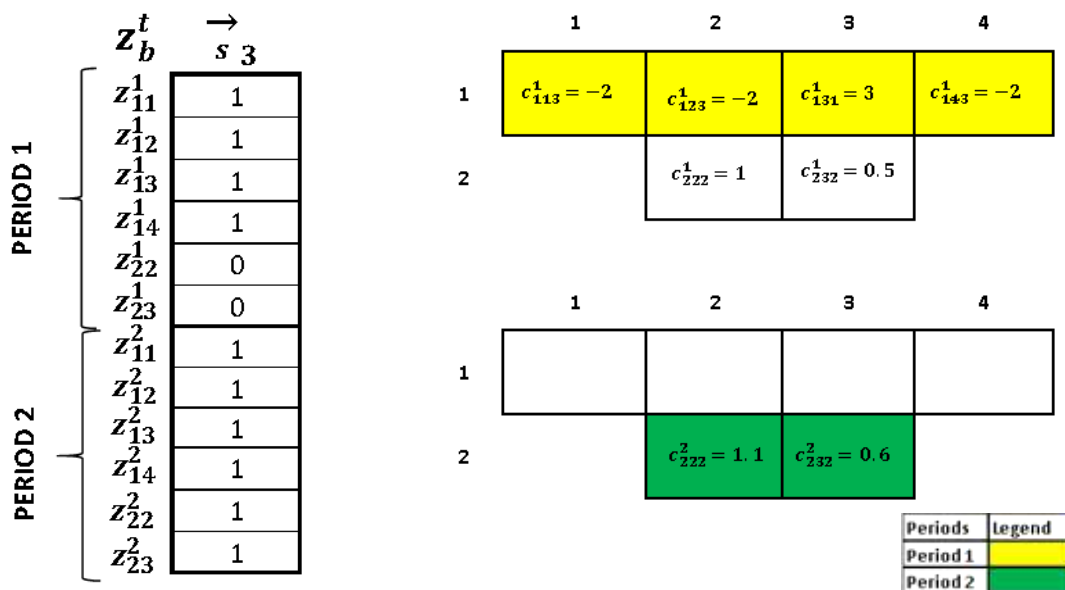


Figure 4.9: Subproblem solution column and the mining plans generated per period

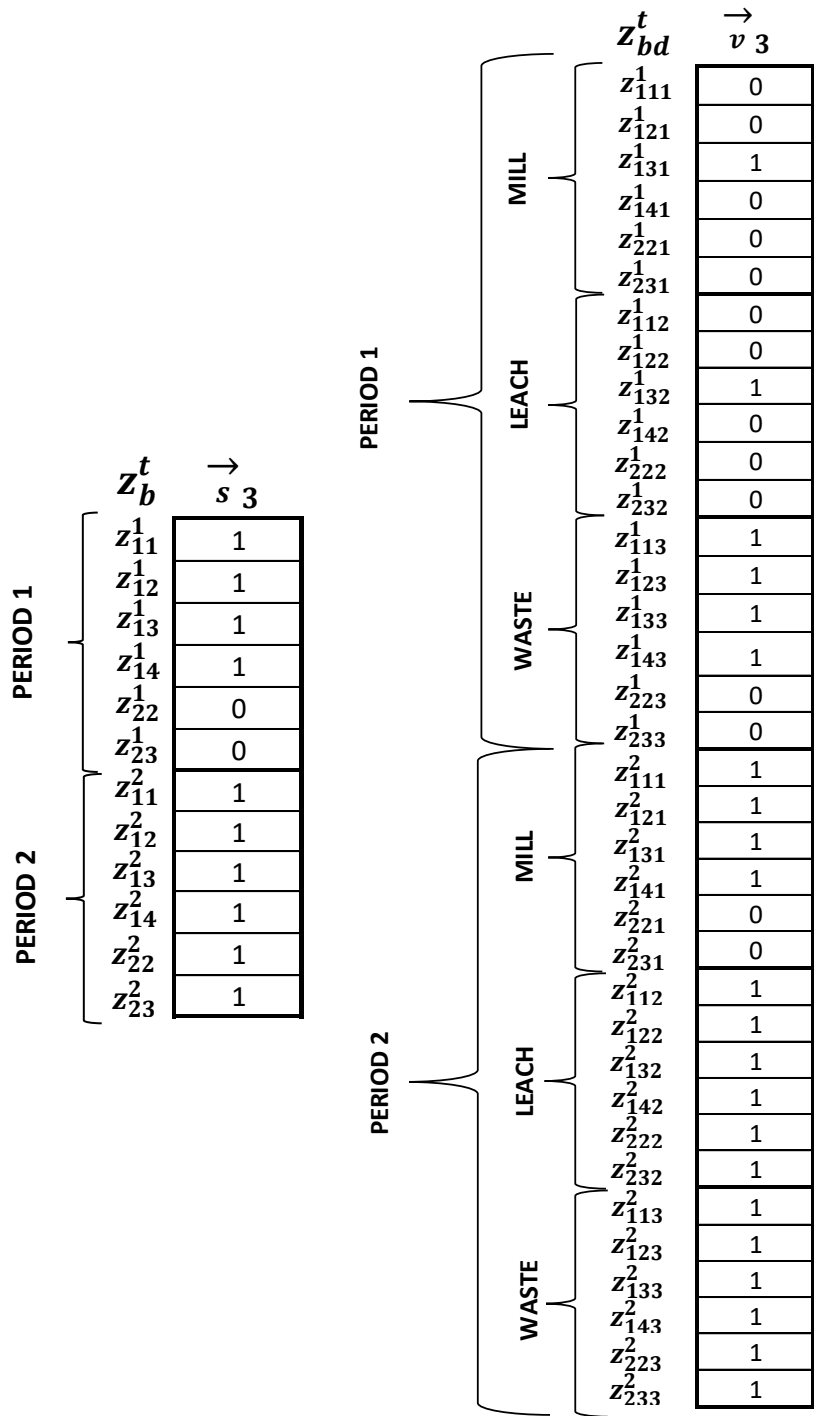


Figure 4.10: Subproblem solution column on the left is split into destination segments on the right in order to represent the original problem variable z_{bd}^t node structure in the master problem

		z_{bd}^t	\vec{v}_1	\vec{v}_2	\vec{v}_3
PERIOD 1	MILL	z_{111}^1	0	0	0
		z_{121}^1	0	0	0
		z_{131}^1	1	0	1
		z_{141}^1	0	0	0
		z_{221}^1	1	0	0
		z_{231}^1	1	0	0
	LEACH	z_{112}^1	0	0	0
		z_{122}^1	0	0	0
		z_{132}^1	1	0	1
		z_{142}^1	0	0	0
		z_{222}^1	1	0	0
		z_{232}^1	1	0	0
	WASTE	z_{113}^1	1	0	1
		z_{123}^1	1	0	1
		z_{133}^1	1	0	1
		z_{143}^1	1	0	1
		z_{223}^1	1	0	0
		z_{233}^1	1	0	0
PERIOD 2	MILL	z_{111}^2	0	0	1
		z_{121}^2	0	0	1
		z_{131}^2	0	1	1
		z_{141}^2	0	0	1
		z_{221}^2	0	1	0
		z_{231}^2	0	1	0
	LEACH	z_{112}^2	0	0	1
		z_{122}^2	0	0	1
		z_{132}^2	0	1	1
		z_{142}^2	0	0	1
		z_{222}^2	0	1	1
		z_{232}^2	0	1	1
	WASTE	z_{113}^2	0	1	1
		z_{123}^2	0	1	1
		z_{133}^2	0	1	1
		z_{143}^2	0	1	1
		z_{223}^2	0	1	1
		z_{233}^2	0	1	1

Figure 4.11: Current state of the partitions after appending the \vec{v}_3 solution column to the previous set of partitions. At this stage \vec{v}_3 is not orthogonal to \vec{v}_1 and \vec{v}_2 yet

		z_{bd}^t	$\vec{v}_1 \cap \vec{v}_3$	$\vec{v}_2 \cap \vec{v}_3$	\vec{v}_1/\vec{v}_3	\vec{v}_2/\vec{v}_3	$\vec{v}_3/(\vec{v}_1 + \vec{v}_2)$
PERIOD 1	MILL	z_{111}^1	0	0	0	0	0
		z_{121}^1	0	0	0	0	0
		z_{131}^1	1	0	0	0	0
		z_{141}^1	0	0	0	0	0
		z_{221}^1	0	0	1	0	0
	LEACH	z_{231}^1	0	0	1	0	0
		z_{112}^1	0	0	0	0	0
		z_{122}^1	0	0	0	0	0
		z_{132}^1	1	0	0	0	0
		z_{142}^1	0	0	0	0	0
	WASTE	z_{222}^1	0	0	1	0	0
		z_{232}^1	0	0	1	0	0
		z_{113}^1	1	0	0	0	0
		z_{123}^1	1	0	0	0	0
		z_{133}^1	1	0	0	0	0
PERIOD 2	MILL	z_{143}^1	1	0	0	0	0
		z_{223}^1	0	0	1	0	0
		z_{233}^1	0	0	1	0	0
		z_{111}^2	0	0	0	0	1
		z_{121}^2	0	0	0	0	1
	LEACH	z_{131}^2	0	1	0	0	0
		z_{141}^2	0	0	0	0	1
		z_{221}^2	0	0	0	1	0
		z_{231}^2	0	0	0	1	0
		z_{112}^2	0	0	0	0	1
	WASTE	z_{122}^2	0	0	0	0	1
		z_{132}^2	0	1	0	0	0
		z_{142}^2	0	0	0	0	1
		z_{222}^2	0	1	0	0	0
		z_{232}^2	0	1	0	0	0
WASTE	z_{113}^2	0	1	0	0	0	
	z_{123}^2	0	1	0	0	0	
	z_{133}^2	0	1	0	0	0	
	z_{143}^2	0	1	0	0	0	
	z_{223}^2	0	1	0	0	0	
z_{233}^2	0	1	0	0	0		

Figure 4.12: New set of partitions after the orthogonalization process

In the previous iteration, the linear combination of λ_1 and λ_2 variables formed a solution space in two dimensions, while the orthogonalization process in the current iteration resulted with a solution space in higher dimensions spanned by $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ variables. With the contraction operation, the original z_{bd}^t variables stored in the partitions \vec{v}_1 to \vec{v}_5 will be replaced with the variables λ_1 to λ_5 and the master problem will be formulated.

$$\lambda_1 \rightarrow \vec{v}_1 = \{z_{131}^1, z_{132}^1, z_{113}^1, z_{123}^1, z_{133}^1, z_{143}^1\}$$

$$\lambda_2 \rightarrow \vec{v}_2 = \{z_{131}^2, z_{132}^2, z_{222}^2, z_{232}^2, z_{113}^2, z_{123}^2, z_{133}^2, z_{143}^2, z_{223}^2, z_{233}^2\}$$

$$\lambda_3 \rightarrow \vec{v}_3 = \{z_{221}^1, z_{231}^1, z_{222}^1, z_{232}^1, z_{223}^1, z_{233}^1\}$$

$$\lambda_4 \rightarrow \vec{v}_4 = \{z_{221}^2, z_{231}^2\}$$

$$\lambda_5 \rightarrow \vec{v}_5 = \{z_{111}^2, z_{121}^2, z_{141}^2, z_{112}^2, z_{122}^2, z_{142}^2\}$$

Master Problem Formulation:

Objective Function:

$$\begin{aligned} Max Z = & \lambda_1 c_{113}^1 + \lambda_1 c_{123}^1 + \lambda_1 c_{143}^1 + \lambda_1 (c_{131}^1 - c_{132}^1) \\ & + \lambda_1 (c_{131}^1 - c_{132}^1) + \lambda_1 (c_{132}^1 - c_{133}^1) + \lambda_1 (c_{133}^1 - c_{131}^2) \\ & + \lambda_3 (c_{221}^1 - c_{222}^1) + \lambda_3 (c_{222}^1 - c_{223}^1) + \lambda_3 (c_{223}^1 - c_{221}^2) \\ & + \lambda_3 (c_{231}^1 - c_{232}^1) + \lambda_3 (c_{232}^1 - c_{233}^1) + \lambda_3 (c_{233}^1 - c_{231}^2) \\ & + \lambda_2 (c_{131}^2 - c_{132}^2) + \lambda_2 (c_{132}^2 - c_{133}^2) + \lambda_2 c_{133}^2 \\ & + \lambda_4 (c_{221}^2 - c_{222}^2) + \lambda_4 (c_{222}^2 - c_{223}^2) + \lambda_2 c_{223}^2 \\ & + \lambda_4 (c_{231}^2 - c_{232}^2) + \lambda_2 (c_{232}^2 - c_{233}^2) + \lambda_2 c_{233}^2 \\ & + \lambda_2 c_{113}^2 + \lambda_2 c_{123}^2 + \lambda_2 c_{143}^2 \end{aligned} \quad (4.73)$$

Subject To:

Ensures Sequencing Across Time Periods:

$$\lambda_1 \leq \lambda_2 \quad \text{replaces (eq 4.16, 4.17, 4.18, 4.19)} \quad (4.74)$$

$$\lambda_3 \leq \lambda_4 \quad \text{replaces (eq 4.20, 4.21)} \quad (4.75)$$

Ensures Sequencing Between Destinations Within Time Periods:

$$\lambda_1 \leq \lambda_2 \quad \text{replaces (eq 4.22)} \quad (4.76)$$

$$\lambda_1 \leq \lambda_1 \quad \text{replaces (eq 4.23)} \quad (4.77)$$

$$\lambda_3 \leq \lambda_3 \quad \text{replaces (eq 4.24, 4.25, 4.26, 4.27)} \quad (4.78)$$

$$\lambda_2 \leq \lambda_2 \quad \text{replaces (eq 4.28, 4.29, 4.31, 4.33)} \quad (4.79)$$

$$\lambda_4 \leq \lambda_2 \quad \text{replaces (eq 4.30, 4.32)} \quad (4.80)$$

Block Sequencing Constraints:

$$\lambda_3 \leq \lambda_1 \quad \text{replaces(eq 4.34, 4.35, 4.36, 4.37, 4.38, 4.39)} \quad (4.81)$$

$$\lambda_2 \leq \lambda_2 \quad \text{replaces(eq 4.40, 4.41, 4.42, 4.43, 4.44, 4.45)} \quad (4.82)$$

Capacity Constraints:**Mill Capacity Constraints:**

$$\lambda_1 + \lambda_3 + \lambda_3 \leq 1 \quad \text{replaces(eq 4.46)} \quad (4.83)$$

$$(\lambda_2 - \lambda_1) + (\lambda_4 - \lambda_3) + (\lambda_4 - \lambda_3) \leq 1 \quad \text{replaces(eq 4.47)} \quad (4.84)$$

Leach Capacity Constraints:

$$(\lambda_2 - \lambda_2) + (\lambda_2 - \lambda_4) + (\lambda_2 - \lambda_4) \leq 1 \quad \text{replaces(eq 4.48)} \quad (4.85)$$

Total Mining Capacity Constraints:

$$\begin{aligned} \lambda_1 + \lambda_1 + \lambda_1 + (\lambda_1 - \lambda_1) + (\lambda_1 - \lambda_1) + \lambda_1 + \lambda_3 + (\lambda_3 - \lambda_3) + (\lambda_3 - \lambda_3) + \lambda_3 \\ + (\lambda_3 - \lambda_3) + (\lambda_3 - \lambda_3) \leq 3 \end{aligned} \quad \text{replaces(eq 4.49)} \quad (4.86)$$

$$\begin{aligned} \lambda_2 + \lambda_2 + (\lambda_2 - \lambda_1) + (\lambda_2 - \lambda_2) + (\lambda_2 - \lambda_2) + \lambda_2 + (\lambda_4 - \lambda_3) + (\lambda_2 - \lambda_4) \\ + (\lambda_2 - \lambda_2) + (\lambda_4 - \lambda_3) + (\lambda_2 - \lambda_4) + (\lambda_2 - \lambda_2) \leq 3 \end{aligned} \quad \text{replaces(eq 4.50)} \quad (4.87)$$

Variable Restrictions:

$$0 \leq \lambda_1 \leq 1 \quad (4.88)$$

$$0 \leq \lambda_2 \leq 1 \quad (4.89)$$

$$0 \leq \lambda_3 \leq 1 \quad (4.90)$$

$$0 \leq \lambda_4 \leq 1 \quad (4.91)$$

After eliminating the redundant constraints and finalizing the cost calculations for the objective function, simplified mathematical model for the master problem in the following is obtained.

Objective Function:

$$\text{Max } Z = -2.3\lambda_1 + 5.7\lambda_2 + 0.5\lambda_3 + 1.4\lambda_4 \quad (4.92)$$

Subject To:**Ensures Sequencing Across Time Periods:**

$$\lambda_1 \leq \lambda_2 \quad (4.93)$$

$$\lambda_3 \leq \lambda_4 \quad (4.94)$$

Ensures Sequencing Between Destinations Within Time Periods:

$$\lambda_4 \leq \lambda_2 \quad (4.95)$$

Block Sequencing Constraints:

$$\lambda_3 \leq \lambda_1 \quad (4.96)$$

Capacity Constraints:**Mill Capacity Constraints:**

$$\lambda_1 + 2\lambda_3 \leq 1 \quad (4.97)$$

$$\lambda_2 - \lambda_1 + 2\lambda_4 - 2\lambda_3 \leq 1 \quad (4.98)$$

Leach Capacity Constraints:

$$2\lambda_2 - 2\lambda_4 \leq 1 \quad (4.99)$$

Total Mining Capacity Constraints:

$$4\lambda_1 + 2\lambda_3 \leq 3 \quad (4.100)$$

$$6\lambda_2 - \lambda_1 - 2\lambda_3 \leq 3 \quad (4.101)$$

Variable Restrictions:

$$0 \leq \lambda_1 \leq 1 \quad (4.102)$$

$$0 \leq \lambda_2 \leq 1 \quad (4.103)$$

$$0 \leq \lambda_3 \leq 1 \quad (4.104)$$

$$0 \leq \lambda_4 \leq 1 \quad (4.105)$$

The solution to the master problem is achieved by implementing the simplex algorithm.

$$\lambda_1 = 1/3, \lambda_2 = 2/3, \lambda_3 = 1/3, \lambda_4 = 1/3$$

Each λ variable is essentially representing a set of z_{bd}^t variables from the original problem, henceforth the value of the λ variables can be disintegrated to the corresponding z_{bd}^t variables of the original problem. Since the contraction operation preserves the original problem structure, the objective function value of the master problem calculated by λ variables and the original problem calculated by replacing the solution values of the λ variables with the associated z_{bd}^t variables will be equal to each other.

The next step is to check if the dual values generated from the master problem solution is equal to the duals of the previous iteration to confirm the optimality condition. If not, then the previous steps will be repeated by solving the subproblem with the modified block values, orthogonalizing the new partition with the previous set of partitions and solving the master problem with the corresponding variables to the new set of partitions until the optimality conditions are satisfied. Although this example was not carried to optimality, it is believed that the steps illustrated provide a basic understanding of the BZ algorithm.

4.2 Pseudoflow Algorithm

The Pseudoflow algorithm is so far the fastest known solution algorithm for the network flow problems modeled as max-flow or min-cut type problems. The strength of the Pseudoflow algorithm is exploited in the BZ algorithm as well as the new integer solution algorithm. The subproblem of the BZ algorithm is a multi-period sequencing problem which constitutes a totally unimodular structure that allows the Pseudoflow algorithm to take the advantage of the underlying network structure. Moreover, the new integer solution algorithm further splits these subproblem max closures into single time period integer feasible sub-closures by iteratively solving the max flow problems with the Pseudoflow algorithm. As this integerizing step may require many iterations to satisfy the feasibility conditions, the fast working mechanism of the Pseudoflow algorithm motivates the implementation of the new integer solution algorithm on large scale open pit mining problems.

The theories behind the working mechanism of the Pseudoflow algorithm is extensively discussed in the original paper by Hochbaum (2008), therefore only the summary of the steps of the Pseudoflow algorithm will be presented. Moreover, the computational study of the Pseudoflow algorithm and the push-relabel algorithm is presented by Chandran and Hochbaum (2009). Also, the detailed review of the Pseudoflow algorithm is provided by Van Dunem (2016).

The specific steps of the single iteration of the Pseudoflow algorithm is summarized below (Hochbaum 2008, Van Dunem 2016):

1. Initialize the algorithm by selecting a normalized tree. One simple normalized tree corresponds to a Pseudoflow in G_{st} (s, t graph that contains two distinguished nodes s and t ; source and sink nodes respectively) saturating all arcs adjacent to the source $A(s)$ and all the arcs adjacent to the sink $A(t)$, and zero on all other arcs.
2. Find a residual arc from a strong set of nodes to a weak set of nodes called as merger arc. If such an arc does not exist, the current solution is optimal.
3. If residual arcs going from a strong node to a weak node exist then, the selected merger arc is appended to the tree, the excess arc of the strong merger branch is removed, and the strong branch is merged with the weak branch.

4. Push the entire excess of the respective strong branch along the unique path from the root of the strong branch to the root of the weak branch.
5. Split any arc encountered along the path described in 3) which does not have sufficient residual capacity to accommodate the amount pushed. The tail node of that arc becomes a root of a new strong branch with excess equal to the difference between the amount pushed and the residual capacity.
6. The process of pushing excess and splitting is called *normalization*. The residual capacity of the split arc is pushed further until it either reaches another arc to split or the deficit arc adjacent to the root of the weak branch.

4.3 The New Integer Solution Algorithm

The new integer solution algorithm is developed mainly by exploiting the strengths of the Bienstock-Zuckerberg algorithm outlined in the previous section. Importantly, the algorithm works towards optimality in conjunction with the BZ algorithm. Although the BZ algorithm is presently the most powerful decomposition algorithm to solve the LP relaxation of the mine production scheduling problems towards proven optimality, the solutions are fractional which results in impractical mine plans. However, the information that will lead solutions towards integrality is also preserved in each iteration. The new integer solution algorithm will enjoy the implicit use of this information together with all of the theories behind the working mechanism of the BZ algorithm while unlocking new integer solution spaces for the master problem to satisfy the integrality conditions. The algorithm ensures that every column solution generated by the subproblem is partitioned into components that are integer feasible to the capacity constraints of the original problem. This can be achieved simply by converting the original problem into a single time period max flow problem (ultimate pit problem) and parametrizing the block values. While it may take many iterations to be integer feasible to the capacity constraints, the Pseudoflow algorithm will process the underlying network structure quite fast which may result with solution times around 2-3 seconds per iteration. Since mine production scheduling problems also include blending constraints, the duals generated by solving the master problem at each iteration will guide the subproblem solutions towards feasibility for the blending constraints. At each iteration, the orthogonalization process will further increase the dimension of the solution space where integer feasible solutions for the master problem are also available. Furthermore, the contraction operation

will allow the master problem to work with a reduced number of rows and columns. This will ensure fast solution times to be achieved for the master problem. Once the optimal LP solution is achieved, there will be set of partitions or solution columns which are integer feasible to the original problem space. The master problem that results with the optimal LP solution can be solved with the integer feasible solution columns to generate an optimal integer solution. The flowchart of the new integer solution algorithm is shown in Figure 4.13.

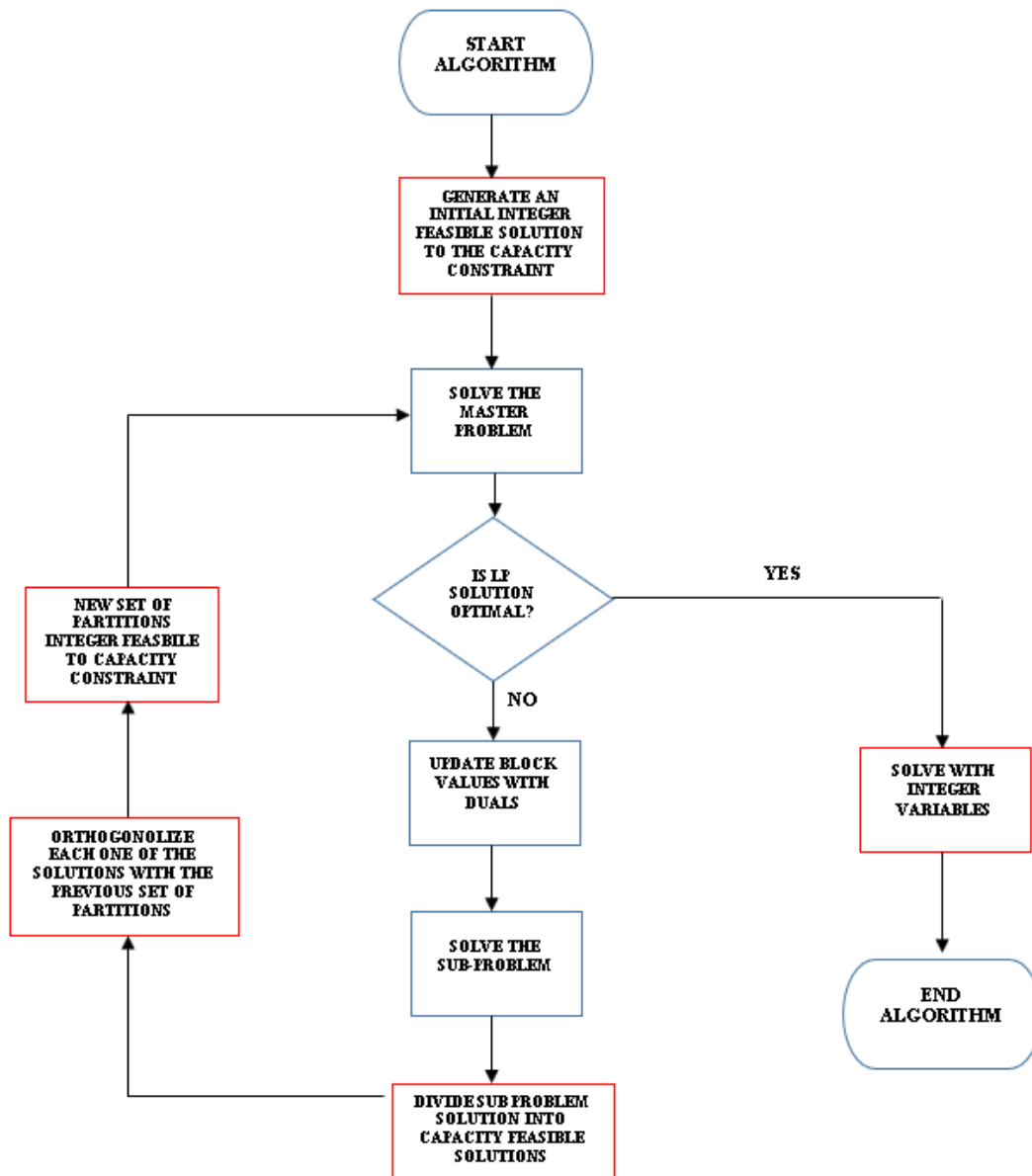


Figure 4.13: Flowchart of the solution methodology. The blue colored boxes represent the original BZ algorithm steps, the red colored boxes illustrate the steps of the new integer solution algorithm

As the algorithm proceeds with splitting the original problem into master and subproblems, the subproblem solutions play an important role in terms of defining the coordinates of the extreme point of a polytope located on the intersection of the hyperplanes. It is important to mention that the solution to the subproblem is not necessarily integer feasible to the capacity constraints of the original problem. In the regular BZ algorithm, if the partitions defining the solution space of the master problem are infeasible to the capacity constraints, the λ variables which are essentially the weight of each partition may result with fractional values to honor the requirements enforced by the capacity constraints. As this is very natural from a linear programming point of view, if the λ variables are defined as binary variables, the solution to the master problem may end up being infeasible. Henceforth, the subproblem solutions which are indeed max closures, must be partitioned into sub-closures that are integer feasible to the capacity constraints of the original problem. As it is illustrated in Figure 4.14, the outer polytope represents the subproblem solution space where the blue dots are the extreme points which may not appear in the capacity feasible region bounded with a green outline. The solution space to the original problem which is further defined by the hyperplanes representing the additional side constraints will be inside the capacity feasible region and is shown in Figure 4.14 outlined in black. While the yellow colored corner point defines the LP optimal solution to the original problem, the red points are the set of integer feasible points to the original problem space.

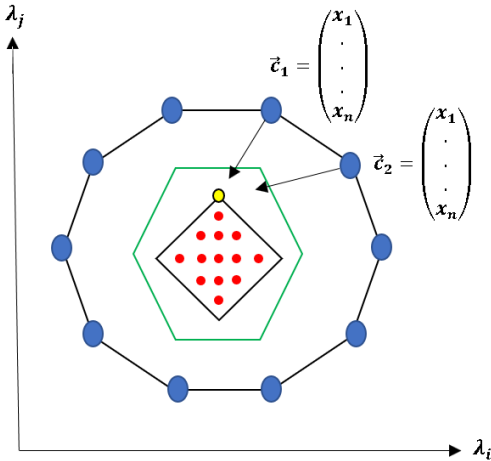


Figure 4.14: Cross section of a polytope illustrating the solution space of the new integer solution algorithm

At each iteration k , the solution to the subproblem generates a max closure \vec{C}_k which may violate the capacity constraints. Although \vec{C}_k only contains the blocks mined in the closure, we still know the actual periods the blocks are mined in since the subproblem is a multi-time period sequencing problem. If the problem has t number of periods, there may exist at least t number of capacity constraints. Hence, the number of blocks or total tonnage mined in closure \vec{C}_k at each period can be checked against the capacity requirements of that period. Any time period where the capacity requirements to be satisfied will be further split into sub-closures where each sub-closure honors the capacity requirements. The splitting may continue for a given period until the remaining blocks for that period satisfies the capacity requirements. The mining problem may consist of a number of capacity constraints in an upper bound form where it may not be possible to generate a closure which all constraints are binding. The underlying reason could be a gap problem which is inevitable in parametrization concepts if it exists. Nevertheless, we haven't observed any downside of not being exact with the processing and production requirements as long as the sub-closures are integer feasible. Eventually, by splitting \vec{C}_k into sub-closures, each period will end up having its own integer feasible sub-closures as long as the blocks are mined for that period in \vec{C}_k .

Theorem 4.1: *The set of sub-closures $\{\vec{S}_1, \vec{S}_2, \dots, \vec{S}_l\}$ obtained by partitioning the max closure of the subproblem \vec{C}_k will span the solution space governed by \vec{C}_k .*

Proof: Let \vec{C}_m be the subproblem closure for some iteration m and \vec{C}_n be the subproblem closure for some iteration n . Let $\{\vec{S}_1, \vec{S}_2, \dots, \vec{S}_k\}$ be the integer feasible sub-closures for iteration m and $\{\vec{S}_{k+1}, \vec{S}_{k+2}, \dots, \vec{S}_r\}$ be the integer feasible sub-closures for iteration n . By splitting the closure \vec{C}_m into $\{\vec{S}_1, \vec{S}_2, \dots, \vec{S}_k\}$ and \vec{C}_n into $\{\vec{S}_{k+1}, \vec{S}_{k+2}, \dots, \vec{S}_r\}$, the sets \vec{C}_m and \vec{C}_n are being partitioned into orthogonal components or axes as shown in Figure 4.15. Since $\{\vec{S}_1, \vec{S}_2, \dots, \vec{S}_k\}$ are orthogonal axes in k -dimensional space, any solution that exists in k -dimensional space can be represented by taking the linear combination of $\vec{S}_1, \vec{S}_2, \dots, \vec{S}_k$ therefore \vec{C}_m is captured. The same argument holds for \vec{C}_n since it is also decomposed into orthogonal axes of $\{\vec{S}_{k+1}, \vec{S}_{k+2}, \dots, \vec{S}_r\}$.

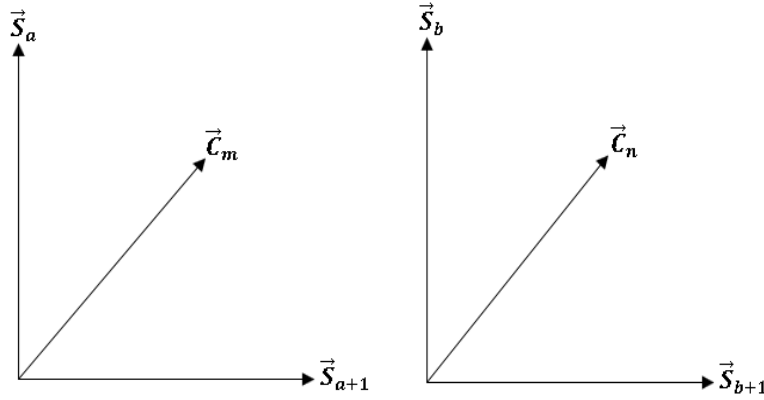


Figure 4.15: Orthogonal partitions of \vec{C}_m and \vec{C}_n on a plane

Assume that $\vec{C}_1, \vec{C}_2, \dots, \vec{C}_q$ are orthogonal columns and their orthogonal partitions are $\vec{S}_1, \vec{S}_2, \dots, \vec{S}_r$.

If $C = \{\vec{C}_1, \vec{C}_2, \dots, \vec{C}_q\}$ and $S = \{\vec{S}_1, \vec{S}_2, \dots, \vec{S}_r\}$, then their convex hull can be represented as:

$$Span(C) = \{\vec{C}_1\lambda_1 + \vec{C}_2\lambda_2 + \dots + \vec{C}_q\lambda_q\}$$

$$Span(S) = \{\vec{S}_1\lambda_1 + \vec{S}_2\lambda_2 + \dots + \vec{S}_r\lambda_r\}$$

$Span(C)$ takes place in q-dimensional space and $Span(S)$ covers the r-dimensional space where $r > q$, therefore $\rightarrow Span(C) \subseteq Span(S)$ which completes the proof that any linear combination of the sub-closures spans the solution space where the original closure takes place.

The max closure \vec{C}_k is the best solution column or a mining plan generated by solving the subproblem that enters the basis of the system in iteration k, processed as a function of dual variables obtained by solving the master problem. As it was proven by the authors of the BZ algorithm, the partitions generated by iteratively orthogonalizing the \vec{C}_k spans the solution space where the extreme point of the original problem or the optimal LP solution is captured.

Corollary 4.2: *The further partitioning of max closure \vec{C}_k into sub-closures $\{\vec{S}_1, \vec{S}_2 \dots \vec{S}_l\}$ will also lead to an optimal LP solution to the original problem due to Theorem 4.1.*

It is important to introduce the significance of the guidance of the dual values towards integer feasibility. We know that the net worth of the mining decision variables is determined as a function of dual vector π by the following equation $\bar{c}_i = c_i - \pi a_k$. The dual column π can be characterized as a function of the side constraint it is associated with. The side constraints can be categorized as capacity type and blending type constraints. Since the contribution of the dual

values towards subproblem solution space is determined by the side constraints, the role of the dual values will be analyzed for the capacity and blending type constraints separately. Capacity type constraints limit the number of blocks handled or processed in some way such as total mining production, processed tonnage, amount of material a crusher can handle, total available truck hours, waste dump capacity or amount of material stored in stockpile. Given a mine planning problem constrained with only upper bound capacity constraints, implementing the regular BZ algorithm to solve the subproblem by penalizing the block values with the capacity duals is in fact a well-known parametrization approach. The duals will modify the block values that exist in the capacity constraint with the same penalty.

Axiom 4.3: *As the penalty values increase, in the absence of gaps there may be an opportunity to obtain max closure \vec{C}_k which is integer feasible to the original problem capacity constraints.*

In the light of axiom 4.3, we can expect to generate an integer feasible solution towards capacity constraint space at some iteration if the dual values constantly increase in consecutive iterations.

Proposition 4.4: *The value of the duals does not constitute a monotonically non-decreasing pattern on consecutive iterations.*

Proof: The primal and dual formulation of the master problem is given in Figure 4.16. It has already been proven that the master problem objective function values are monotonically non-decreasing. Once the master problem is optimal for a particular iteration, the dual objective function value must be equal to the primal objective function value due to the strong duality theorem. This shows that since primal objective function values are monotonically non-decreasing, the dual objective function values must be also monotonically non-decreasing.

Primal	Dual
$Max Z = c^T V \lambda$	$Min D = \mu + h \pi$
<u>Subject To:</u>	<u>Subject To:</u>
$EV \lambda \leq 0 \longrightarrow \delta \text{ \{Sequencing Const.\}}$	$\mu IV + \pi HV + \delta EV \geq c^T V$
$IV \lambda \leq 1 \longrightarrow \mu \text{ \{Reserve Const.\}}$	$\mu \geq 0$
$HV \lambda \leq h \longrightarrow \pi \text{ \{Side Const.\}}$	$\pi \geq 0$
$\lambda \in \{0,1\}$	$\delta \geq 0$

Figure 4.16: Primal-dual relationship of the master problem

It is clear that the objective function coefficients of the dual of a problem $(1, h)$ are constant in every iteration which means that any improvement on the objective function value must be justified by increasing either one or both of the value of the duals π and μ . Also, the dual variable δ does not have any contribution to the objective function value. We also know that the dual values may be non-zero only if the primal constraints are binding. Let's investigate the behavior of the dual variables, first when a mine plan which is infeasible to the capacity constraint is obtained by solving the subproblem and passed to the master problem solution space. If the partitions are infeasible to the capacity constraint, the λ variables will be fractional which means some of the reserve constraints and some of the sequencing constraints will be non-binding. This will result with some of the μ variables having 0 values. Since the capacity constraints will be binding, all the π variables will be greater than or equal to 0. Hence, if consecutive iterations consist of infeasible partitions, π or μ must increase in order to justify the improvement of the objective function value. If the subproblem solution generates a mine plan feasible to the capacity constraint, we know that this is the most valuable integer feasible plan since in a max flow problem the penalized block values will be the best ones to be mined. Therefore, once this plan is passed to the master problem, any λ variable associated with this plan must become 1 in the master problem solution space. Any resource constraint that contain this λ variable must be a binding constraint which makes some of the μ variables become greater than or equal to 0. In this case, the justification of the improvement in the objective function value can be made by increasing either π or μ . In either case, there is no guarantee that the π variable will get a value greater than its former value, in fact the value of π may even decrease if the value of μ increases. Therefore, we cannot claim a monotonically non-decreasing pattern for the capacity dual values.

In a sense, the proposition 4.4 is very important to justify the need for an integerizing approach towards the subproblem solution space, since the regular BZ algorithm does not guarantee a mining plan integer feasible to the original problem capacity constraint space. Although an integer feasible region for the capacity constraints at each iteration can be secured, the original problem may consist of blending constraints where mining plans generated by the subproblem may need to be integer feasible for the blending constraints. Let's investigate the significance of the dual values generated for the blending type of constraints. First of all, the blending type of constraints can be average grade, material type proportions or risk proportions required at any type of processing destinations. In order to clarify the role of the duals for the

blending constraints, a mine plan will be assumed to be constrained with only lower bound blending constraints.

Proposition 4.5: *The duals generated for the blending constraints by solving the master problem drive the subproblem solution space towards feasibility for the blending constraints of the original problem.*

Proof: Let's assume the following is a lower bound grade blending constraint of the original problem for the time period 1.

$$\sum_{b \in B} (\underline{G}_p^1 - g_b) p_b x_{bp}^1 \leq 0 \quad p \in P \text{ (process destinations)} \quad (4.106)$$

Then the master problem formulation for the blending constraint would be as follows:

$$\sum_{s \in S} (\underline{G}_p^1 - g_s) p_s \lambda_s \leq 0 \quad p \in P \quad (4.107)$$

The master problem variable λ_s is essentially the weight of the closure or partition C_s for $s=1$ to S . The variable λ_s will only exist in the blending constraint for the closure C_s , if any blocks in the closure has an existing processing option. At each iteration, the optimal solution to the master problem generates a dual variable π corresponding to the blending constraint (4.107). Then, in order to run the subproblem, the blocks that has a potential value at process destination p will be penalized with the dual value π as shown in the following equation:

$$c_{bp}^{1*} = c_{bp}^1 - \pi_p^1 (\underline{G}_p^1 - g_b) \quad p \in P, b \in B \quad (4.108)$$

Let's analyze the contribution of dual value π_p^1 towards the modified value c_{bp}^{1*} . First of all, we know that $\pi_p^1 \geq 0$ since the blending constraint (4.107) is written as an upper bound constraint. Since c_{bp}^1 is constant and $\pi_p^1 \geq 0$, the value of the c_{bp}^{1*} will be heavily dependent on the individual grade of a block g_b . If $g_b > \underline{G}_p^1$, it means that the grade of the block is greater than the required average grade, therefore the value of the block will be increased where $c_{bp}^{1*} > c_{bp}^1$. If $g_b < \underline{G}_p^1$, then the grade of the block does not honor the average grade requirements, therefore the value of the block will be decreased where $c_{bp}^{1*} < c_{bp}^1$. This shows that the duals of the blending constraints will implicitly favor the blocks that are integer feasible to the average grade requirements and penalize any blocks that fail to meet the average grade criteria.

To summarize, proposition 4.4 shows that the closure C_k is not guaranteed to be integer feasible towards capacity constraints whereas proposition 4.5 shows that the duals obtained for the blending constraints drives the subproblem solution towards feasibility in the case of the blending constraints. In the light of proposition 4.4 and proposition 4.5, we can clearly see the need for an integerizing approach to make each closure C_k integer feasible towards capacity constraints.

The iterative integerizing process where the closure C_k is divided into integer feasible sub-closures is the backbone of this algorithm. The complexity of the integerizing process depends on the number of the capacity constraints. First of all, let's examine the case where there is only one capacity constraint that limits the number of ore blocks processed per period t with a right-hand side value of CAP^t . At each iteration, the subproblem closure C_k which represents the best mining plan needs to be split into partitions where each partition E_t represents the blocks mined in time period t . In other words, multi time period closure C_k is divided into single time period components such as $C_k = \{E_1, E_2 \dots E_T\}$. Furthermore, each closure $\{E_1, E_2 \dots E_T\}$ will be split into ore and waste components such as $\{O_1, O_2 \dots O_T\}$ and $\{W_1, W_2 \dots W_T\}$. In the next step, the number of blocks mined in each period's ore closure will be checked against CAP^t for $t=1$ to T . Any set $\{E_1, E_2 \dots E_T\}$ that satisfies the capacity requirements will be converted into a "by" form and stored in a set H_k that holds only the capacity feasible columns of iteration k . If at some period r where $|O_r| > CAP^r$, then the ore closure O_r has failed to meet with the capacity requirements CAP^r at period r which means the set O_r needs to be integerized. The integerizing process will be carried out iteratively similar to the parametrization approach until the feasibility conditions are met. Given "R" the set of infeasible periods and "S" the number of iterations, the penalty variable α_s^t is calculated by taking the average of the block values within set O_r as shown in the following:

$$\alpha_s^t = \frac{\sum_{b \in O_t} c_b^t}{|O_t|} \quad t \in R, s \in S \quad (4.109)$$

Once α_s^t is calculated, the original block values of the blocks in set O_r will be penalized as follows:

$$c_{bs}^{t*} = c_b^t - \alpha_s^t \quad b \in O_t, t \in R \quad (4.110)$$

A single time period block sequencing problem will be formulated with the blocks in the set E_r by using the modified block values c_{bs}^{t*} . Since the problem has an underlying network structure, the Pseudoflow algorithm will be used to solve the problem. Let I_s^r be the set of blocks mined with

the modified block values c_{bs}^{t*} and let Ore_s^r be the set of ore blocks in set I_s^r . If $|Ore_s^r| > CAP^r$, then the new set I_s^r is infeasible therefore the blocks in the set E_r should be penalized further by increasing the value of α_s^t until $|Ore_s^r| \leq CAP^r$. Once the ore blocks in set I_s^r satisfy the feasibility conditions of the capacity constraint, they should be taken out from the set E_r such that $E_r' = E_r \setminus I_s^r$ and place in the integer feasible partition set H_k . The set of ore blocks O_r' inside the set of remaining blocks E_r' will be checked against CAP^r . If it fails the capacity requirement, then the block values in set O_r' are penalized with a new α_s^t calculated by taking the average of the block values within set O_r' and the single time period network problem is solved for E_r' . The rest of the process will repeat similarly as mentioned above until the remaining blocks are integer feasible for the capacity constraints. The steps outlined above to integerize the subproblem solution so that each sub-closure meets the capacity constraint requirements can be implemented even if the mine plan consists of multi capacity destinations. Let's assume that there are p number of processing destinations with a capacity requirement of CAP_p^t . Similar to the single destination problem, the subproblem closure C_k will be split into period closures $\{E_1, E_2 \dots E_T\}$ where each period closure will be split into ore closures per destination such as $\{O_1^p, O_2^p \dots O_T^p\}$. The penalty variable α_{sp}^t will be calculated by taking the average of the block values for each destination p within set

O_t^p as shown in the following:

$$\alpha_{sp}^t = \frac{\sum_{b \in O_t^p} c_{bsp}^t}{|O_t^p|} \quad t \in R, s \in S, p \in P \quad (4.111)$$

The original block values at destination p, in set O_t^p will be penalized as follows:

$$c_{bsp}^{t*} = c_{bsp}^t - \alpha_{sp}^t \quad b \in O_t^p, t \in R, p \in P \quad (4.112)$$

This shows that the ore blocks are penalized based on the average value of the blocks in the predetermined destinations accomplished by the dual values before solving the subproblem. For example, if there exist 2 processing destinations such as mill and leach. Then, the blocks designated to mill will be penalized with the average block values at mill and the leach blocks will be penalized with the average value of the leach blocks. The rest of the steps will be similar to the single destination case except that now each period's ore set $\{O_1^p, O_2^p \dots O_T^p\}$ needs to meet with the capacity requirements of each destination p at each period t which is CAP_p^t .

Once the subproblem closure or the best mining plan C_k generated under the direction of dual variables is integerized for the capacity constraints of the original problem, the set of integer feasible columns H_k is ready to be orthogonalized with the previous iteration's set of orthogonal partitions V_{k-1} . It is important to highlight that the refinement rules introduced in the BZ algorithm shown in the previous section are useful when a single column is orthogonalized with a set of partitions. The new integer solution algorithm requires new refinement rules since each integer feasible column in set H_k needs to be orthogonalized with each partition in set V^{k-1} ; in other words, a set of orthogonal integer feasible columns need to be orthogonalized with the previous iteration's set of partitions. Assume that V^{k-1} is comprised of $\{\vec{v}_1 \dots, \vec{v}_r\}$ and H_k is composed of $\{\vec{h}_1 \dots, \vec{h}_m\}$ where within each column, the blocks are labeled with 1 if they are part of the partition. First, we need to intersect each one of the columns in H_k with each one of the columns in V^{k-1} . If the blocks in column \vec{v}_j are also in column \vec{h}_b , then the blocks are placed in an intersection column \vec{i} and the intersection column \vec{i} is placed in the intersection set \vec{I} as shown in (4.113). The set operation (4.114) shows that non-intersecting part of the column \vec{v}_j is a new column \vec{r} placed in set \vec{R} . If the blocks exist in column \vec{h}_b but do not exist in any of the columns in V^{k-1} , then the blocks are stored in column \vec{n} which is placed in set \vec{N} as shown in (4.115). In the end, the set of columns in \vec{I} , \vec{R} and \vec{N} unite to create the set of orthogonal partitions \vec{V}_k (4.116).

$$\vec{I} = \{\vec{i} = \vec{v}_j \wedge \vec{h}_b : 1 \leq j \leq r, 1 \leq b \leq m\} \quad (4.113)$$

$$\vec{R} = \left\{ \vec{r} = \vec{v}_j \setminus \left(\sum_{b=1}^{mxr} \vec{I}_b \right) : 1 \leq j \leq r \right\} \quad (4.114)$$

$$\vec{N} = \left\{ \vec{n} = \vec{h}_b \setminus \left(\sum_{j=1}^r \vec{v}_j \right) : 1 \leq b \leq m \right\} \quad (4.115)$$

$$\vec{V}_k = \{\vec{I}\} \cup \{\vec{R}\} \cup \{\vec{N}\} \quad (4.116)$$

The orthogonalization process leads to a higher dimensional solution space spanned by the linear combination of each partition $\{\vec{v}_1, \dots, \vec{v}_s\}$ in set \vec{V}_k with the variables $\lambda_i \forall i = 1 \dots s$. Since the orthogonal partitions are derived from the set of integer feasible columns H_k , each partition in set \vec{V}_k is guaranteed to be integer feasible for the capacity constraints. Furthermore, even if the columns in H_k are integer feasible for the blending constraints, after the orthogonalization some

of the columns may become infeasible. Let's assume set $\overrightarrow{h_1} = \{x_1, x_2 \dots x_m\}$ where $\overrightarrow{h_1}$ satisfies the blending constraints. Let the orthogonal partitions derived from $\overrightarrow{h_1}$ be $\overrightarrow{v_1} = \{x_1, x_2 \dots x_n\}$ and $\overrightarrow{v_2} = \{x_n, x_{n+1} \dots x_m\}$ where $|\overrightarrow{v_1}| + |\overrightarrow{v_2}| = m$. Since $\overrightarrow{h_1}$ is integer feasible for the blending constraints, $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$ have a mutually exclusive relationship in terms of being infeasible towards blending constraints. Because $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$ are sub-partitions of $\overrightarrow{h_1}$, based on theorem 4.1, any solution space governed by $\overrightarrow{h_1}$ will be captured from the span of $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$. In other words, the blend of $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$ still preserves the feasibility conditions for the blending constraints.

Once the optimality conditions mentioned in the BZ section are satisfied, the optimal LP solution will be generated. The set of partitions generated in the final iteration will play a key role in achieving an integer solution for the original problem. It is important to state that the final set of partitions may be comprised of capacity feasible and combination of blending feasible and infeasible columns. The relationship between the original problem constraint space and the span of the final set of partitions is illustrated in Figure 4.17. The blue colored region represents the solution space generated by the partitions whereas the outer polytope is the solution space of the original problem. It should be pointed out that the linear combination of the partitions captures a much smaller space where the optimal LP solution which is the extreme point colored with yellow is included as well as some of the integer solutions shown with red color. This shows that once the LP optimal solution is captured, the algorithm is ready to develop an optimal integer solution within the span of the final set of partitions. This can be simply done by declaring the final set of λ variables as binary variables and resolving the last iteration's master problem. The reduced number of rows and columns in the master problem allow the integer solution algorithms of CPLEX to exploit the underlying mathematical structure and obtain an integer solution very fast. Although the final integer solutions are optimal integer solutions within the solution space of the final partitions, it cannot be proven that the solutions are true optimal integer solutions of the original problem. Nevertheless, it was mentioned by the authors the BZ algorithm that the integrality gap between the integer programming solution and the linear programming relaxation is often small. The results indicate that there is a high potential for achieving an integrality gap of less than 1% for the large-scale mining problems constrained with capacity and blending constraints which will be presented in the next chapter.

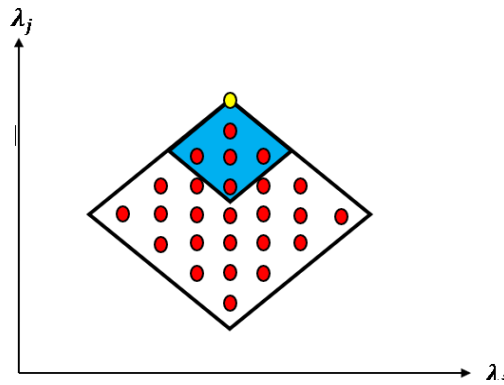


Figure 4.17: Solution space of the original problem and the final set of partitions

4.3.1 Small 2D Example

In this section, the implementation of the new integer solution algorithm will be demonstrated on a small 2D example. For the clarity of the example, the destinations of the blocks are pre-determined which means that if the block is classified as an ore it must be processed once it is mined, otherwise it must be wasted. Figure 4.18 and Figure 4.19 illustrates the deposit characteristics and the economic block model respectively, and Figure 4.20 shows the production scheduling requirements.

	1	2	3	4	5	6	7
1	WASTE	ORE	WASTE	ORE	WASTE	ORE	WASTE
2		ORE	ORE	WASTE	ORE	ORE	
3			ORE	ORE	ORE		

Figure 4.18: Classification of the blocks into ore and waste

	1	2	3	4	5	6	7
1	$c_{11}^1 = -2$	$c_{12}^1 = 4$	$c_{13}^1 = -2$	$c_{14}^1 = 4$	$c_{15}^1 = -1$	$c_{16}^1 = 3$	$c_{17}^1 = -2$
2		$c_{22}^1 = 8$	$c_{23}^1 = 3$	$c_{24}^1 = -5$	$c_{25}^1 = 1$	$c_{26}^1 = 2$	
3			$c_{31}^1 = 8$	$c_{32}^1 = 4$	$c_{33}^1 = 2$		

Figure 4.19: Undiscounted economic values of the blocks.

PERIODS	Process Capacity	Mine Life	3 years
	Blocks	Slope Angle	45⁰
Period 1	3	Discount Factor	12.50%
Period 2	3		
Period 3	3		

Figure 4.20: Production scheduling requirements.

In this example, only a single process capacity constraint will be used. The number of ore blocks mined and processed per period cannot exceed 3 ore blocks. The life of mine is 3 years and the pit walls must not exceed 45 degrees. The new integer solution algorithm embedded into BZ algorithm will be implemented in order to generate an optimal mining sequence. The original model formulation with the variables derived in the previous chapter is given below.

Original Problem Formulation:

z_{bd}^t = the fraction of block b, that is extracted by time period t

Objective Function:

$$\begin{aligned}
Max Z = & z_{11}^1(c_{11}^1 - c_{11}^2) + z_{12}^1(c_{12}^1 - c_{12}^2) + z_{13}^1(c_{13}^1 - c_{13}^2) + z_{14}^1(c_{14}^1 - c_{14}^2) \\
& + z_{15}^1(c_{15}^1 - c_{15}^2) + z_{16}^1(c_{16}^1 - c_{16}^2) + z_{17}^1(c_{17}^1 - c_{17}^2) + z_{22}^1(c_{22}^1 - c_{22}^2) \\
& + z_{23}^1(c_{23}^1 - c_{23}^2) + z_{24}^1(c_{24}^1 - c_{24}^2) + z_{25}^1(c_{25}^1 - c_{25}^2) + z_{26}^1(c_{26}^1 - c_{26}^2) \\
& + z_{33}^1(c_{33}^1 - c_{33}^2) + z_{34}^1(c_{34}^1 - c_{34}^2) + z_{35}^1(c_{35}^1 - c_{35}^2) + z_{11}^2(c_{11}^2 - c_{11}^3) \\
& + z_{12}^2(c_{12}^2 - c_{12}^3) + z_{13}^2(c_{13}^2 - c_{13}^3) + z_{14}^2(c_{14}^2 - c_{14}^3) + z_{15}^2(c_{15}^2 - c_{15}^3)
\end{aligned} \tag{4.117}$$

$$\begin{aligned}
& +z_{16}^2(c_{16}^2 - c_{16}^3) + z_{17}^2(c_{17}^2 - c_{17}^3) + z_{22}^2(c_{22}^2 - c_{22}^3) + z_{23}^2(c_{23}^2 - c_{23}^3) \\
& +z_{24}^2(c_{24}^2 - c_{24}^3) + z_{25}^2(c_{25}^2 - c_{25}^3) + z_{26}^2(c_{26}^2 - c_{26}^3) + z_{33}^2(c_{33}^2 - c_{33}^3) \\
& +z_{34}^2(c_{34}^2 - c_{34}^3) + z_{35}^2(c_{35}^2 - c_{35}^3) + z_{11}^3c_{11}^3 + z_{12}^3c_{12}^3 + z_{13}^3c_{13}^3 \\
& +z_{14}^3c_{14}^3 + z_{15}^3c_{15}^3 + z_{16}^3c_{16}^3 + z_{17}^3c_{17}^3 + z_{22}^3c_{22}^3 + z_{23}^3c_{23}^3 + z_{24}^3c_{24}^3 \\
& +z_{25}^3c_{25}^3 + z_{26}^3c_{26}^3 + z_{33}^3c_{33}^3 + z_{34}^3c_{34}^3 + z_{35}^3c_{35}^3
\end{aligned}$$

Subject To:

Ensures Sequencing Across Time Periods:

$z_{11}^1 \leq z_{11}^2$	(4.118)	$z_{11}^2 \leq z_{11}^3$	(4.119)
$z_{12}^1 \leq z_{12}^2$	(4.120)	$z_{12}^2 \leq z_{12}^3$	(4.121)
$z_{13}^1 \leq z_{13}^2$	(4.122)	$z_{13}^2 \leq z_{13}^3$	(4.123)
$z_{14}^1 \leq z_{14}^2$	(4.124)	$z_{14}^2 \leq z_{14}^3$	(4.125)
$z_{15}^1 \leq z_{15}^2$	(4.126)	$z_{15}^2 \leq z_{15}^3$	(4.127)
$z_{16}^1 \leq z_{16}^2$	(4.128)	$z_{16}^2 \leq z_{16}^3$	(4.129)
$z_{17}^1 \leq z_{17}^2$	(4.130)	$z_{17}^2 \leq z_{17}^3$	(4.131)
$z_{22}^1 \leq z_{22}^2$	(4.132)	$z_{22}^2 \leq z_{22}^3$	(4.133)
$z_{23}^1 \leq z_{23}^2$	(4.134)	$z_{23}^2 \leq z_{23}^3$	(4.135)
$z_{24}^1 \leq z_{24}^2$	(4.136)	$z_{24}^2 \leq z_{24}^3$	(4.137)
$z_{25}^1 \leq z_{25}^2$	(4.138)	$z_{25}^2 \leq z_{25}^3$	(4.139)
$z_{26}^1 \leq z_{26}^2$	(4.140)	$z_{26}^2 \leq z_{26}^3$	(4.141)
$z_{33}^1 \leq z_{33}^2$	(4.142)	$z_{33}^2 \leq z_{33}^3$	(4.143)
$z_{34}^1 \leq z_{34}^2$	(4.144)	$z_{34}^2 \leq z_{34}^3$	(4.145)
$z_{35}^1 \leq z_{35}^2$	(4.146)	$z_{35}^2 \leq z_{35}^3$	(4.147)

Block Sequencing Constraints:

$z_{22}^1 \leq z_{11}^1$	(4.148)	$z_{22}^2 \leq z_{11}^2$	(4.149)	$z_{22}^3 \leq z_{11}^3$	(4.150)
$z_{22}^1 \leq z_{12}^1$	(4.151)	$z_{22}^2 \leq z_{12}^2$	(4.152)	$z_{22}^3 \leq z_{12}^3$	(4.153)
$z_{22}^1 \leq z_{13}^1$	(4.154)	$z_{22}^2 \leq z_{13}^2$	(4.155)	$z_{22}^3 \leq z_{13}^3$	(4.156)
$z_{23}^1 \leq z_{12}^1$	(4.157)	$z_{23}^2 \leq z_{12}^2$	(4.158)	$z_{23}^3 \leq z_{12}^3$	(4.159)
$z_{23}^1 \leq z_{13}^1$	(4.160)	$z_{23}^2 \leq z_{13}^2$	(4.161)	$z_{23}^3 \leq z_{13}^3$	(4.162)
$z_{23}^1 \leq z_{14}^1$	(4.163)	$z_{23}^2 \leq z_{14}^2$	(4.164)	$z_{23}^3 \leq z_{14}^3$	(4.165)
$z_{24}^1 \leq z_{13}^1$	(4.166)	$z_{24}^2 \leq z_{13}^2$	(4.167)	$z_{24}^3 \leq z_{13}^3$	(4.168)

$$\begin{array}{llll}
z_{24}^1 \leq z_{14}^1 & (4.169) & z_{24}^2 \leq z_{14}^2 & (4.170) & z_{24}^3 \leq z_{14}^3 & (4.171) \\
z_{24}^1 \leq z_{15}^1 & (4.172) & z_{24}^2 \leq z_{15}^2 & (4.173) & z_{24}^3 \leq z_{15}^3 & (4.174) \\
z_{25}^1 \leq z_{14}^1 & (4.175) & z_{25}^2 \leq z_{14}^2 & (4.176) & z_{25}^3 \leq z_{14}^3 & (4.177) \\
z_{25}^1 \leq z_{15}^1 & (4.178) & z_{25}^2 \leq z_{15}^2 & (4.179) & z_{25}^3 \leq z_{15}^3 & (4.180) \\
z_{25}^1 \leq z_{16}^1 & (4.181) & z_{25}^2 \leq z_{16}^2 & (4.182) & z_{25}^3 \leq z_{16}^3 & (4.183) \\
z_{26}^1 \leq z_{15}^1 & (4.184) & z_{26}^2 \leq z_{15}^2 & (4.185) & z_{26}^3 \leq z_{15}^3 & (4.186) \\
z_{26}^1 \leq z_{16}^1 & (4.187) & z_{26}^2 \leq z_{16}^2 & (4.188) & z_{26}^3 \leq z_{16}^3 & (4.189) \\
z_{26}^1 \leq z_{17}^1 & (4.190) & z_{26}^2 \leq z_{17}^2 & (4.191) & z_{26}^3 \leq z_{17}^3 & (4.192) \\
z_{33}^1 \leq z_{22}^1 & (4.193) & z_{33}^2 \leq z_{22}^2 & (4.194) & z_{33}^3 \leq z_{22}^3 & (4.195) \\
z_{33}^1 \leq z_{23}^1 & (4.196) & z_{33}^2 \leq z_{23}^2 & (4.197) & z_{33}^3 \leq z_{23}^3 & (4.198) \\
z_{33}^1 \leq z_{24}^1 & (4.199) & z_{33}^2 \leq z_{24}^2 & (4.200) & z_{33}^3 \leq z_{24}^3 & (4.201) \\
z_{34}^1 \leq z_{23}^1 & (4.202) & z_{34}^2 \leq z_{23}^2 & (4.203) & z_{34}^3 \leq z_{23}^3 & (4.204) \\
z_{34}^1 \leq z_{24}^1 & (4.205) & z_{34}^2 \leq z_{24}^2 & (4.206) & z_{34}^3 \leq z_{24}^3 & (4.207) \\
z_{34}^1 \leq z_{25}^1 & (4.208) & z_{34}^2 \leq z_{25}^2 & (4.209) & z_{34}^3 \leq z_{25}^3 & (4.210) \\
z_{35}^1 \leq z_{24}^1 & (4.211) & z_{35}^2 \leq z_{24}^2 & (4.212) & z_{35}^3 \leq z_{24}^3 & (4.213) \\
z_{35}^1 \leq z_{25}^1 & (4.214) & z_{35}^2 \leq z_{25}^2 & (4.215) & z_{35}^3 \leq z_{25}^3 & (4.216) \\
z_{35}^1 \leq z_{26}^1 & (4.217) & z_{35}^2 \leq z_{26}^2 & (4.218) & z_{35}^3 \leq z_{26}^3 & (4.219)
\end{array}$$

Capacity Constraints:

Process Capacity Constraints:

$$z_{12}^1 + z_{14}^1 + z_{16}^1 + z_{22}^1 + z_{23}^1 + z_{25}^1 + z_{26}^1 + z_{33}^1 + z_{34}^1 + z_{35}^1 \leq 3 \quad (4.220)$$

$$\begin{aligned}
& (z_{12}^2 - z_{12}^1) + (z_{14}^2 - z_{14}^1) + (z_{16}^2 - z_{16}^1) + (z_{22}^2 - z_{22}^1) + (z_{23}^2 - z_{23}^1) + (z_{25}^2 - z_{25}^1) \\
& + (z_{26}^2 - z_{26}^1) + (z_{33}^2 - z_{33}^1) + (z_{34}^2 - z_{34}^1) + (z_{35}^2 - z_{35}^1) \leq 3
\end{aligned} \quad (4.221)$$

$$\begin{aligned}
& (z_{12}^3 - z_{12}^2) + (z_{14}^3 - z_{14}^2) + (z_{16}^3 - z_{16}^2) + (z_{22}^3 - z_{22}^2) + (z_{23}^3 - z_{23}^2) + (z_{25}^3 - z_{25}^2) \\
& + (z_{26}^3 - z_{26}^2) + (z_{33}^3 - z_{33}^2) + (z_{34}^3 - z_{34}^2) + (z_{35}^3 - z_{35}^2) \leq 3
\end{aligned} \quad (4.222)$$

Variable Restrictions:

$$0 \leq z_{ij}^t \leq 1 \quad \forall i \in \{1,2,3\}, \forall j \in \{1,2,3,4,5,6,7\}, \forall t \in \{1,2,3\} \quad (4.223)$$

Iteration 1

The first step of the algorithm will start by splitting the ultimate pit closure \vec{C}_1 into integer feasible partitions \vec{h}_k , where each partition represents a single time period mine plan that honors the capacity requirements of a given period. We know that \vec{C}_1 is composed of blocks in the ultimate

pit such as $\vec{C}_1 = \{x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}, x_{33}, x_{34}, x_{35}\}$, where the ore blocks are $\vec{O}_1 = \{x_{12}, x_{14}, x_{16}, x_{22}, x_{23}, x_{25}, x_{26}, x_{33}, x_{34}, x_{35}\}$. The process capacity is limited to 3 blocks per period such that $CAP^t = 3$ for $t=1,2,3$. Since $|\vec{O}_1| > CAP^1$, the blocks in ore set \vec{O}_1 will be penalized with α_s^t which is the average value of the ore blocks in set \vec{O}_1 at period t for iteration s as shown below:

$$\alpha_1^1 = \frac{\sum_{b \in O_1} c_b^1}{|\vec{O}_1|} = \frac{39}{10} = 3.9 \quad (4.224)$$

Once, the original block values c_b^1 as shown in Figure 4.19 are penalized with $\alpha_1^1 = 3.9$, a single time period block sequencing problem is modeled as a network flow problem and the Pseudoflow algorithm is implemented to solve the problem. The solution to the problem is shown in Figure 4.21. Let \vec{h}_1 be the blocks in the parametrized plan and \overrightarrow{Ore}_1 be the set of ore blocks mined in the plan. Since $|\overrightarrow{Ore}_1| = 3$, the capacity requirements are satisfied, and we are done with period 1.

PERIOD 1							
	1	2	3	4	5	6	7
1	-2	0.1	-2	0.1	-1	-0.9	-2
2		4.1	-0.9	-5	-2.9	-1.9	
3			4.1	0.1	-1.9		

Figure 4.21: Iteration 1- Integer feasible mine plan for period 1 with penalized values

The initial set \vec{C}_1 which holds the blocks in the ultimate pit will be updated by extracting the integer feasible plan for period 1 such as $\vec{C}_2 = \vec{C}_1 \setminus \vec{h}_1$. The remaining blocks are shown in set \vec{C}_2 such as $\vec{C}_2 = \{x_{15}, x_{16}, x_{17}, x_{23}, x_{24}, x_{25}, x_{26}, x_{33}, x_{34}, x_{35}\}$ and the remaining ore blocks are $\vec{O}_2 = \{x_{16}, x_{23}, x_{25}, x_{26}, x_{33}, x_{34}, x_{35}\}$. Since $|\vec{O}_2| > CAP^2$, the ore blocks need to be penalized. This time the average value of the ore blocks α_1^2 in set \vec{O}_2 is calculated as 3.28. Once the original block values c_b^1 are penalized and the maximum flow network problem is solved, no ore blocks are mined. This may happen in real mining problems also where the penalty value is too high to allow an economic extraction sequence. The solution is incrementally adjusting the penalty values

until a feasible sequence is achieved. In this particular example, a penalty value of 2.9 will be picked and the process will be continued. The feasible mining sequence obtained by solving the network flow problem with the adjusted penalty value $\alpha_2^2 = 2.9$ is shown in Figure 4.22 where $|\overline{Ore}_1^2| = 2$. Since $|\overline{Ore}_1^2| < CAP^2$, the mined blocks are stored in set \vec{h}_2 .

PERIOD 2							
	1	2	3	4	5	6	7
1					-1	0.1	-2
2			0.1	-5	-1.9	-0.9	
3			5.1	1.1	-0.9		

Figure 4.22: Iteration 1- Integer feasible mine plan for period 2 with penalized values

Once the period 2 integer feasible mine plan is achieved, the remaining blocks are determined by extracting \vec{h}_2 from \vec{C}_2 such as $\vec{C}_3 = \vec{C}_2 \setminus \vec{h}_2$ where $\vec{C}_3 = \{x_{16}, x_{17}, x_{24}, x_{25}, x_{26}, x_{33}, x_{34}, x_{35}\}$ and the new ore set $\vec{O}_3 = \{x_{25}, x_{26}, x_{33}, x_{34}, x_{35}\}$. The ore blocks are further penalized because there are more blocks than the process capacity requirement for period 3; $|\vec{O}_3| > CAP^3$. Again, the average ore value 3.4 is too high for a penalty; therefore, it will be discounted. The penalty value $\alpha_2^3 = 2.3$ is selected and the max flow problem is solved. The resulting feasible mining sequence is shown in Figure 4.23. The number of ore blocks mined $|\overline{Ore}_1^3| = 3$ satisfies the capacity requirements of period 3, henceforth; the mined blocks are stored in set \vec{h}_3 .

PERIOD 3							
	1	2	3	4	5	6	7
1					-1		-2
2				-5	-1.9	-0.9	
3			5.1	1.1	-0.9		

Figure 4.23: Iteration 1- Integer feasible mine plan for period 3 with penalized values.

The pit will be updated by extracting the feasible region \vec{h}_3 from \vec{C}_3 such as $\vec{C}_4 = \vec{C}_3 \setminus \vec{h}_3$ as shown in Figure 4.24 where $\vec{C}_4 = \{x_{17}, x_{26}, x_{35}\}$ and the new ore set $\vec{O}_4 = \{x_{26}, x_{35}\}$. Since period 3 is the last period and the remaining ore blocks satisfy the capacity requirements; $|\vec{O}_4| < CAP^4$, the set of remaining blocks \vec{C}_4 will become a new partition \vec{h}_4 .

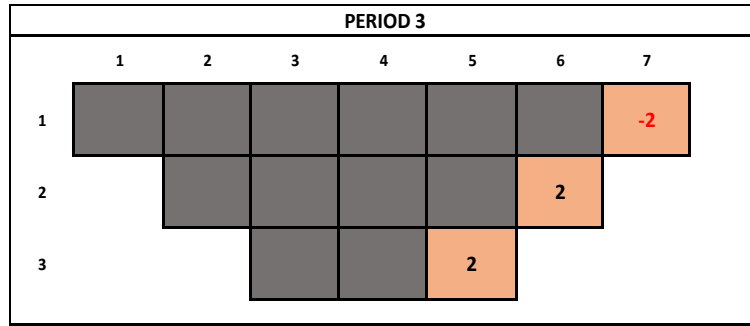


Figure 4.24: Iteration 1- Integer feasible mine plan for period 3 with penalized values.

Once the integer feasible pits $\vec{h}_1, \vec{h}_2, \vec{h}_3, \vec{h}_4$ are mapped into their physical locations as shown in Figure 4.25, the sequencing of the pits across the periods becomes apparent. It is clear that the yellow colored pit \vec{h}_1 needs to be mined in period 1 in order to extract the green colored pit \vec{h}_2 in period 2. The blue colored pit \vec{h}_3 can be mined in period 3, once \vec{h}_1 and \vec{h}_2 are taken. The pink colored pit \vec{h}_4 will become available in period 3 only if $\vec{h}_1, \vec{h}_2, \vec{h}_3$ are extracted. Another important fact is, the integer feasible pits $\vec{h}_1, \vec{h}_2, \vec{h}_3, \vec{h}_4$ are orthogonal which can be identified easily from Figure 4.25 that no blocks are shared between any pits. Since the columns $\vec{h}_1, \vec{h}_2, \vec{h}_3, \vec{h}_4$ are integer feasible and orthogonal, an initial set of partitions $\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{v}_4$ can be constructed as shown in Figure 4.26.

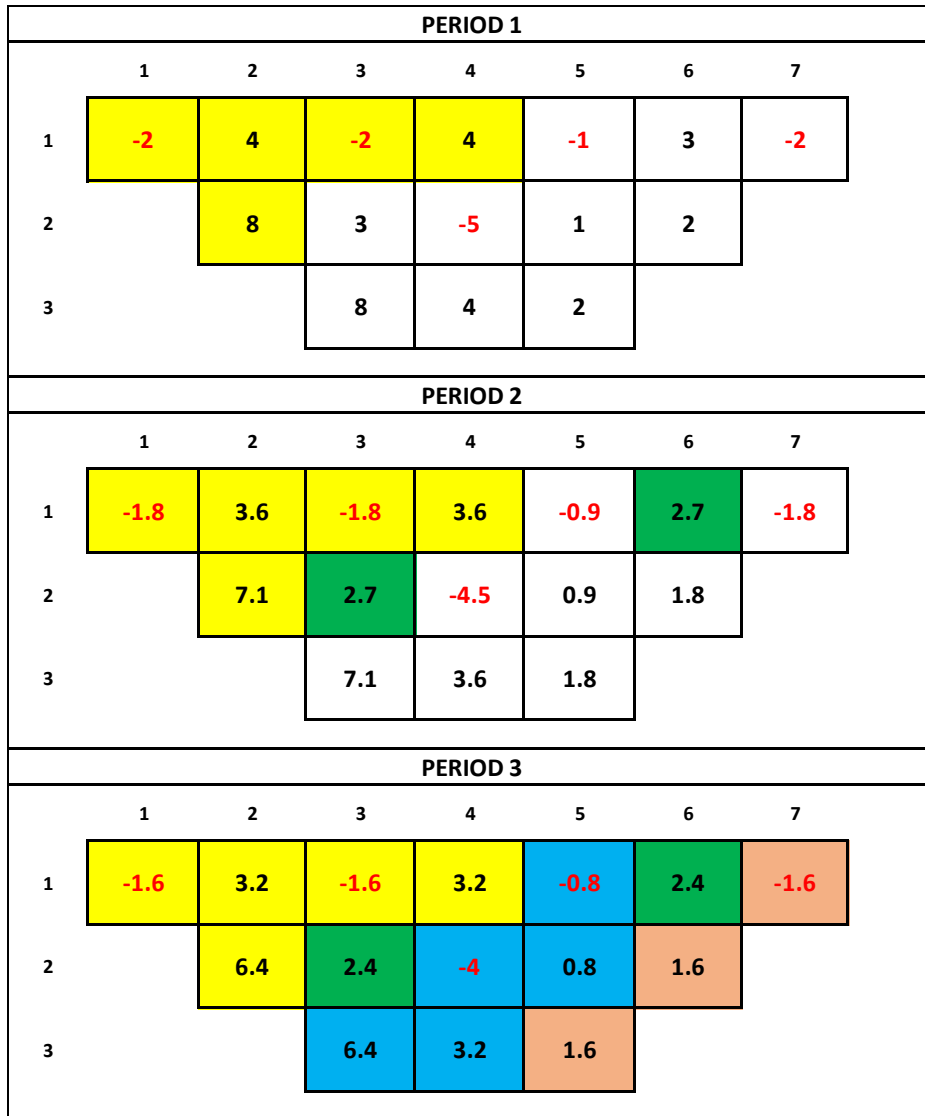


Figure 4.25: Iteration 1- Integer feasible pits per period. Period 1 pit is shown with yellow, period 2 pit is shown with green and period 3 pits are shown with blue and pink colors

z_b^t	$\rightarrow v_1$	$\rightarrow v_2$	$\rightarrow v_3$	$\rightarrow v_4$
11	1	0	0	0
12	1	0	0	0
13	1	0	0	0
14	1	0	0	0
15	0	0	0	0
16	0	0	0	0
17	0	0	0	0
22	1	0	0	0
23	0	0	0	0
24	0	0	0	0
25	0	0	0	0
26	0	0	0	0
33	0	0	0	0
34	0	0	0	0
35	0	0	0	0
11	1	0	0	0
12	1	0	0	0
13	1	0	0	0
14	1	0	0	0
15	0	0	0	0
16	0	1	0	0
17	0	0	0	0
22	1	0	0	0
23	0	1	0	0
24	0	0	0	0
25	0	0	0	0
26	0	0	0	0
33	0	0	0	0
34	0	0	0	0
35	0	0	0	0
11	1	0	0	0
12	1	0	0	0
13	1	0	0	0
14	1	0	0	0
15	0	0	1	0
16	0	1	0	0
17	0	0	0	1
22	1	0	0	0
23	0	1	0	0
24	0	0	1	0
25	0	0	1	0
26	0	0	0	1
33	0	0	1	0
34	0	0	1	0
35	0	0	0	1

Figure 4.26: Iteration 1- Initial set of integer feasible orthogonal partitions. Colors represent the blocks considered in the partition

The columns of the orthogonal partitions $\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{v}_4$ will lead to a contraction operation as mentioned before where each z_b^t variable that exists in the same partition will be replaced with the corresponding λ variable. Below is the master problem generated from the original problem by direct variable substitution.

$$\lambda_1 \rightarrow \vec{v}_1 = \{z_{11}^1, z_{12}^1, z_{13}^1, z_{14}^1, z_{22}^1, z_{11}^2, z_{12}^2, z_{13}^2, z_{14}^2, z_{22}^2, z_{11}^3, z_{12}^3, z_{13}^3, z_{14}^3, z_{22}^3\}$$

$$\lambda_2 \rightarrow \vec{v}_2 = \{z_{16}^2, z_{23}^2, z_{16}^3, z_{23}^3\}$$

$$\lambda_3 \rightarrow \vec{v}_3 = \{z_{15}^3, z_{24}^3, z_{25}^3, z_{33}^3, z_{34}^3\}$$

$$\lambda_4 \rightarrow \vec{v}_4 = \{z_{17}^3, z_{26}^3, z_{35}^3\}$$

Master Problem Formulation:

Objective Function:

$$\begin{aligned} \text{Max } Z = & \lambda_1(c_{11}^1 - c_{11}^2) + \lambda_1(c_{12}^1 - c_{12}^2) + \lambda_1(c_{13}^1 - c_{13}^2) + \lambda_1(c_{14}^1 - c_{14}^2) \\ & + \lambda_1(c_{22}^1 - c_{22}^2) + \lambda_1(c_{11}^2 - c_{11}^3) + \lambda_1(c_{12}^2 - c_{12}^3) + \lambda_1(c_{13}^2 - c_{13}^3) \\ & + \lambda_1(c_{14}^2 - c_{14}^3) + \lambda_2(c_{16}^2 - c_{16}^3) + \lambda_1(c_{22}^2 - c_{22}^3) + \lambda_2(c_{23}^2 - c_{23}^3) \\ & + \lambda_1 c_{11}^3 + \lambda_1 c_{12}^3 + \lambda_1 c_{13}^3 + \lambda_1 c_{14}^3 + \lambda_3 c_{15}^3 + \lambda_2 c_{16}^3 + \lambda_4 c_{17}^3 + \lambda_1 c_{22}^3 \\ & + \lambda_2 c_{23}^3 + \lambda_3 c_{24}^3 + \lambda_3 c_{25}^3 + \lambda_4 c_{26}^3 + \lambda_3 c_{33}^3 + \lambda_3 c_{34}^3 + \lambda_4 c_{35}^3 \end{aligned} \quad (4.225)$$

Subject To:

Block Sequencing Constraints:

$$\lambda_2 \leq \lambda_1 \quad (4.226)$$

$$\lambda_3 \leq \lambda_2 \quad (4.227)$$

$$\lambda_4 \leq \lambda_3 \quad (4.228)$$

Capacity Constraints:

Process Capacity Constraints:

$$\lambda_1 + \lambda_1 + \lambda_1 \leq 3 \quad (4.229)$$

$$(\lambda_1 - \lambda_1) + (\lambda_1 - \lambda_1) + \lambda_2 + (\lambda_1 - \lambda_1) + \lambda_2 \leq 3 \quad (4.230)$$

$$\begin{aligned} & (\lambda_1 - \lambda_1) + (\lambda_1 - \lambda_1) + (\lambda_2 - \lambda_2) + (\lambda_1 - \lambda_1) + (\lambda_2 - \lambda_2) + \lambda_3 \\ & + \lambda_4 + \lambda_3 + \lambda_3 + \lambda_4 \leq 3 \end{aligned} \quad (4.231)$$

Variable Restrictions:

$$0 \leq \lambda_i \leq 1 \quad \forall i \in \{1,2,3,4\} \quad (4.232)$$

Once the mathematical model is simplified, the following model is generated.

Objective Function:

$$Max Z = 12\lambda_1 + 5.4\lambda_2 + 5.6\lambda_3 + 1.6\lambda_4 \quad (4.233)$$

Subject To:

Block Sequencing Constraints:

$$\lambda_2 \leq \lambda_1 \quad (4.234)$$

$$\lambda_3 \leq \lambda_2 \quad (4.235)$$

$$\lambda_4 \leq \lambda_3 \quad (4.236)$$

Capacity Constraints:

Process Capacity Constraints:

$$3\lambda_1 \leq 3 \quad (4.237)$$

$$2\lambda_2 \leq 3 \quad (4.238)$$

$$3\lambda_3 + 2\lambda_4 \leq 3 \quad (4.239)$$

Variable Restrictions:

$$0 \leq \lambda_i \leq 1 \quad \forall i \in \{1,2,3,4\} \quad (4.240)$$

The solution to the model results with $\lambda_1 = 1$, $\lambda_2 = 1$, $\lambda_3 = 1$ and $\lambda_4 = 0$, which means that the value of the mining decision variables z_b^t in set $\vec{v}_1, \vec{v}_2, \vec{v}_3$ are also equal to 1. The NPV of the master problem is 23. The dual values generated for the capacity constraints (4.234), (4.235), and (4.236) are $\mu_1 = 5.8$, $\mu_2 = 0$, $\mu_3 = 1.87$ respectively. It is clear that since the constraints (4.234) and (4.236) are binding, $\mu_1, \mu_3 > 0$ and the non-binding constraint (4.235) resulted with $\mu_2 = 0$.

Iteration 2

The original block values of the ore blocks are penalized with the dual values. Period 1 ore block values will be penalized with μ_1 and period 3 ore block values will be penalized with μ_3 . Then, the subproblem which is a multi-time period sequencing problem is solved with the Pseudoflow algorithm. The subproblem solution which is currently the best mining plan is shown in Figure 4.27 where all the blocks are mined in period 2. Since this plan is not feasible for the period 2 capacity constraint, it should be partitioned into capacity feasible plans. The splitting process will be exactly same as iteration 1 because the closure obtained from the subproblem includes all the blocks. Moreover, the resulting partitions are equivalent to the period 3 partitions generated in Iteration 1 shown in Figure 4.25. The physical locations of the integer feasible pits are shown in Figure 4.28. The mathematical structure of the subproblem max closure and the

integer feasible sub-closures are shown in Figure 4.29 where the subproblem solution column \vec{s}_1 is decomposed into the integer feasible components $\vec{h}_1, \vec{h}_2, \vec{h}_3, \vec{h}_4$. It can be easily seen that $\vec{s}_1 = \vec{h}_1 \cup \vec{h}_2 \cup \vec{h}_3 \cup \vec{h}_4$. Furthermore, we know that the orthogonal partitions of the previous iteration are $\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{v}_4$. Once the new integer feasible partitions $\vec{h}_1, \vec{h}_2, \vec{h}_3, \vec{h}_4$ are appended to the previous set of partitions as shown in Figure 4.30, it is clear that the columns do not constitute an orthogonal structure. Henceforth, the orthogonalization process will take place between the two sets of partitions. The final set of partitions obtained from the orthogonalization process is demonstrated in Figure 4.31. Also, the blocks captured in each partition is colored and these partitions are mapped into their physical location as illustrated in Figure 4.32 to clarify the relationship between the orthogonal columns and the actual pit.

PERIOD 1							
	1	2	3	4	5	6	7
1	-2	-1.8	-2	-1.8	-1	-2.8	-2
2		2.2	-2.8	-5	-4.8	-3.8	
3			2.2	-1.8	-3.8		

PERIOD 2							
	1	2	3	4	5	6	7
1	-1.8	3.6	-1.8	3.6	-0.9	2.7	-1.8
2		7.1	2.7	-4.5	0.9	1.8	
3			7.1	3.6	1.8		

PERIOD 3							
	1	2	3	4	5	6	7
1	-1.6	1.3	-1.6	1.3	-0.8	0.5	-1.6
2		4.5	0.5	-4	-1.1	-0.3	
3			4.5	1.3	-0.3		

Figure 4.27: Iteration 2- Mine plan generated by solving the subproblem shown by yellow color

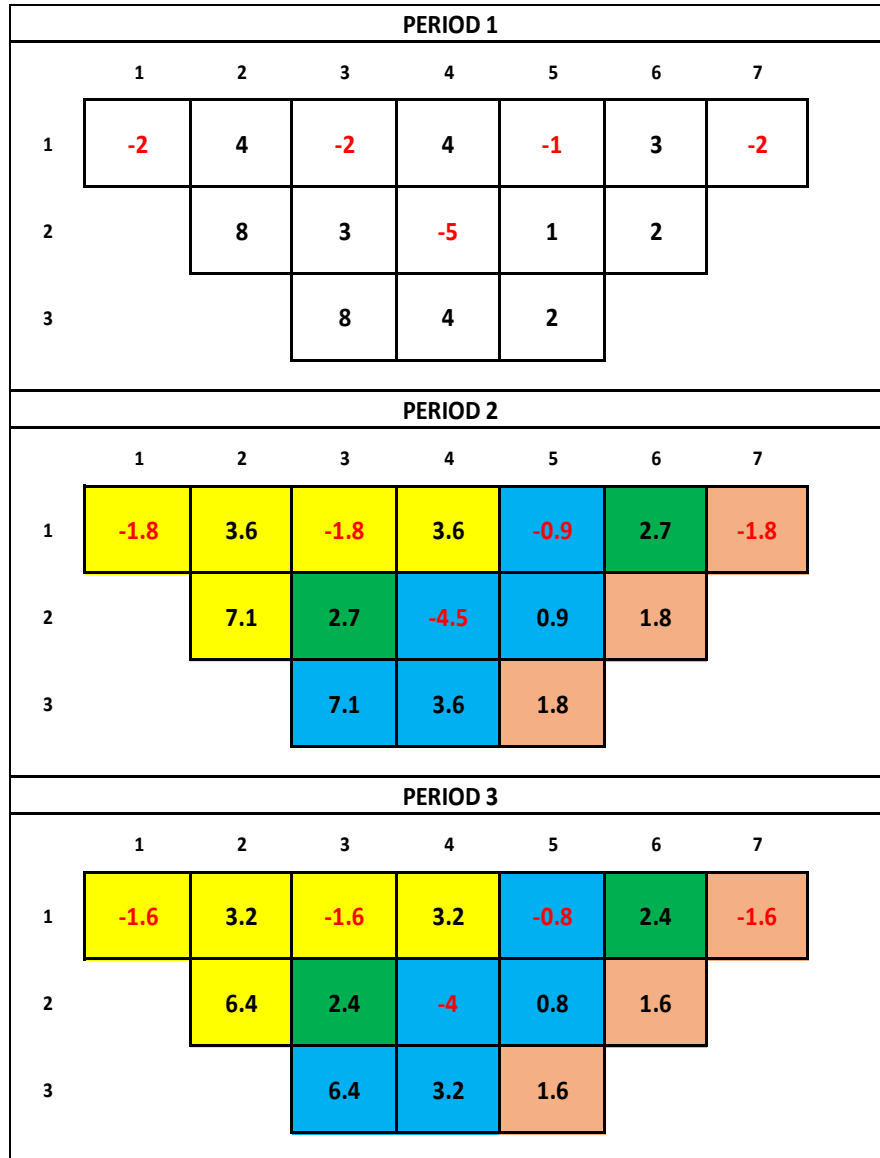


Figure 4.28: Iteration 2- Period 2 pit is split into integer feasible mine plans where each color represents a pit

	z_b^t	\vec{s}_1		z_b^t	\vec{h}_1	\vec{h}_2	\vec{h}_3	\vec{h}_4	
PERIOD 1	11	0	PERIOD 1	11	0	0	0	0	
	12	0		12	0	0	0	0	
	13	0		13	0	0	0	0	
	14	0		14	0	0	0	0	
	15	0		15	0	0	0	0	
	16	0		16	0	0	0	0	
	17	0		17	0	0	0	0	
	22	0		22	0	0	0	0	0
	23	0		23	0	0	0	0	0
	24	0		24	0	0	0	0	0
	25	0		25	0	0	0	0	0
	26	0		26	0	0	0	0	0
	33	0		33	0	0	0	0	0
	34	0		34	0	0	0	0	0
	35	0		35	0	0	0	0	0
PERIOD 2	11	1	PERIOD 2	11	1	0	0	0	
	12	1		12	1	0	0	0	
	13	1		13	1	0	0	0	
	14	1		14	1	0	0	0	
	15	1		15	0	0	1	0	
	16	1		16	0	1	0	0	
	17	1		17	0	0	0	1	
	22	1		22	1	0	0	0	
	23	1		23	0	1	0	0	
	24	1		24	0	0	1	0	
	25	1		25	0	0	1	0	
	26	1		26	0	0	0	1	
	33	1		33	0	0	1	0	
	34	1		34	0	0	1	0	
	35	1		35	0	0	0	1	
PERIOD 3	11	1	PERIOD 3	11	1	0	0	0	
	12	1		12	1	0	0	0	
	13	1		13	1	0	0	0	
	14	1		14	1	0	0	0	
	15	1		15	0	0	1	0	
	16	1		16	0	1	0	0	
	17	1		17	0	0	0	1	
	22	1		22	1	0	0	0	
	23	1		23	0	1	0	0	
	24	1		24	0	0	1	0	
	25	1		25	0	0	1	0	
	26	1		26	0	0	0	1	
	33	1		33	0	0	1	0	
	34	1		34	0	0	1	0	
	35	1		35	0	0	0	1	

Figure 4.29: Iteration 2- Subproblem solution column on the left is split into integer feasible columns on the right

	z_b^t	\vec{v}_1	\vec{v}_2	\vec{v}_3	\vec{v}_4
PERIOD 1	11	1	0	0	0
	12	1	0	0	0
	13	1	0	0	0
	14	1	0	0	0
	15	0	0	0	0
	16	0	0	0	0
	17	0	0	0	0
	22	1	0	0	0
	23	0	0	0	0
	24	0	0	0	0
	25	0	0	0	0
	26	0	0	0	0
	33	0	0	0	0
	34	0	0	0	0
	35	0	0	0	0
PERIOD 2	11	1	0	0	0
	12	1	0	0	0
	13	1	0	0	0
	14	1	0	0	0
	15	0	0	0	0
	16	0	1	0	0
	17	0	0	0	0
	22	1	0	0	0
	23	0	1	0	0
	24	0	0	0	0
	25	0	0	0	0
	26	0	0	0	0
	33	0	0	0	0
	34	0	0	0	0
	35	0	0	0	0
PERIOD 3	11	1	0	0	0
	12	1	0	0	0
	13	1	0	0	0
	14	1	0	0	0
	15	0	0	1	0
	16	0	1	0	0
	17	0	0	0	1
	22	1	0	0	0
	23	0	1	0	0
	24	0	0	1	0
	25	0	0	1	0
	26	0	0	0	1
	33	0	0	1	0
	34	0	0	1	0
	35	0	0	0	1

	z_b^t	\vec{h}_1	\vec{h}_2	\vec{h}_3	\vec{h}_4
PERIOD 1	11	0	0	0	0
	12	0	0	0	0
	13	0	0	0	0
	14	0	0	0	0
	15	0	0	0	0
	16	0	0	0	0
	17	0	0	0	0
	22	0	0	0	0
	23	0	0	0	0
	24	0	0	0	0
	25	0	0	0	0
	26	0	0	0	0
	33	0	0	0	0
	34	0	0	0	0
	35	0	0	0	0
PERIOD 2	11	1	0	0	0
	12	1	0	0	0
	13	1	0	0	0
	14	1	0	0	0
	15	0	0	1	0
	16	0	1	0	0
	17	0	0	0	1
	22	1	0	0	0
	23	0	1	0	0
	24	0	0	1	0
	25	0	0	1	0
	26	0	0	0	1
	33	0	0	1	0
	34	0	0	1	0
	35	0	0	0	1
PERIOD 3	11	1	0	0	0
	12	1	0	0	0
	13	1	0	0	0
	14	1	0	0	0
	15	0	0	1	0
	16	0	1	0	0
	17	0	0	0	1
	22	1	0	0	0
	23	0	1	0	0
	24	0	0	1	0
	25	0	0	1	0
	26	0	0	0	1
	33	0	0	1	0
	34	0	0	1	0
	35	0	0	0	1

Figure 4.30: Iteration 2- Current state of the partitions after appending the integer feasible columns \vec{h} to the previous set of partitions \vec{v} . At this stage the columns \vec{h} are not orthogonal to \vec{v} yet

	z_b^t	$\vec{v}_1 \cap \vec{h}_1$	$\vec{v}_2 \cap \vec{h}_2$	$\vec{v}_3 \cap \vec{h}_3$	$\vec{v}_4 \cap \vec{h}_4$	$\vec{v}_1 / \Sigma \vec{I}$	$\vec{h}_3 / \Sigma \vec{v}$	$\vec{h}_4 / \Sigma \vec{v}$
PERIOD 1	11	0	0	0	0	1	0	0
	12	0	0	0	0	1	0	0
	13	0	0	0	0	1	0	0
	14	0	0	0	0	1	0	0
	15	0	0	0	0	0	0	0
	16	0	0	0	0	0	0	0
	17	0	0	0	0	0	0	0
	22	0	0	0	0	1	0	0
	23	0	0	0	0	0	0	0
	24	0	0	0	0	0	0	0
	25	0	0	0	0	0	0	0
	26	0	0	0	0	0	0	0
	33	0	0	0	0	0	0	0
	34	0	0	0	0	0	0	0
	35	0	0	0	0	0	0	0
PERIOD 2	11	1	0	0	0	0	0	0
	12	1	0	0	0	0	0	0
	13	1	0	0	0	0	0	0
	14	1	0	0	0	0	0	0
	15	0	0	0	0	0	1	0
	16	0	1	0	0	0	0	0
	17	0	0	0	0	0	0	1
	22	1	0	0	0	0	0	0
	23	0	1	0	0	0	0	0
	24	0	0	0	0	0	1	0
	25	0	0	0	0	0	1	0
	26	0	0	0	0	0	0	1
	33	0	0	0	0	0	1	0
	34	0	0	0	0	0	1	0
	35	0	0	0	0	0	0	1
PERIOD 3	11	1	0	0	0	0	0	0
	12	1	0	0	0	0	0	0
	13	1	0	0	0	0	0	0
	14	1	0	0	0	0	0	0
	15	0	0	1	0	0	0	0
	16	0	1	0	0	0	0	0
	17	0	0	0	1	0	0	0
	22	1	0	0	0	0	0	0
	23	0	1	0	0	0	0	0
	24	0	0	1	0	0	0	0
	25	0	0	1	0	0	0	0
	26	0	0	0	1	0	0	0
	33	0	0	1	0	0	0	0
	34	0	0	1	0	0	0	0
	35	0	0	0	1	0	0	0

Figure 4.31: Iteration 2- Integer feasible orthogonal partitions. Colors represent the blocks considered in the partition

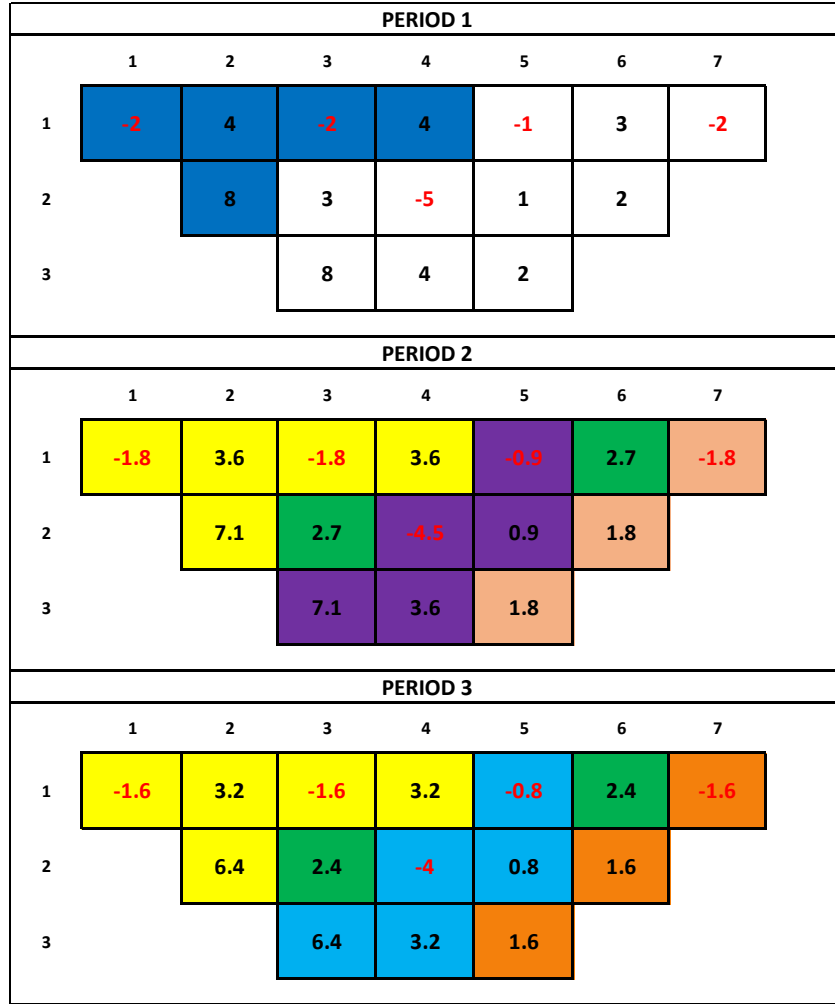


Figure 4.32: Iteration 2- Integer feasible orthogonal partitions presented as pits per period

The original variables z_b^t that appear in the orthogonal columns \vec{v}_1 to \vec{v}_7 will be replaced with the variables λ_1 to λ_7 as part of the contraction operation and the master problem will be formulated as follows:

$$\lambda_1 \rightarrow \vec{v}_1 = \{ z_{11}^2, z_{12}^2, z_{13}^2, z_{14}^2, z_{22}^2, z_{11}^3, z_{12}^3, z_{13}^3, z_{14}^3, z_{22}^3 \}$$

$$\lambda_2 \rightarrow \vec{v}_2 = \{ z_{16}^2, z_{23}^2, z_{16}^3, z_{23}^3 \}$$

$$\lambda_3 \rightarrow \vec{v}_3 = \{ z_{15}^3, z_{24}^3, z_{25}^3, z_{33}^3, z_{34}^3 \}$$

$$\lambda_4 \rightarrow \vec{v}_4 = \{ z_{17}^3, z_{26}^3, z_{35}^3 \}$$

$$\lambda_5 \rightarrow \vec{v}_5 = \{ z_{11}^1, z_{12}^1, z_{13}^1, z_{14}^1, z_{22}^1 \}$$

$$\lambda_6 \rightarrow \vec{v}_6 = \{ z_{15}^2, z_{24}^2, z_{25}^2, z_{33}^2, z_{34}^2 \}$$

$$\lambda_7 \rightarrow \vec{v}_7 = \{ z_{17}^2, z_{26}^2, z_{35}^2 \}$$

Master Problem Formulation:

Objective Function:

$$\begin{aligned} \text{Max } Z = & \lambda_5(c_{11}^1 - c_{11}^2) + \lambda_5(c_{12}^1 - c_{12}^2) + \lambda_5(c_{13}^1 - c_{13}^2) + \lambda_5(c_{14}^1 - c_{14}^2) \\ & + \lambda_5(c_{22}^1 - c_{22}^2) + \lambda_1(c_{11}^2 - c_{11}^3) + \lambda_1(c_{12}^2 - c_{12}^3) + \lambda_1(c_{13}^2 - c_{13}^3) \\ & + \lambda_1(c_{14}^2 - c_{14}^3) + \lambda_6(c_{15}^2 - c_{15}^3) + \lambda_2(c_{16}^2 - c_{16}^3) + \lambda_7(c_{17}^2 - c_{17}^3) \\ & + \lambda_1(c_{22}^2 - c_{22}^3) + \lambda_2(c_{23}^2 - c_{23}^3) + \lambda_6(c_{24}^2 - c_{24}^3) + \lambda_6(c_{25}^2 - c_{25}^3) \\ & + \lambda_7(c_{26}^2 - c_{26}^3) + \lambda_6(c_{33}^2 - c_{33}^3) + \lambda_6(c_{34}^2 - c_{34}^3) + \lambda_7(c_{35}^2 - c_{35}^3) + \lambda_1 c_{11}^3 \\ & + \lambda_1 c_{12}^3 + \lambda_1 c_{13}^3 + \lambda_1 c_{14}^3 + \lambda_3 c_{15}^3 + \lambda_2 c_{16}^3 + \lambda_4 c_{17}^3 + \lambda_1 c_{22}^3 + \lambda_2 c_{23}^3 \\ & + \lambda_3 c_{24}^3 + \lambda_3 c_{25}^3 + \lambda_4 c_{26}^3 + \lambda_3 c_{33}^3 + \lambda_3 c_{34}^3 + \lambda_4 c_{35}^3 \end{aligned} \quad (4.241)$$

Ensures Sequencing Across Time Periods:

$$\lambda_5 \leq \lambda_1 \quad (4.242)$$

$$\lambda_6 \leq \lambda_3 \quad (4.243)$$

$$\lambda_7 \leq \lambda_4 \quad (4.244)$$

Block Sequencing Constraints:

$$\lambda_2 \leq \lambda_1 \quad (4.245)$$

$$\lambda_6 \leq \lambda_2 \quad (4.246)$$

$$\lambda_7 \leq \lambda_6 \quad (4.247)$$

$$\lambda_3 \leq \lambda_2 \quad (4.248)$$

$$\lambda_4 \leq \lambda_3 \quad (4.249)$$

Capacity Constraints:

Process Capacity Constraints:

$$\lambda_5 + \lambda_5 + \lambda_5 \leq 3 \quad (4.250)$$

$$\begin{aligned} & (\lambda_1 - \lambda_5) + (\lambda_1 - \lambda_5) + \lambda_2 + (\lambda_1 - \lambda_5) + \lambda_2 + \lambda_6 \\ & + \lambda_7 + \lambda_6 + \lambda_6 + \lambda_7 \leq 3 \end{aligned} \quad (4.251)$$

$$\begin{aligned} & (\lambda_1 - \lambda_1) + (\lambda_1 - \lambda_1) + (\lambda_2 - \lambda_2) + (\lambda_1 - \lambda_1) + (\lambda_2 - \lambda_2) + (\lambda_3 - \lambda_6) \\ & + (\lambda_4 - \lambda_7) + (\lambda_3 - \lambda_6) + (\lambda_3 - \lambda_6) + (\lambda_4 - \lambda_7) \leq 3 \end{aligned} \quad (4.252)$$

Variable Restrictions:

$$0 \leq \lambda_i \leq 1 \quad \forall i \in \{1,2,3,4,5,6,7\} \quad (4.253)$$

The mathematical model is simplified as follows:

$$\text{Max } Z = 10.7\lambda_1 + 5.4\lambda_2 + 5.6\lambda_3 + 1.6\lambda_4 + 1.3\lambda_5 + 0.7\lambda_6 + 0.2\lambda_7 \quad (4.254)$$

Ensures Sequencing Across Time Periods:

$$\lambda_5 \leq \lambda_1 \quad (4.255)$$

$$\lambda_6 \leq \lambda_3 \quad (4.256)$$

$$\lambda_7 \leq \lambda_4 \quad (4.257)$$

Block Sequencing Constraints:

$$\lambda_2 \leq \lambda_1 \quad (4.258)$$

$$\lambda_6 \leq \lambda_2 \quad (4.259)$$

$$\lambda_7 \leq \lambda_6 \quad (4.260)$$

$$\lambda_3 \leq \lambda_2 \quad (4.261)$$

$$\lambda_4 \leq \lambda_3 \quad (4.262)$$

Capacity Constraints:

Process Capacity Constraints:

$$3\lambda_5 \leq 3 \quad (4.263)$$

$$3\lambda_1 - 3\lambda_5 + 2\lambda_2 + 3\lambda_6 + 2\lambda_7 \leq 3 \quad (4.264)$$

$$3\lambda_3 - 3\lambda_6 + 2\lambda_4 - 2\lambda_7 \leq 3 \quad (4.265)$$

Variable Restrictions:

$$0 \leq \lambda_i \leq 1 \quad \forall i \in \{1,2,3,4,5,6,7\} \quad (4.266)$$

By applying the simplex algorithm, the following solution is achieved:

$$\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 1, \lambda_4 = 0.5, \lambda_5 = 1, \lambda_6 = 0.33, \lambda_7 = 0 \text{ where } Z=24.03.$$

$$\text{Duals For Capacity Constraints} \rightarrow \mu_1^1 = 1.4, \quad \mu_1^2 = 1, \quad \mu_1^3 = 0.9$$

Iteration 3

After penalizing the original block values with the dual variables $\mu_1^1, \mu_1^2, \mu_1^3$, the subproblem is solved. The max closure which is essentially a mine plan generated by solving the subproblem is shown in Figure 4.33. With the modified block values, the most valuable plan determined by the subproblem is to only mine blocks in period 1 and period 2. Let \vec{C}_1 be the set of blocks mined in period 1 such as $\vec{C}_1 = \{x_{11}, x_{12}, x_{13}, x_{14}, x_{22}\}$. Let \vec{O}_1 be the set of ore blocks in \vec{C}_1 such as $\vec{O}_1 = \{x_{12}, x_{14}, x_{22}\}$. Since $|\vec{O}_1| = CAP^1$, the blocks in \vec{C}_1 are transferred to the integer feasible set \vec{h}_1 . Let the closure \vec{C}_2 holds the blocks mined in period 2 such as;

$\vec{C}_2 = \{x_{15}, x_{16}, x_{17}, x_{23}, x_{24}, x_{25}, x_{26}, x_{33}, x_{34}, x_{35}\}$ and the ore blocks in \vec{C}_2 are stored in ore set; $\vec{O}_2 = \{x_{16}, x_{23}, x_{26}, x_{33}, x_{34}, x_{35}\}$. The fact that $|\vec{O}_2| > CAP^2$ shows that the period 2 pit needs to be split into the integer feasible mine plans.

PERIOD 1							
	1	2	3	4	5	6	7
1	-2	2.6	-2	2.6	-1	1.6	-2
2		6.6	1.6	-5	-0.4	0.6	
3			6.6	2.6	0.6		

PERIOD 2							
	1	2	3	4	5	6	7
1	-1.8	2.6	-1.8	2.6	-0.9	1.7	-1.8
2		6.1	1.7	-4.5	-0.1	0.8	
3			6.1	2.6	0.8		

PERIOD 3							
	1	2	3	4	5	6	7
1	-1.6	2.3	-1.6	2.3	-0.8	1.5	-1.6
2		5.5	1.5	-4	-0.1	0.7	
3			5.5	2.3	0.7		

Figure 4.33: Iteration 3- Mine plan generated by solving the subproblem. Period 1 pit is shown with yellow color and period 2 pit is shown with blue color

Initially, the penalty parameter α_1^2 is calculated as 2.9 by taking the average of the ore values in set \vec{O}_2 . Once the ore blocks are penalized, the single time period sequencing problem is solved but no solution is achieved due to the uneconomic modified block values. The new penalty variable α_2^2 is selected as 0.75. Once the new max flow problem is solved, the mine plan shown in Figure 4.34 is obtained. The number of ore blocks mined is $|\overline{Ore}_1^2| = 3$ which meets with the capacity requirements; therefore, the mined blocks are transferred to the integer feasible partition \vec{h}_2 . The period 2 set \vec{C}_2 will be updated by extracting the new partition \vec{h}_2 such as $\vec{C}_3 = \vec{C}_2 \setminus \vec{h}_2$.

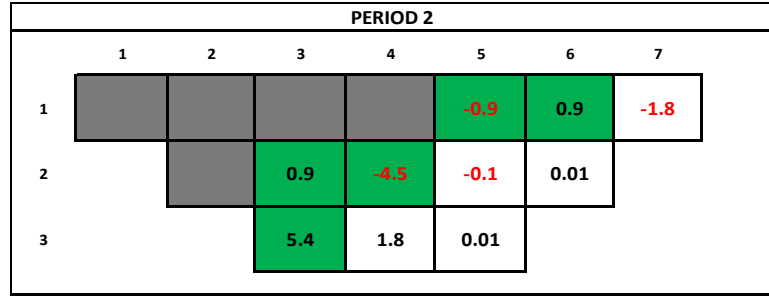


Figure 4.34: Integer feasible mine plan for period 2 with penalized values

The remaining blocks in the pit are $\vec{C}_3 = \{x_{17}, x_{25}, x_{26}, x_{34}, x_{35}\}$ as shown in Figure (4.35). The ore blocks in the remaining closure are $\vec{O}_3 = \{x_{26}, x_{34}, x_{35}\}$ where $|\vec{O}_3| = CAP^3$ which means the feasibility criteria is satisfied and the integerizing process is finalized by transferring the remaining blocks to the partitions \vec{h}_3 .

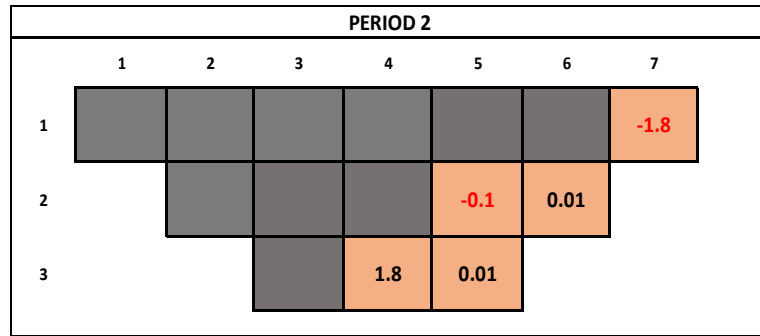


Figure 4.35: Integer feasible mine plan for period 2 with penalized values

Once the integerizing step is completed, the pit demonstrated in Figure 4.36 which is composed of integer feasible mine plans is achieved. The subproblem column together with its integer feasible partitions is shown in Figure 4.37 where the max closure \vec{s}_2 is decomposed into integer feasible sub-closures $\vec{h}_1, \vec{h}_2, \vec{h}_3$. The previous set of orthogonal partitions and the current integer feasible partitions are also presented together in Figure 4.38 to emphasize the non-orthogonal relation between the two set of partitions. Once these partitions are orthogonalized, the resulting set of columns are shown in Figure 4.39 where the blocks that exist in the same partition have the same color. Finally, the integer feasible orthogonal partitions are mapped into their physical locations shown in Figure 4.40.

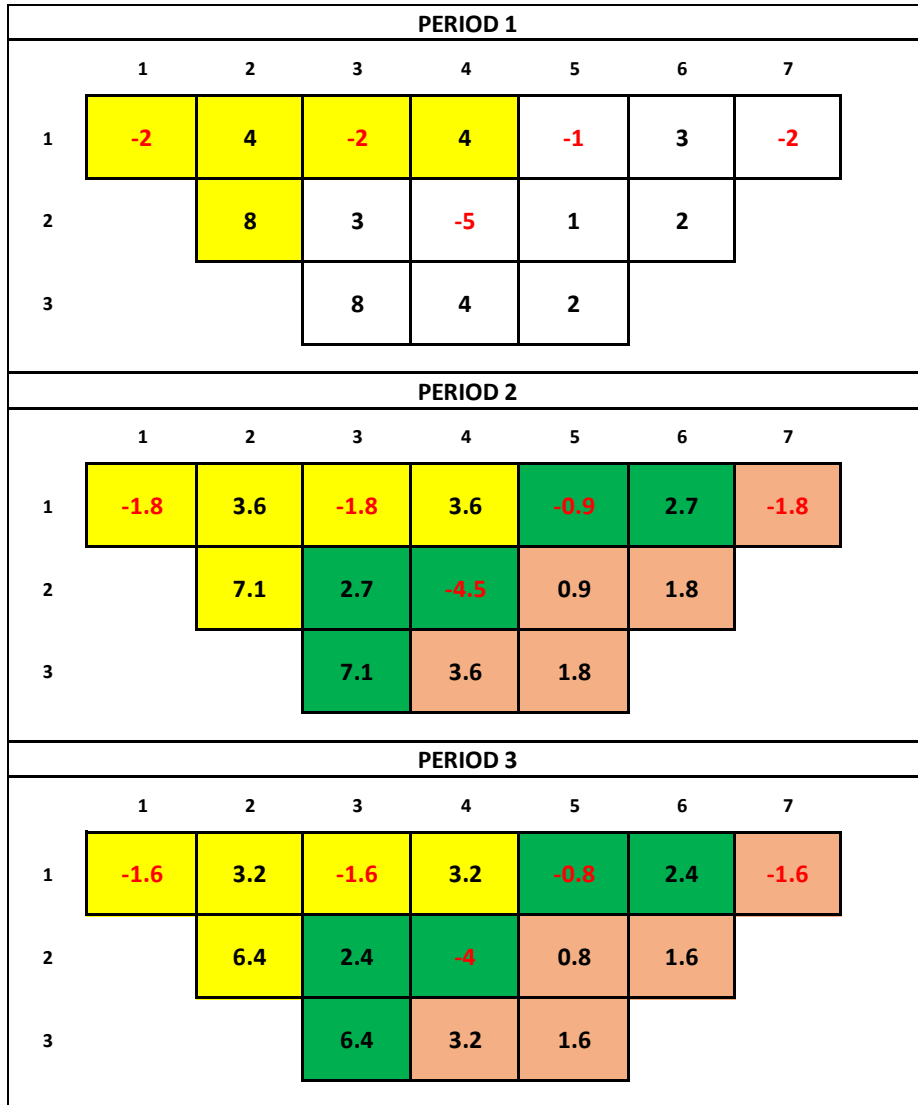


Figure 4.36: Iteration 2- Subproblem mine plan is split into integer feasible mine plans where each color represents a pit

	z_b^t	\vec{s}_2		z_b^t	\vec{h}_1	\vec{h}_2	\vec{h}_3
PERIOD 1	11	1	PERIOD 1	11	1	0	0
	12	1		12	1	0	0
	13	1		13	1	0	0
	14	1		14	1	0	0
	15	0		15	0	0	0
	16	0		16	0	0	0
	17	0		17	0	0	0
	22	1		22	1	0	0
	23	0		23	0	0	0
	24	0		24	0	0	0
	25	0		25	0	0	0
	26	0		26	0	0	0
	33	0		33	0	0	0
	34	0		34	0	0	0
	35	0		35	0	0	0
PERIOD 2	11	1	PERIOD 2	11	1	0	0
	12	1		12	1	0	0
	13	1		13	1	0	0
	14	1		14	1	0	0
	15	1		15	0	1	0
	16	1		16	0	1	0
	17	1		17	0	0	1
	22	1		22	1	0	0
	23	1		23	0	1	0
	24	1		24	0	1	0
	25	1		25	0	0	1
	26	1		26	0	0	1
	33	1		33	0	1	0
	34	1		34	0	0	1
	35	1		35	0	0	1
PERIOD 3	11	1	PERIOD 3	11	1	0	0
	12	1		12	1	0	0
	13	1		13	1	0	0
	14	1		14	1	0	0
	15	1		15	0	1	0
	16	1		16	0	1	0
	17	1		17	0	0	1
	22	1		22	1	0	0
	23	1		23	0	1	0
	24	1		24	0	1	0
	25	1		25	0	0	1
	26	1		26	0	0	1
	33	1		33	0	1	0
	34	1		34	0	0	1
	35	1		35	0	0	1

Figure 4.37: Iteration 3- Subproblem solution column on the left is split into integer feasible columns on the right

	z_b^t	\vec{v}								z_b^t	\vec{h}		
		v_1	v_2	v_3	v_4	v_5	v_6	v_7			h_1	h_2	h_3
PERIOD 1	11	0	0	0	0	1	0	0	PERIOD 1	11	1	0	0
	12	0	0	0	0	1	0	0		12	1	0	0
	13	0	0	0	0	1	0	0		13	1	0	0
	14	0	0	0	0	1	0	0		14	1	0	0
	15	0	0	0	0	0	0	0		15	0	0	0
	16	0	0	0	0	0	0	0		16	0	0	0
	17	0	0	0	0	0	0	0		17	0	0	0
	22	0	0	0	0	1	0	0		22	1	0	0
	23	0	0	0	0	0	0	0		23	0	0	0
	24	0	0	0	0	0	0	0		24	0	0	0
	25	0	0	0	0	0	0	0		25	0	0	0
	26	0	0	0	0	0	0	0		26	0	0	0
	33	0	0	0	0	0	0	0		33	0	0	0
	34	0	0	0	0	0	0	0		34	0	0	0
	35	0	0	0	0	0	0	0		35	0	0	0
PERIOD 2	11	1	0	0	0	0	0	0	PERIOD 2	11	1	0	0
	12	1	0	0	0	0	0	0		12	1	0	0
	13	1	0	0	0	0	0	0		13	1	0	0
	14	1	0	0	0	0	0	0		14	1	0	0
	15	0	0	0	0	0	1	0		15	0	1	0
	16	0	1	0	0	0	0	0		16	0	1	0
	17	0	0	0	0	0	0	1		17	0	0	1
	22	1	0	0	0	0	0	0		22	1	0	0
	23	0	1	0	0	0	0	0		23	0	1	0
	24	0	0	0	0	0	1	0		24	0	1	0
	25	0	0	0	0	0	1	0		25	0	0	1
	26	0	0	0	0	0	0	1		26	0	0	1
	33	0	0	0	0	0	1	0		33	0	1	0
	34	0	0	0	0	0	1	0		34	0	0	1
	35	0	0	0	0	0	0	1		35	0	0	1
PERIOD 3	11	1	0	0	0	0	0	0	PERIOD 3	11	1	0	0
	12	1	0	0	0	0	0	0		12	1	0	0
	13	1	0	0	0	0	0	0		13	1	0	0
	14	1	0	0	0	0	0	0		14	1	0	0
	15	0	0	1	0	0	0	0		15	0	1	0
	16	0	1	0	0	0	0	0		16	0	1	0
	17	0	0	0	1	0	0	0		17	0	0	1
	22	1	0	0	0	0	0	0		22	1	0	0
	23	0	1	0	0	0	0	0		23	0	1	0
	24	0	0	1	0	0	0	0		24	0	1	0
	25	0	0	1	0	0	0	0		25	0	0	1
	26	0	0	0	1	0	0	0		26	0	0	1
	33	0	0	1	0	0	0	0		33	0	1	0
	34	0	0	1	0	0	0	0		34	0	0	1
	35	0	0	0	1	0	0	0		35	0	0	1

Figure 4.38: Iteration 3- Current state of the partitions after appending the integer feasible columns \vec{h} to the previous set of partitions \vec{v} . At this stage the columns \vec{h} are not orthogonal to \vec{v} yet

	z_b^t	$\vec{v}_1 \cap \vec{h}_1$	$\vec{v}_5 \cap \vec{h}_1$	$\vec{v}_2 \cap \vec{h}_2$	$\vec{v}_3 \cap \vec{h}_2$	$\vec{v}_6 \cap \vec{h}_2$	$\vec{v}_3 \cap \vec{h}_3$	$\vec{v}_4 \cap \vec{h}_3$	$\vec{v}_6 \cap \vec{h}_3$	$\vec{v}_7 \cap \vec{h}_3$
PERIOD 1	11	0	1	0	0	0	0	0	0	0
	12	0	1	0	0	0	0	0	0	0
	13	0	1	0	0	0	0	0	0	0
	14	0	1	0	0	0	0	0	0	0
	15	0	0	0	0	0	0	0	0	0
	16	0	0	0	0	0	0	0	0	0
	17	0	0	0	0	0	0	0	0	0
	22	0	1	0	0	0	0	0	0	0
	23	0	0	0	0	0	0	0	0	0
	24	0	0	0	0	0	0	0	0	0
	25	0	0	0	0	0	0	0	0	0
	26	0	0	0	0	0	0	0	0	0
	33	0	0	0	0	0	0	0	0	0
	34	0	0	0	0	0	0	0	0	0
	35	0	0	0	0	0	0	0	0	0
PERIOD 2	11	1	0	0	0	0	0	0	0	0
	12	1	0	0	0	0	0	0	0	0
	13	1	0	0	0	0	0	0	0	0
	14	1	0	0	0	0	0	0	0	0
	15	0	0	0	0	1	0	0	0	0
	16	0	0	1	0	0	0	0	0	0
	17	0	0	0	0	0	0	0	0	1
	22	1	0	0	0	0	0	0	0	0
	23	0	0	1	0	0	0	0	0	0
	24	0	0	0	0	1	0	0	0	0
	25	0	0	0	0	0	0	0	1	0
	26	0	0	0	0	0	0	0	0	1
	33	0	0	0	0	1	0	0	0	0
	34	0	0	0	0	0	0	0	1	0
	35	0	0	0	0	0	0	0	0	1
PERIOD 3	11	1	0	0	0	0	0	0	0	0
	12	1	0	0	0	0	0	0	0	0
	13	1	0	0	0	0	0	0	0	0
	14	1	0	0	0	0	0	0	0	0
	15	0	0	0	1	0	0	0	0	0
	16	0	0	1	0	0	0	0	0	0
	17	0	0	0	0	0	0	1	0	0
	22	1	0	0	0	0	0	0	0	0
	23	0	0	1	0	0	0	0	0	0
	24	0	0	0	1	0	0	0	0	0
	25	0	0	0	0	0	1	0	0	0
	26	0	0	0	0	0	0	1	0	0
	33	0	0	0	1	0	0	0	0	0
	34	0	0	0	0	0	1	0	0	0
	35	0	0	0	0	0	0	1	0	0

Figure 4.39: Iteration 3- Integer feasible orthogonal partitions. Colors represent the blocks considered in the partition

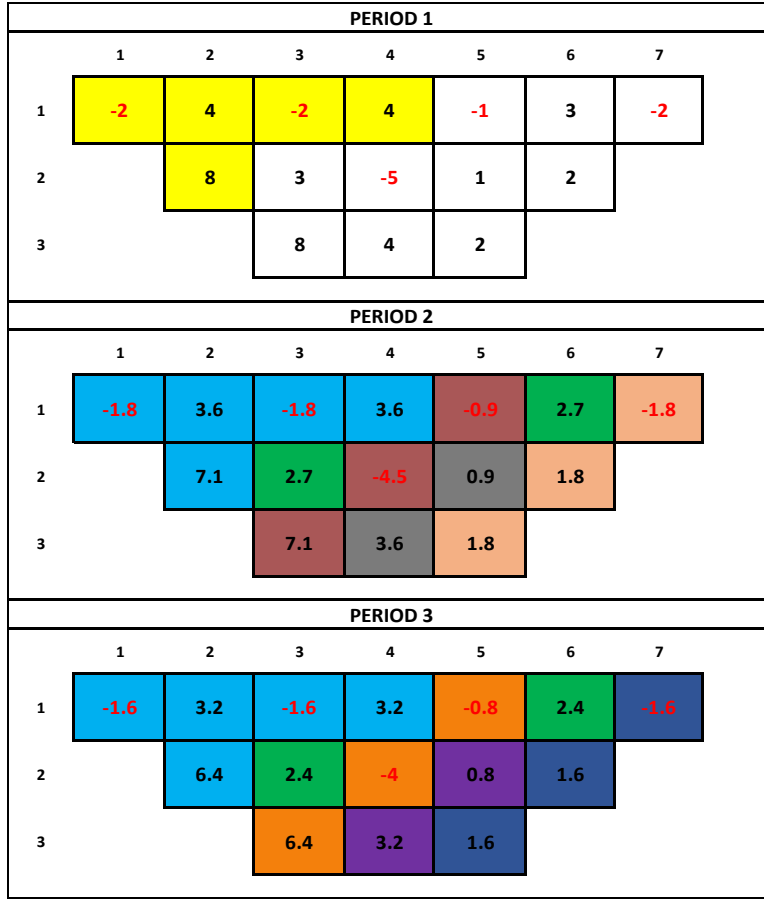


Figure 4.40: Iteration 3- Integer feasible orthogonal partitions presented as pits per period

Below is the master problem formulation written by direct substitution of the original variables z_b^t that appear in the orthogonal columns \vec{v}_1 to \vec{v}_9 with the variables λ_1 to λ_9 .

$$\lambda_1 \rightarrow \vec{v}_1 = \{z_{11}^2, z_{12}^2, z_{13}^2, z_{14}^2, z_{22}^2, z_{11}^3, z_{12}^3, z_{13}^3, z_{14}^3, z_{22}^3\}$$

$$\lambda_2 \rightarrow \vec{v}_2 = \{z_{11}^1, z_{12}^1, z_{13}^1, z_{14}^1, z_{22}^1\}$$

$$\lambda_3 \rightarrow \vec{v}_3 = \{z_{16}^2, z_{23}^2, z_{16}^3, z_{23}^3\}$$

$$\lambda_4 \rightarrow \vec{v}_4 = \{z_{15}^3, z_{24}^3, z_{33}^3\}$$

$$\lambda_5 \rightarrow \vec{v}_5 = \{z_{15}^2, z_{24}^2, z_{33}^2\}$$

$$\lambda_6 \rightarrow \vec{v}_6 = \{z_{25}^3, z_{34}^3\}$$

$$\lambda_7 \rightarrow \vec{v}_7 = \{z_{17}^3, z_{26}^3, z_{35}^3\}$$

$$\lambda_8 \rightarrow \vec{v}_8 = \{z_{25}^2, z_{34}^2\}$$

$$\lambda_9 \rightarrow \vec{v}_9 = \{z_{17}^2, z_{26}^2, z_{35}^2\}$$

Master Problem Formulation:

Objective Function:

$$\begin{aligned} \text{Max } Z = & \lambda_2(c_{11}^1 - c_{11}^2) + \lambda_2(c_{12}^1 - c_{12}^2) + \lambda_2(c_{13}^1 - c_{13}^2) + \lambda_2(c_{14}^1 - c_{14}^2) \\ & + \lambda_2(c_{22}^1 - c_{22}^2) + \lambda_1(c_{11}^2 - c_{11}^3) + \lambda_1(c_{12}^2 - c_{12}^3) + \lambda_1(c_{13}^2 - c_{13}^3) \\ & + \lambda_1(c_{14}^2 - c_{14}^3) + \lambda_5(c_{15}^2 - c_{15}^3) + \lambda_3(c_{16}^2 - c_{16}^3) + \lambda_9(c_{17}^2 - c_{17}^3) \\ & + \lambda_1(c_{22}^2 - c_{22}^3) + \lambda_3(c_{23}^2 - c_{23}^3) + \lambda_5(c_{24}^2 - c_{24}^3) + \lambda_8(c_{25}^2 - c_{25}^3) \\ & + \lambda_9(c_{26}^2 - c_{26}^3) + \lambda_5(c_{33}^2 - c_{33}^3) + \lambda_8(c_{34}^2 - c_{34}^3) + \lambda_9(c_{35}^2 - c_{35}^3) + \lambda_1 c_{11}^3 \\ & + \lambda_1 c_{12}^3 + \lambda_1 c_{13}^3 + \lambda_1 c_{14}^3 + \lambda_4 c_{15}^3 + \lambda_3 c_{16}^3 + \lambda_7 c_{17}^3 + \lambda_1 c_{22}^3 + \lambda_3 c_{23}^3 \\ & + \lambda_4 c_{24}^3 + \lambda_6 c_{25}^3 + \lambda_7 c_{26}^3 + \lambda_4 c_{33}^3 + \lambda_6 c_{34}^3 + \lambda_7 c_{35}^3 \end{aligned} \quad (4.267)$$

By Constraints Across Time Periods:

$$\lambda_2 \leq \lambda_1 \quad (4.268)$$

$$\lambda_6 \leq \lambda_3 \quad (4.269)$$

$$\lambda_5 \leq \lambda_4 \quad (4.270)$$

$$\lambda_8 \leq \lambda_6 \quad (4.271)$$

$$\lambda_9 \leq \lambda_7 \quad (4.272)$$

Block Sequencing Constraints:

$$\lambda_3 \leq \lambda_1 \quad (4.273)$$

$$\lambda_5 \leq \lambda_3 \quad (4.274)$$

$$\lambda_8 \leq \lambda_5 \quad (4.275)$$

$$\lambda_9 \leq \lambda_8 \quad (4.276)$$

$$\lambda_4 \leq \lambda_3 \quad (4.277)$$

$$\lambda_6 \leq \lambda_4 \quad (4.278)$$

$$\lambda_7 \leq \lambda_6 \quad (4.279)$$

Capacity Constraints:

Process Capacity Constraints:

$$\lambda_2 + \lambda_2 + \lambda_2 \leq 3 \quad (4.280)$$

$$\begin{aligned} & (\lambda_1 - \lambda_2) + (\lambda_1 - \lambda_2) + \lambda_3 + (\lambda_1 - \lambda_2) + \lambda_3 + \lambda_8 \\ & + \lambda_9 + \lambda_5 + \lambda_8 + \lambda_9 \leq 3 \end{aligned} \quad (4.281)$$

$$\begin{aligned} & (\lambda_1 - \lambda_1) + (\lambda_1 - \lambda_1) + (\lambda_3 - \lambda_3) + (\lambda_1 - \lambda_1) + (\lambda_3 - \lambda_3) + (\lambda_6 - \lambda_8) \\ & + (\lambda_7 - \lambda_9) + (\lambda_4 - \lambda_5) + (\lambda_6 - \lambda_8) + (\lambda_7 - \lambda_9) \leq 3 \end{aligned} \quad (4.282)$$

Variable Restrictions:

$$0 \leq \lambda_i \leq 1 \quad \forall i \in \{1,2,3,4,5,6,7,8,9\} \quad (4.283)$$

The simplified mathematical model for the master problem is in the following:

$$\text{Max } Z = 10.7\lambda_1 + 1.3\lambda_2 + 5.4\lambda_3 + 1.6\lambda_4 + 0.2\lambda_5 + 4\lambda_6 + 1.6\lambda_7 + 0.5\lambda_8 + 0.2\lambda_9 \quad (4.284)$$

By Constraints Across Time Periods:

$$\lambda_2 \leq \lambda_1 \quad (4.285)$$

$$\lambda_6 \leq \lambda_3 \quad (4.286)$$

$$\lambda_5 \leq \lambda_4 \quad (4.287)$$

$$\lambda_8 \leq \lambda_6 \quad (4.288)$$

$$\lambda_9 \leq \lambda_7 \quad (4.289)$$

Block Sequencing Constraints:

$$\lambda_3 \leq \lambda_1 \quad (4.290)$$

$$\lambda_5 \leq \lambda_3 \quad (4.291)$$

$$\lambda_8 \leq \lambda_5 \quad (4.292)$$

$$\lambda_9 \leq \lambda_8 \quad (4.293)$$

$$\lambda_4 \leq \lambda_3 \quad (4.294)$$

$$\lambda_6 \leq \lambda_4 \quad (4.295)$$

$$\lambda_7 \leq \lambda_6 \quad (4.296)$$

Capacity Constraints:**Process Capacity Constraints:**

$$\lambda_2 + \lambda_2 + \lambda_2 \leq 3 \quad (4.297)$$

$$(\lambda_1 - \lambda_2) + (\lambda_1 - \lambda_2) + \lambda_3 + (\lambda_1 - \lambda_2) + \lambda_3 + \lambda_8 + \lambda_9 + \lambda_5 + \lambda_8 + \lambda_9 \leq 3 \quad (4.298)$$

$$(\lambda_1 - \lambda_1) + (\lambda_1 - \lambda_1) + (\lambda_3 - \lambda_3) + (\lambda_1 - \lambda_1) + (\lambda_3 - \lambda_3) + (\lambda_6 - \lambda_8) + (\lambda_7 - \lambda_9) + (\lambda_4 - \lambda_5) + (\lambda_6 - \lambda_8) + (\lambda_7 - \lambda_9) \leq 3 \quad (4.299)$$

Variable Restrictions:

$$0 \leq \lambda_i \leq 1 \quad \forall i \in \{1,2,3,4,5,6,7,8,9\} \quad (4.300)$$

The solution to the master problem is:

$$\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 1, \lambda_4 = 1, \lambda_5 = 0.33, \lambda_6 = 1, \lambda_7 = 0.5, \lambda_8 = 0.33, \lambda_9 = 0$$

where Z=24.03.

Duals For Capacity Constraints $\rightarrow \mu_1^1 = 1.4, \mu_1^2 = 1, \mu_1^3 = 0.9$

The dual values generated in consecutive iterations are equal, henceforth we can confirm that the optimal LP solution is achieved. Since the sub-problem max closures are broken into integer feasible sub-closures at each iteration, once the optimal LP solution is achieved, it can be concluded that we have a set of integer feasible partitions which are $\{\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{v}_4, \vec{v}_5, \vec{v}_6, \vec{v}_7, \vec{v}_8, \vec{v}_9\}$ that span the optimal LP solution. In the next step, the optimal integer solution that exists within the solution space of these partitions will be identified. This will be done simply by replacing the continuous variable $\lambda_i \in [0,1]$ with the binary variable $\lambda_i \in \{0,1\}$ and re-solving the master problem of the last iteration. The optimal integer solution of the master problem is:

$\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 1, \lambda_4 = 1, \lambda_5 = 1, \lambda_6 = 1, \lambda_7 = 0, \lambda_8 = 0, \lambda_9 = 0$ where $Z=23.13$.

Since each λ variable represents a set of original problem variables, the value λ variable will be also the value of the associated original problem variables. Therefore, if the λ variables are converted back to the original mining variables, the following set which represents the period at which the blocks are mined will be generated:

Period 1 = $\{x_{11}^1, x_{12}^1, x_{13}^1, x_{14}^1, x_{22}^1\}$

Period 2 = $\{x_{16}^2, x_{23}^2, x_{33}^2\}$

Period 3 = $\{x_{15}^3, x_{24}^3, x_{25}^3, x_{34}^3\}$

To verify the results, the true IP solution to the original problem formulation was also determined and matched exactly the solution obtained with the new integer solution algorithm presented in this thesis. As we know the optimal LP and optimal IP solutions, the true optimality gap is 3.75%. The physical locations of the optimal schedules are shown in Figure 4.41.

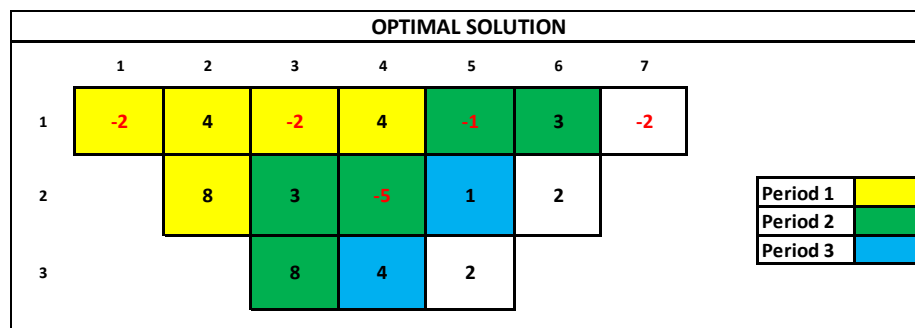


Figure 4.41: Schedules of the optimal mine plan

CHAPTER 5.

IMPLEMENTATION OF THE NEW INTEGER SOLUTION ALGORITHM TO THE LARGE-SCALE OPEN PIT MINING PROBLEMS

In this chapter, case studies will be presented to illustrate the implementation of the new integer solution algorithm on large scale open pit mining problems. Some of these mining problems may allow multiple processing options for a given block where the destination selection will be a function of a dynamic cutoff grade optimization process defined by the state of the system under capacity, average grade blending and risk blending constraints. Some of the mining problems may have multiple sources to feed the blocks into the process stream such as multiple pits and stockpiles. Again, it is important to emphasize that presently there is no known algorithm, either commercially available or presented in the literature, that can provide an optimal integer solution to the open pit mine production scheduling problem with capacity constraints together with lower and upper bound blending constraints. The strength of the integer solution algorithm developed in this thesis will be highlighted on the ability of solving problems that have more than 7 million variables as an integer problem with an optimality gap as small as 0.01 % within 5 hours 30 minutes.

5.1 Case Study 1 (McLaughlin Deposit)

The first case study will demonstrate the implementation of the new integer solution algorithm on scheduling a data set referred to as the McLaughlin Deposit for 10 years. The schematic description of the assumed mining complex is given in Figure 5.1. The economic parameters that will be used to derive the block values are given in Table 5.1. The blocks will be initially subjected to a break even cutoff grade 0.03 oz/t which will separate the invaluable waste blocks from the ore blocks which carry a recoverable value once processed. Any ore block that has a grade less than 0.05 oz/t will be treated at the leach pads or sent to the waste dump once it is mined. Moreover, an ore block with a grade greater than or equal to 0.08 oz/t will be processed at the mill or sent to the waste dump. If the grade of a mined ore block is between 0.05oz/t and

0.08oz/t, it will be either processed at the mill, treated at the leach pads or sent to the waste dump. The ore blocks in these grade intervals are categorized as “undecided blocks” which means that the process destination is not designated based on a cutoff grade; instead the optimum destination will be determined based on the state of the system. Table 5.2 presents the cutoff grade intervals adopted for this particular example. Furthermore, the proportions of the blocks in the process stream that belong to different risk categories such as inferred, indicated and measured will be controlled by enforcing risk constraints which are in a form of blending constraints.

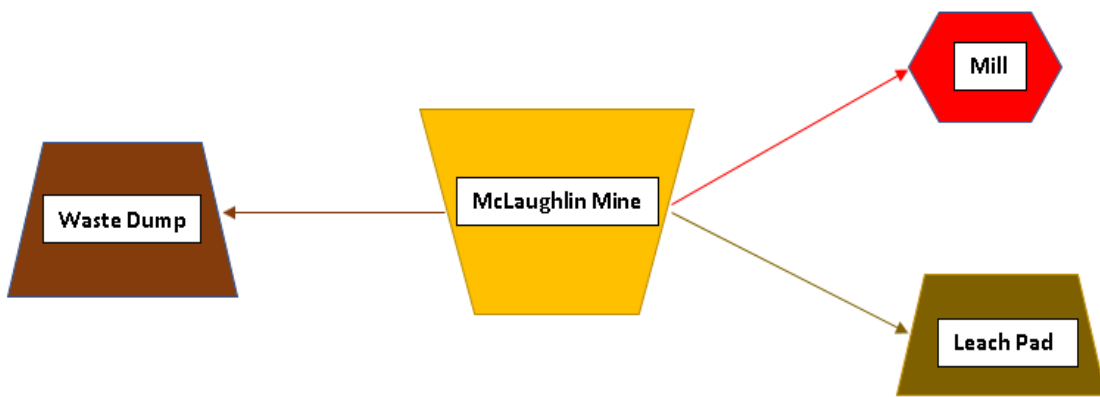


Figure 5.1: McLaughlin Mine process flow

Table 5.1: Economic parameters

PARAMETERS	VALUE
AU	1250 \$/oz
MILL COST	12 \$/t
LEACH COST	6 \$/t
MILL RECOVERY	90%
LEACH RECOVERY	70%
DISCOUNT RATE	12.5%

Table 5.2: Summary of the cutoff grade intervals

CUTOFF	GRADES
BREAKEVEN	$\geq 0.03\text{oz/t}$
LEACH	$< 0.05\text{oz/t}$
UNDECIDED	$\geq 0.05\text{oz/t AND } < 0.08\text{oz/t}$
MILL	$\geq 0.08\text{oz/t}$

The deposit characteristics are illustrated in Table 5.3. It is clear that out of 1.9 million blocks in the block model, only 87 thousand blocks are qualified to be ore blocks with an average grade of 0.061 oz/t. Among the ore blocks, almost half the blocks are leach blocks with an average grade of 0.034 oz/t, and on the remaining half there are 18.1 thousand mill blocks with an average grade of 0.133 oz/t, and 24.4 thousand undecided blocks with an average grade of 0.057 oz/t. Also, Table 5.4 shows that 19.8% of the blocks are categorized as inferred, 36% as indicated and 44.2% as measured.

Table 5.3: McLaughlin deposit characteristics

MATERIAL TYPE	BLOCKS	TONNAGE	AVG GRADE
TOTAL	1,922,749	961,374,500	-
WASTE	1,835,583	917,791,500	-
ORE	87,166	43,583,000	0.061 oz/t
MILL	18,144	9,072,000	0.133 oz/t
LEACH	44,634	22,317,000	0.034 oz/t
UNDECIDED	24,388	12,194,000	0.057 oz/t

Table 5.4: Risk profile of the deposit

RISK CATEGORY	PROPORTION
INFERRED	19.8%
INDICATED	36%
MEASURED	44.2%

5.1.1 The Ultimate Pit

Once the block model characteristics are outlined, the next step is to determine the ultimate pit for the McLaughlin Deposit. The mathematical model for the ultimate pit problem which is essentially a single period version of the subproblem shown in Chapter 4 requires a single value for a mining decision variable; which means that the destination of a block will be pre-determined. Hence, given a set of possible destinations for a single block, the destination where the highest recoverable value can be achieved will be selected. Moreover, the cone pattern generation technique is implemented to create arcs between the blocks to accomplish a uniform 45° slope angle. Then the solution to the ultimate pit problem is determined by implementing the pseudoflow algorithm. Table 5.5 illustrates the number of blocks and tonnage mined in the ultimate pit. It is clear that while the block model consists of 1.9 million blocks, the ultimate pit has only 245.6 thousand blocks. There are about 2 thousand leach blocks that existed in the block model but not mined in the ultimate pit since they are not economical. We can also say that leaving those leach blocks on the ground lead to a decrease of the proportion of the inferred blocks about 1.4 % as shown in Table 5.6. Also, there is no observable change on the average grades mined in the ultimate pit. The value of the pit which is calculated with the undiscounted block values determined by picking the most valuable destination was found to be \$2.2 billion.

Table 5.5: Summary of blocks in the ultimate pit

MATERIAL TYPE	BLOCKS	TONNAGE	AVG GRADE
TOTAL	245,617	122,808,500	-
WASTE	160,445	80,222,500	-
ORE	85,172	42,586,000	0.062 oz/t
MILL	18,140	9,070,000	0.133 oz/t
LEACH	42,737	21,368,500	0.034 oz/t
UNDECIDED	24,295	12,147,500	0.057 oz/t

Table 5.6: Ultimate pit risk profile

RISK CATEGORY	PROPORTION
INFERRED	18.4%
INDICATED	36.7%
MEASURED	44.9%

The results are integrated to MineSight 3D in order to visualize the blocks in the ultimate pit. Figure 5.2 shows a cross section from East-11075 location. The Figure 5.3 demonstrates the plan view together with the potential destinations and Figure 5.4 shows a cross section from North-9800 location. Traditionally, the ultimate pit limit is known as the most economical pit and treated as the final shape of the pit. A question remains, are the ultimate pit limits truly achievable given a particular mining system and related constraints. The comparison of the actual pit limits established after scheduling and the ultimate pit limit will be also given later in the section.

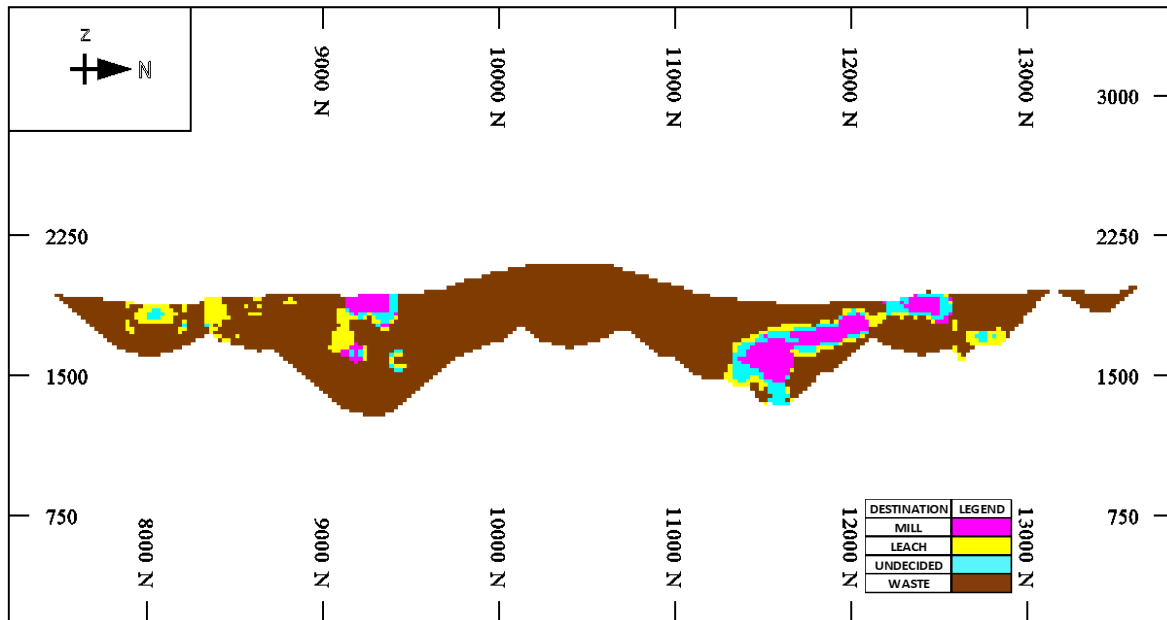


Figure 5.2: East-11075 cross section taken from the ultimate pit

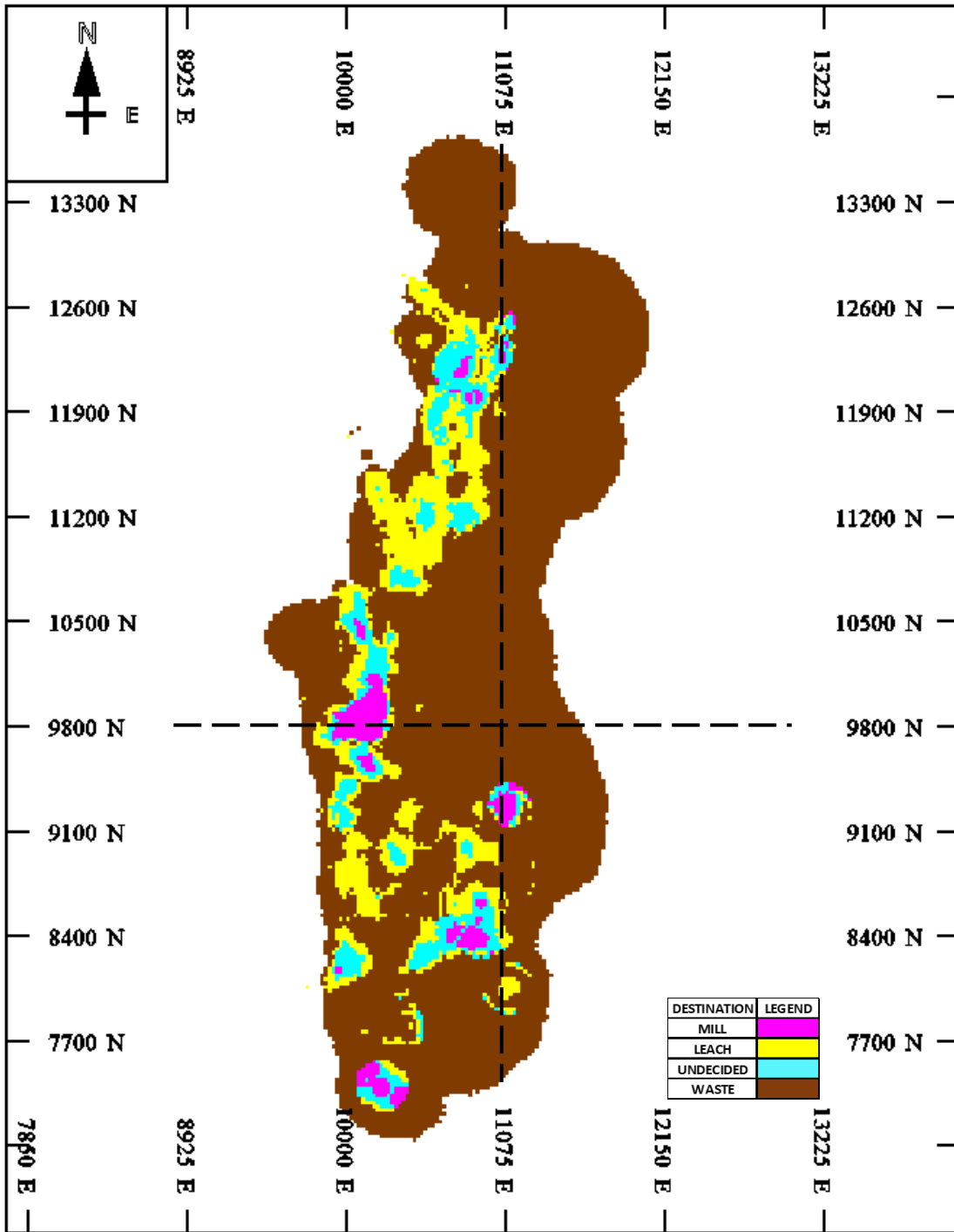


Figure 5.3: Plan view of the ultimate pit

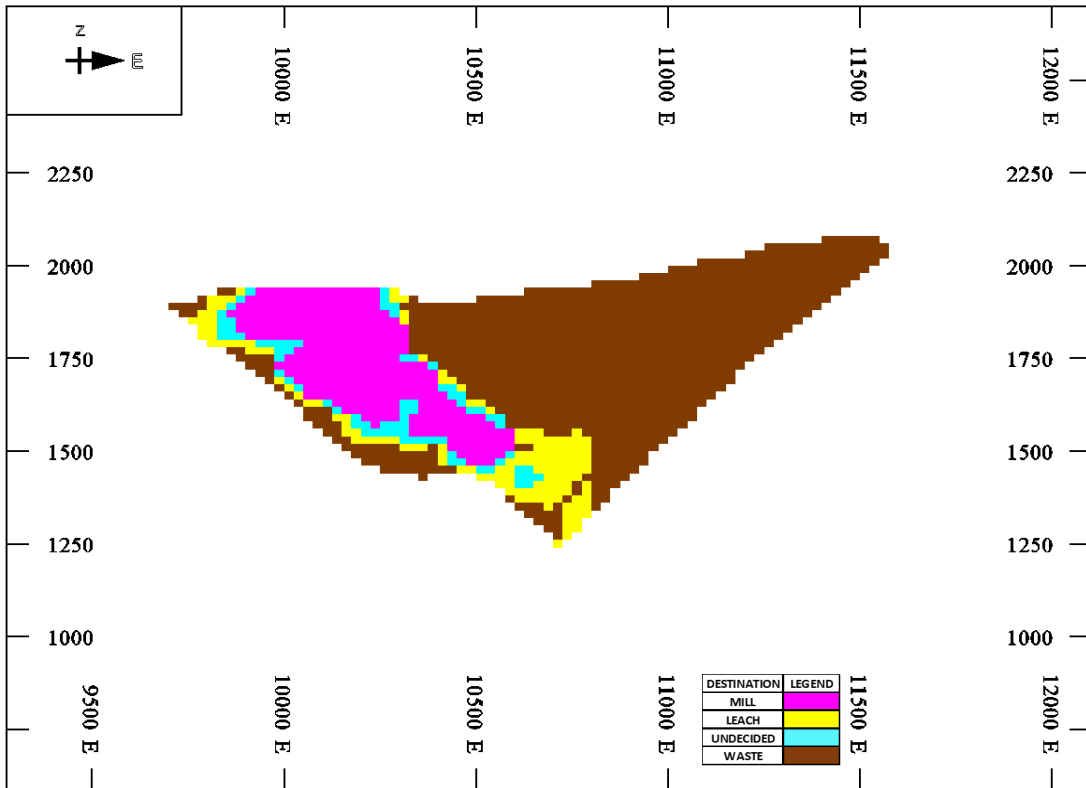


Figure 5.4: North-9800 cross section taken from the ultimate pit

5.1.2 Mine Production Schedule

The mine production schedule will be generated to mine the blocks in the ultimate pit under the guidance of production requirements. The life of the mine is 10 years. The mine plan must comply with the yearly production requirements outlined in Table 5.7. The requirements include restrictions on the maximum tonnage that can be processed at mill and leach pads, minimum average grade required by mill and leach pads, restrictions on the maximum proportion of the ore blocks that belongs to an inferred category which possess high risk and minimum requirements on the proportion of the ore blocks that belongs to indicated and measured risk categories. Risk proportions basically quantify the risk exposure of the ore blocks in the process flow.

Table 5.7: Production requirements for the mine plan

PERIODS	PROCESS CAPACITY (TONNAGE)		AVG GRADE (oz/t)		RISK PROPORTIONS		
	MILL	LEACH	>= MILL	>= LEACH	<= INF	>= IND	>= MEA
1	1,500,000	1,500,000	0.062	0.035	20%	10%	35%
2	1,750,000	1,750,000	0.062	0.035	20%	10%	35%
3	2,000,000	2,000,000	0.062	0.035	20%	10%	35%
4	2,750,000	2,750,000	0.062	0.035	40%	10%	25%
5	3,000,000	3,000,000	0.062	0.035	40%	10%	25%
6	3,000,000	3,000,000	0.062	0.035	40%	10%	25%
7	2,750,000	2,750,000	0.062	0.035	50%	10%	25%
8	2,000,000	2,000,000	0.062	0.035	50%	10%	25%
9	1,750,000	1,750,000	0.062	0.035	50%	10%	25%
10	1,500,000	1,500,000	0.062	0.035	50%	10%	25%

The optimal mine plan is generated by implementing the new integer solution algorithm. The results are shown in Table 5.8. It is clear that all of the yearly requirements are honored. In Figure 5.5 it can be seen that the mill is working at full capacity for the first 7 years and after that there is a shortage of mill blocks to fill the mill capacity. On the other hand, the Figure 5.6 shows that the resulting mine plan is able to fill leach pad capacity every year. Also, the red dotted line that appears in both figures shows the average grade of the ore blocks processed every year. It is obvious that the optimizer prioritizes the high-grade zones in the earlier years of the production in order to prevent the loss in value occurring naturally by the discount factor. Hence, both mill and leach average grades are significantly higher in the earlier periods and gradually decrease until they become equal to the minimum yearly average grade required by the process destination. The Figure 5.7 shows the risk behavior of the mine plan over the years. It is apparent that the earlier years of the production, the optimizer mines from the less riskier areas. This is indeed true in practice where the approved business plan must deliver the amount of ounces promised to the shareholders, therefore the confidence level on the processed ore tonnage plays a key role. The riskier areas are postponed to the later years of the production since the confidence level on the riskier areas can be always increased by adding more drill holes later on.

As it was mentioned before the new integer solution algorithm also provides the theoretical upper bound on a given solution. The theoretical upper bound is calculated by solving the mining

problem as a LP problem and then the quality of the integer solution can be measured from the optimality gap. In this case the LP optimal solution is found as \$1,581,250,000 and the integer solution is found as \$1,581,085,000; as shown in Table 5.9. The optimality gap is 0.01% and it took 5 hours 30 minutes to achieve an integer solution. Such a small optimality gap has never been reported in the literature on a problem of this size (~ 7.3 million variables) with capacity and blending constraints along with multi destinations.

Table 5.8: Summary of the results for the generated mine plan

PERIODS	PROCESSED TONNAGE		AVG GRADE (oz/t)		RISK PROPORTIONS		
	MILL	LEACH	MILL	LEACH	INF	IND	MEA
1	1,500,000	1,500,000	0.196	0.054	7%	27%	66%
2	1,750,000	1,750,000	0.149	0.051	6%	39%	55%
3	2,000,000	2,000,000	0.124	0.044	9%	40%	52%
4	2,750,000	2,750,000	0.089	0.035	13%	38%	49%
5	2,999,500	2,999,000	0.075	0.035	11%	40%	49%
6	3,000,000	3,000,000	0.063	0.035	19%	37%	45%
7	2,750,000	2,750,000	0.062	0.035	25%	35%	40%
8	522,000	2,000,000	0.062	0.035	30%	38%	32%
9	26,000	1,750,000	0.062	0.035	27%	43%	31%
10	63,000	1,500,000	0.074	0.035	41%	34%	25%

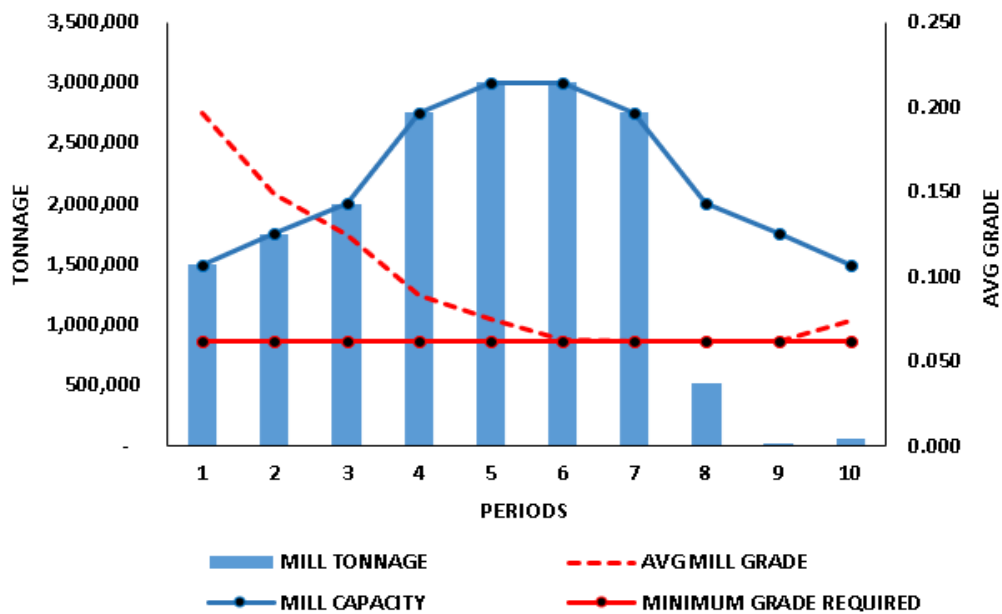


Figure 5.5: Yearly processed tonnage and the average grade at the mill

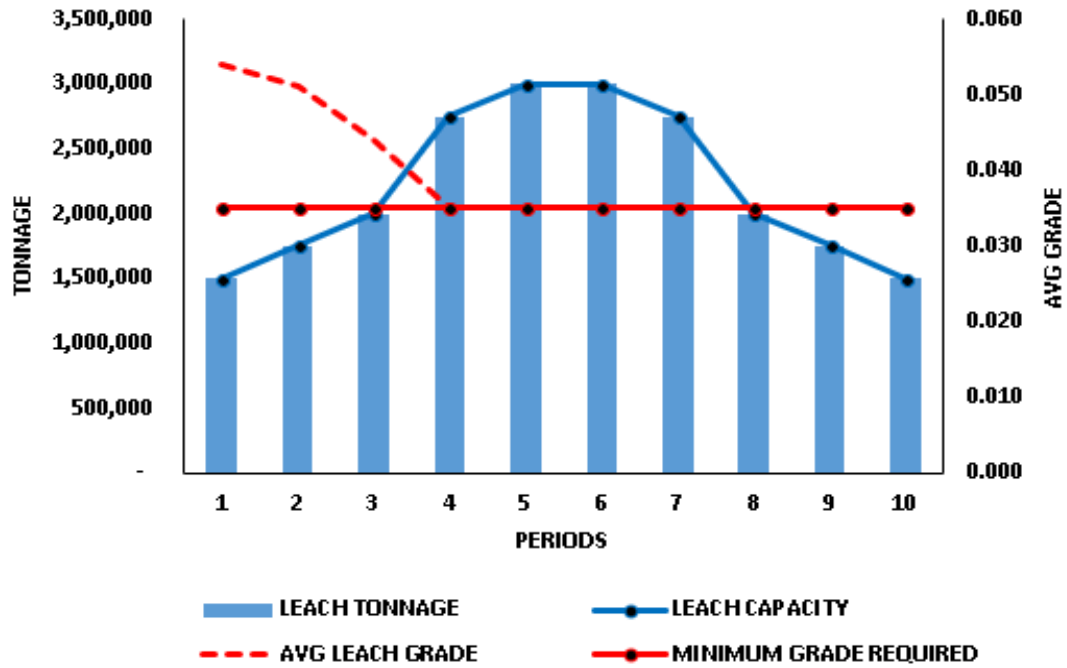


Figure 5.6: Yearly processed tonnage and the average grade at the leach pads

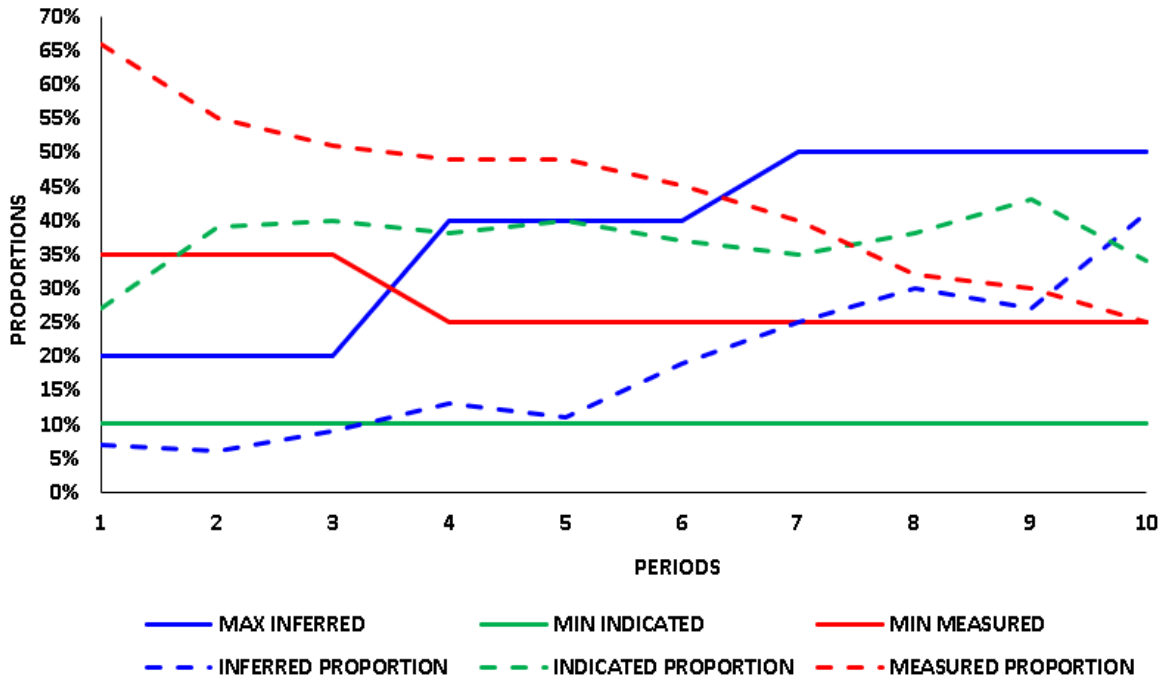


Figure 5.7: Risk behavior of the mine plan over the production years

Table 5.9: Summary of the results

LP NPV @ 12.5 %	\$1,581,250,000
IP NPV @ 12.5 %	\$1,581,085,000
OPTIMALITY GAP %	0.01%
SOLUTION TIME	5 h 30 min

The yearly schedules are presented on a plan view in Figure 5.9, north-south and east west cross sections are shown in Figure 5.8 and Figure 5.10 respectively. Once the pit outline is established based on the yearly production schedules; it is worth making a comparison with the pit outline generated by the traditional ultimate pit as shown in Figure 5.11 and Figure 5.12. First of all, it can be clearly seen in Figure 5.11 that the undecided blocks colored with blue on the traditional ultimate pit plan view received yellow or pink colors on the schedules which shows that the ore blocks are not blindly designated to the mill just because there is more recoverable value; instead the state of the system determined the best destination for each block. Secondly, the ultimate pit is essentially an unconstrained max closure where the mine plan is a constrained max closure. Hence, the true ultimate pit can be only determined once the production schedule is generated.

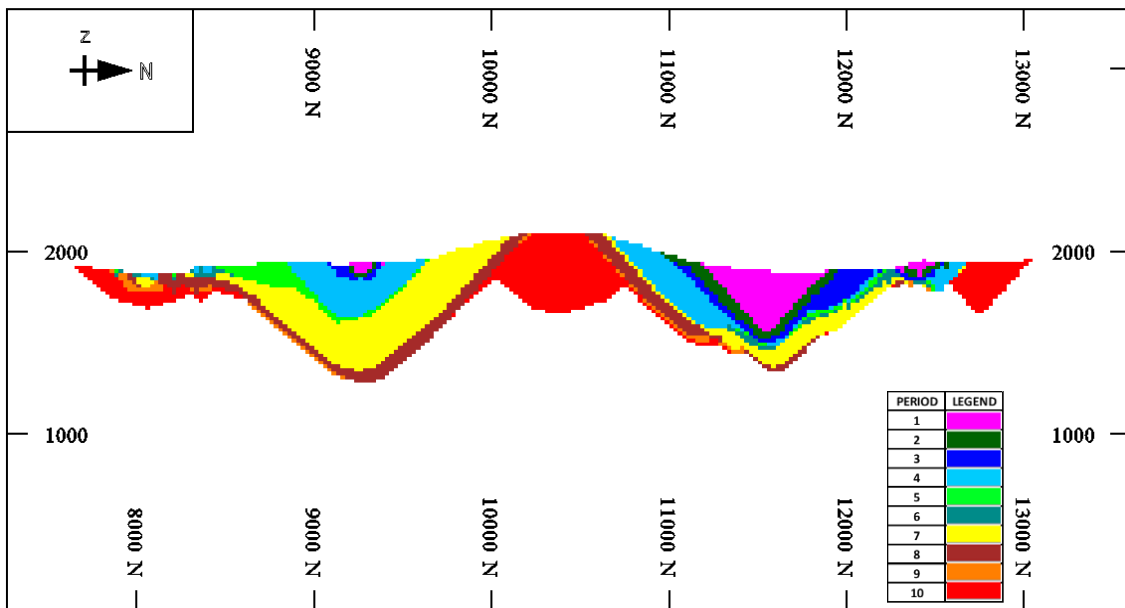


Figure 5.8: Yearly production schedules on East-11075 cross section

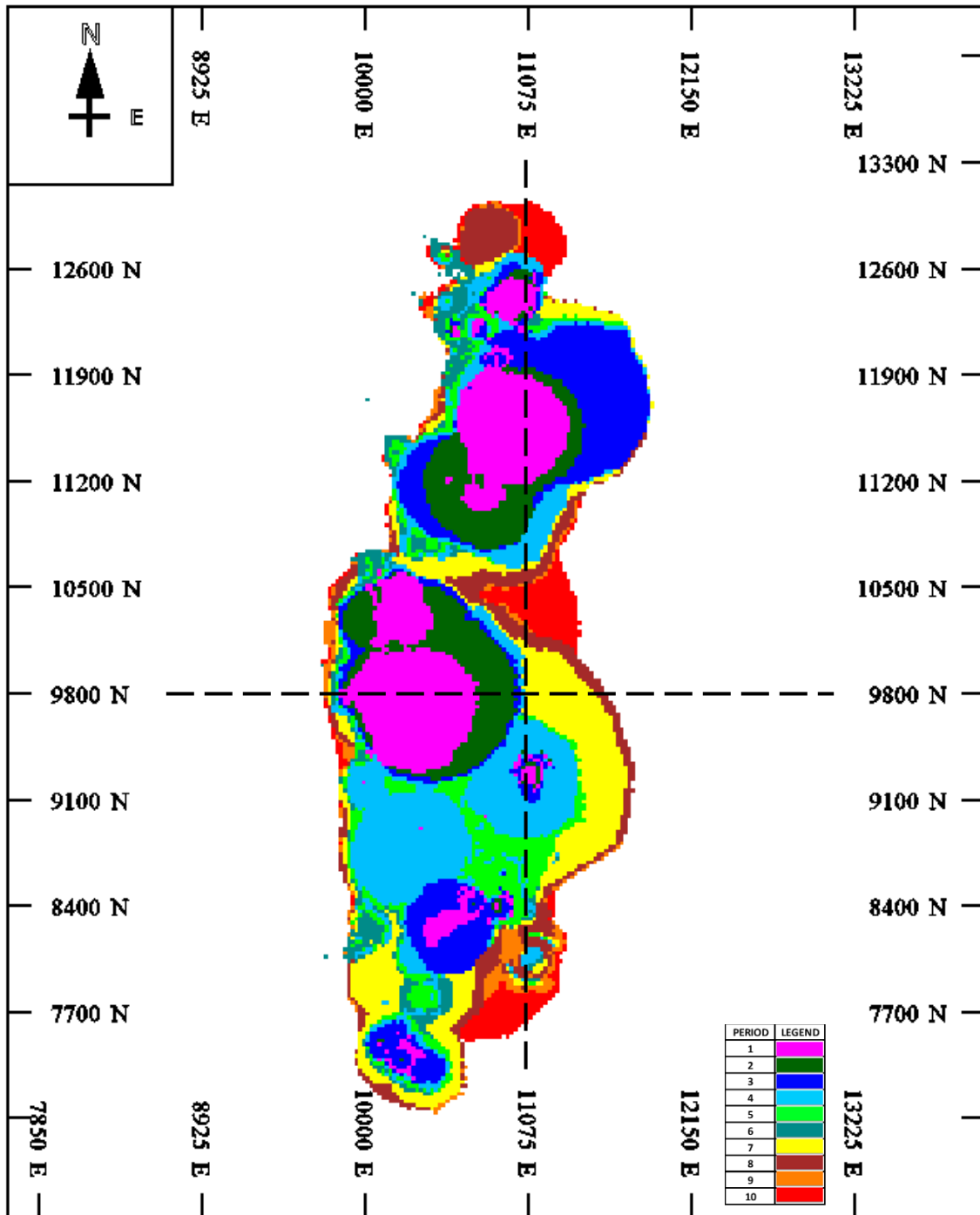


Figure 5.9: Plan view of the yearly production schedules

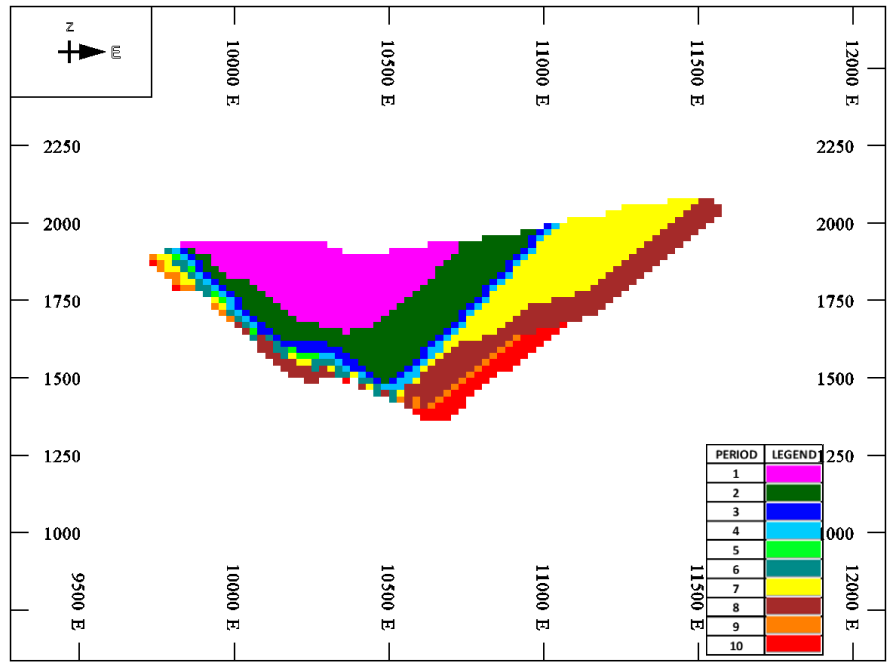


Figure 5.10: Yearly production schedules on North- 9800 cross section

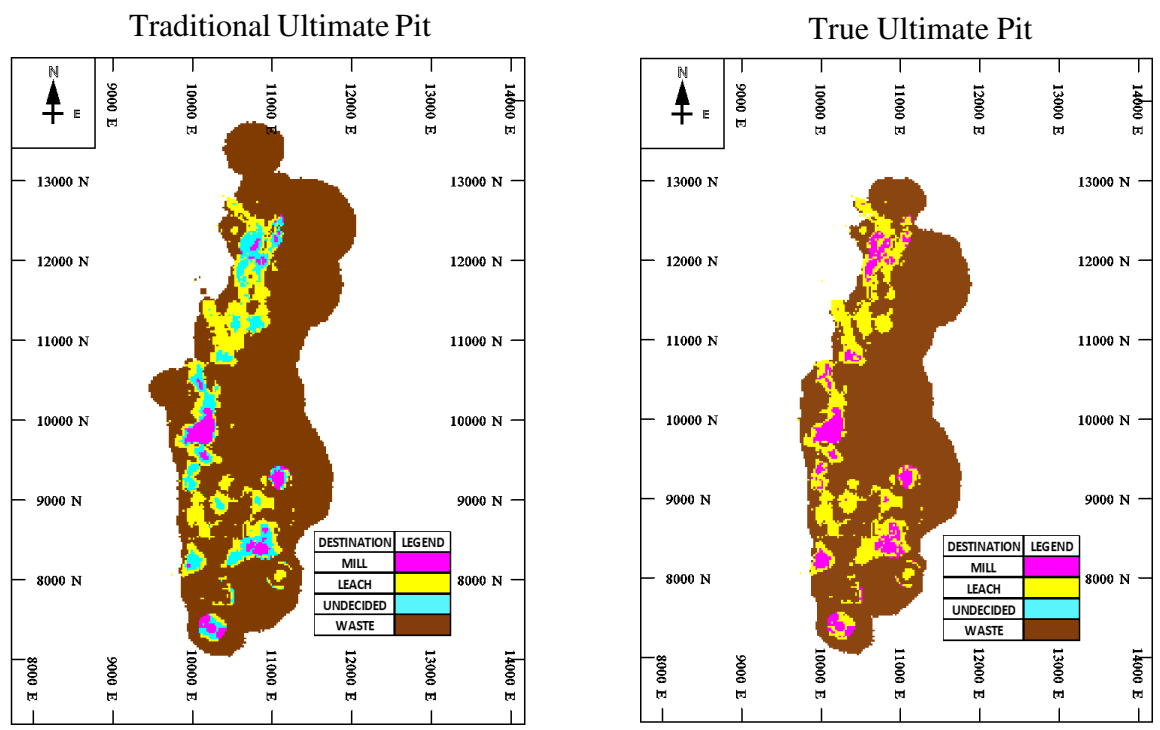


Figure 5.11: Comparison of the traditional ultimate pit vs true ultimate pit on a Plan view

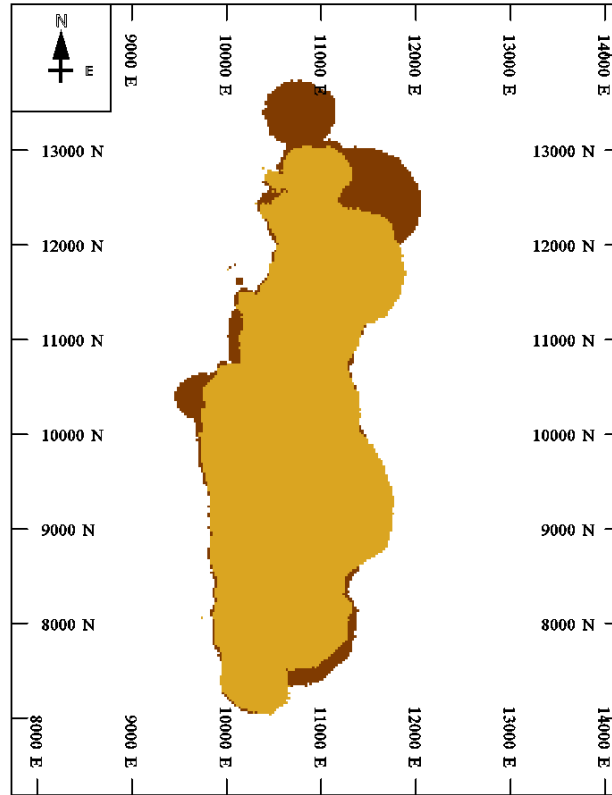


Figure 5.12: Projection of true ultimate pit on traditional ultimate pit

5.2 Case Study 2 (Gold Deposit)

The second case study will demonstrate the implementation of the new integer solution algorithm to schedule the blocks mined from multiple pits and blend them with the blocks pulled out from stockpiles. Figure 5.13 shows the schematic description of the mining complex. As there are 4 stockpiles, they will be assumed to have initial capacities and there will not be any blocks going into the stockpiles. Any ore block sent to a process destination must first pass through the crusher. Henceforth, the crusher capacity is the bottleneck of the system. There will be further capacity restrictions enforced by the mill and also total mining capacity based on the equipment availability.

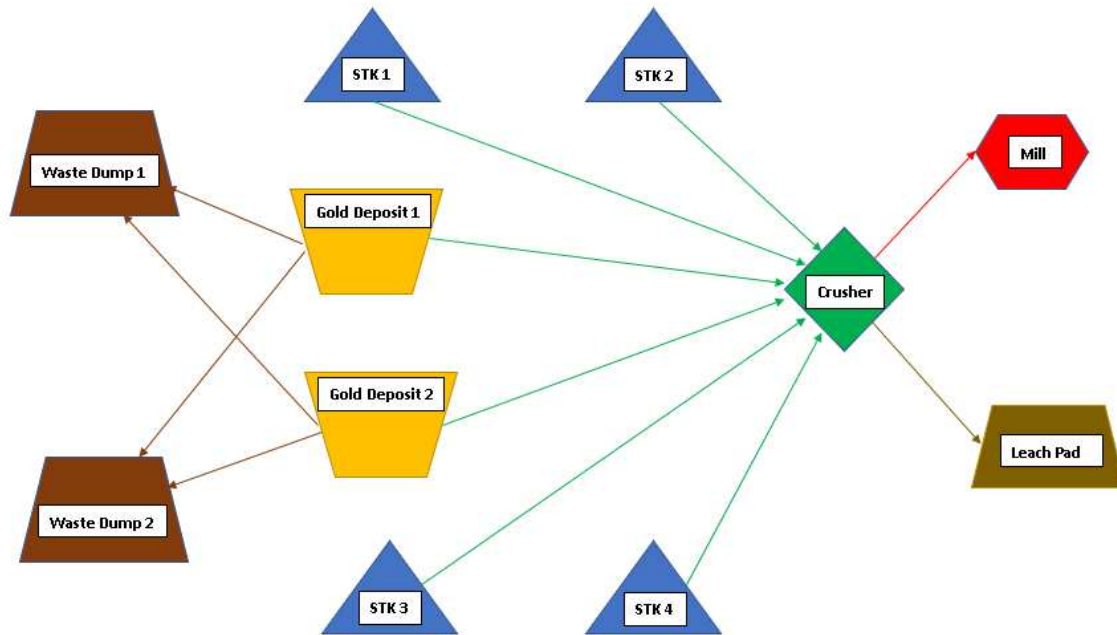


Figure 5.13: Gold Deposit Mine process flow

5.2.1 The Ultimate Pits

In this example, the destinations of the blocks are pre-determined. By using 1200\$/oz gold price, the block values are calculated for the designated destination which can be either mill, leach, waste dump 1 or waste dump 2. Moreover, the pits are composed of 17 geotechnical zones where the slope angles may vary from 34° to 58° from one zone to another. In order to honor the slope requirements, the proposed cone generation technique discussed in Appendix A is used to form the arcs between the blocks. Once the arc file is generated, the ultimate pits are determined by implementing the pseudoflow algorithm. The ultimate pits can be determined with one pseudoflow run where the strong set of trees are formed within the independent pits which will eventually translate into two disconnected set of strong trees where both sets will be extracted. Table 5.10 illustrates the number of blocks and tonnage of material within the ultimate pits. As seen the significant proportion of the ore blocks are leach blocks. Also, the number waste blocks are relatively low which leads to a stripping ratio less than 1. Moreover, the total undiscounted value of the pits is found as \$701,879,000. Figure 5.14 shows a cross section from East-28000 location.

The colors represent the designated destinations of the blocks. Also, the plan view of the ultimate pits is shown in Figure 5.15 and Figure 5.16 shows a cross section from North-43000 location. Once the ultimate pits are determined, the next step is to generate the mine plan.

Table 5.10: Summary of the blocks in the ultimate pit

MATERIAL TYPE	BLOCKS	TONNAGE
TOTAL	46,585	288,598,000
ORE	34,643	220,627,000
MILL	1,386	8,891,500
LEACH	33,257	211,735,500
WASTE 1	8,448	51,582,400
WASTE 2	3,494	16,388,600

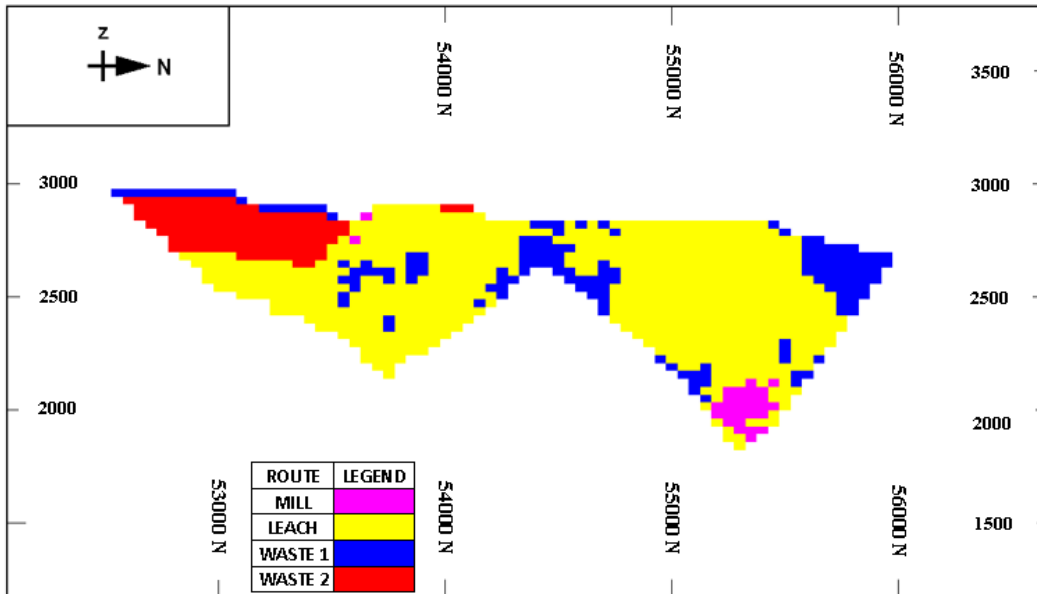


Figure 5.14: East-28000 cross section taken from the ultimate pit

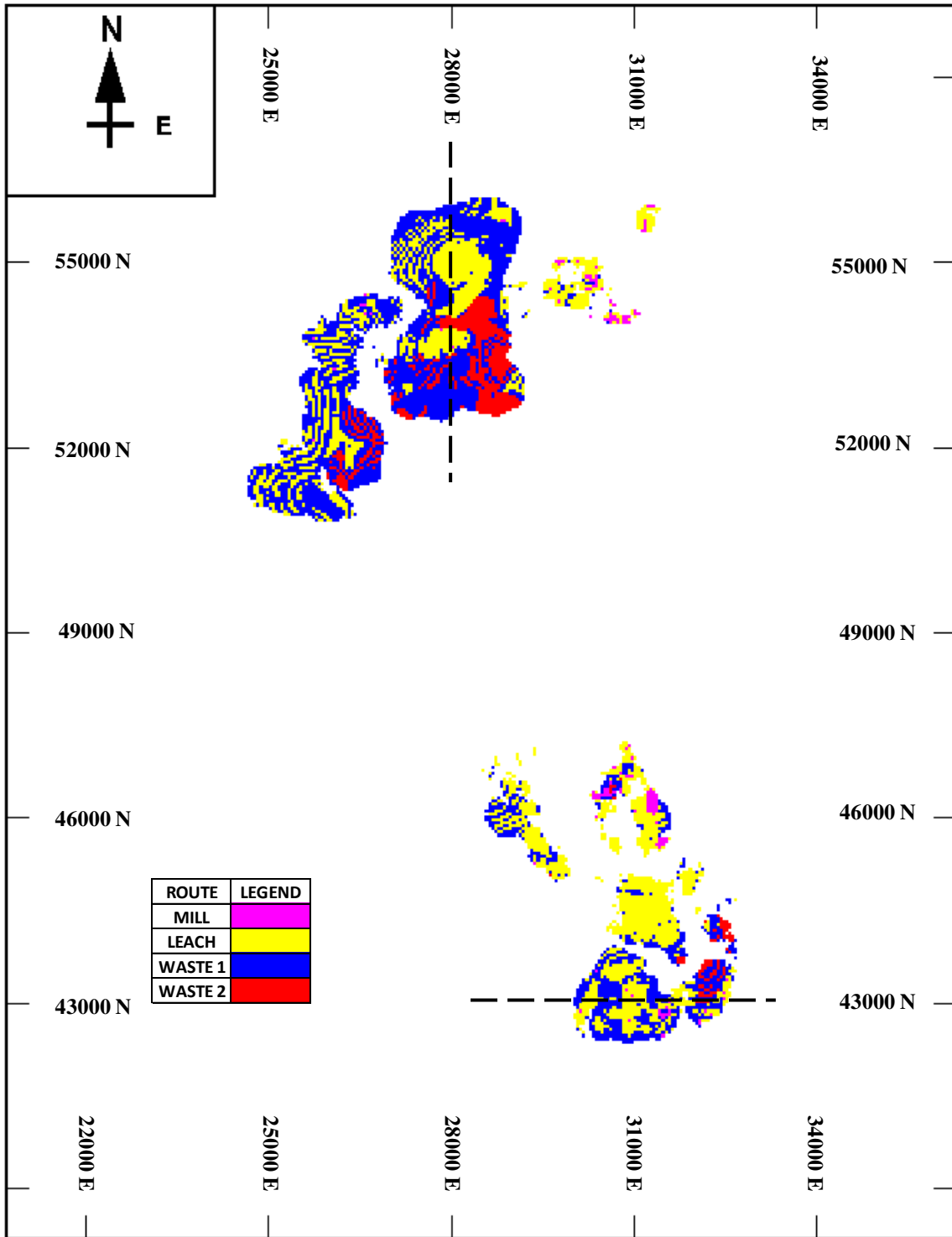


Figure 5.15: Plan view of the ultimate pits

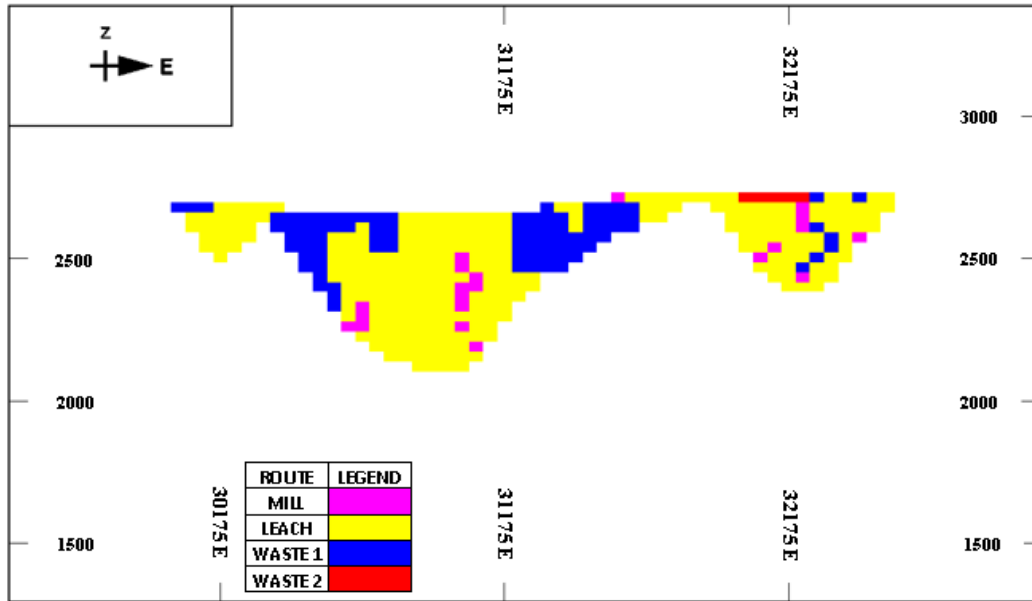


Figure 5.16: North- 43000 cross section taken from the ultimate pit

5.2.2 Mine Production Schedules

The optimal mine plan will be generated for a time horizon of 8 years with the implementation of the new integer solution algorithm. The discount factor will be assumed as 7%. The production requirements encompass crusher capacity, mill capacity and total mining capacity as shown in Table 5.11. The leach pads have enough capacity to handle all the leach materials, henceforth there is no need to include a leach capacity constraint into the model. There are 4 stockpiles at the site with initial capacities and the average grades shown in Table 5.12. Moreover, the stockpile materials can be only processed at the mill. However, the stockpile materials will be treated at the old mill for the first 6 months and after that a new mill will be available to treat the stockpile materials with enhanced recoveries. Table 5.12 also shows that the processing costs of the stockpile materials at the old mill are lower than the costs at the new mill.

Table 5.11: Mine plan production scheduling requirements

PERIOD	CRUSHER CAPACITY	MILL CAPACITY	MINING CAPACITY
1st 6months	11,000,000	906,660	21,500,000
2nd 6months	11,000,000	906,660	21,500,000
3	22,000,000	1,813,320	43,000,000
4	22,000,000	1,813,320	43,000,000
5	22,000,000	1,813,320	43,000,000
6	22,000,000	1,813,320	43,000,000
7	22,000,000	1,813,320	43,000,000
8	22,000,000	1,813,320	43,000,000
9	22,000,000	1,813,320	43,000,000

Table 5.12: Initial stockpile parameters

STOCKPILES	TONNAGE	AVG. GRADE	RECOVERIES		COST (\$/t)		
			OLD MILL	NEW MILL	HAULAGE	OLD MILL	NEW MILL
STK 1	436000	0.150 (oz/t)	60.0%	93.0%	0.5	16.1	19.2
STK 2	385000	0.070 (oz/t)	10.0%	90.0%	0.5	16.1	19.2
STK 3	614000	0.055 (oz/t)	52.4%	87.0%	0.5	16.1	19.2
STK 4	1338000	0.085 (oz/t)	65.0%	90.0%	0.5	16.1	19.2

Since the stockpile materials will be treated differently for the first 6 months, the production requirements of the first year are split into 6 months intervals. After the first year, the mine plan will be generated based on yearly intervals. The results of the mine production schedule are illustrated in Table 5.13 where all the production requirements are honored. First of all, the scheduler prioritized the stockpile 4 to feed the mill in the first 6 months since the stockpile 4 has the highest recovery value. The second half of the first year, stockpile 1 is fully consumed since the stockpile has the highest average grade which translates into a higher profit that needs to be realized before reduced by a discount factor. Also, no mill material is mined from the pits during the second half of the year which shows that the average grade of the ore available from stockpile 1 and stockpile 4 is higher than the available ore in the pit. Furthermore, the behavior of the tonnage and the average grade of the material processed in the mill is shown in Figure 5.17 and the tonnage and the average grade of the material treated in the leach pads is shown in Figure 5.18. It can be observed in Figure 5.17 that the mill capacity is not filled in years 4,7,8 and 9. However, Figure

5.18 shows us that the crusher capacity is fully utilized for those years. This indicates that mill material is not available for extraction without taking leach material, which cannot be accomplished since the crusher is working at capacity. Also, there is a shortage in the crushed material tonnage in the second half of the first year and year 6. It is clearly seen in Figure 5.17 that the mill is fully utilized for those periods. In this case, it appears that the mill material is overlaying the leach material, where the extraction of leach material is not possible without mining more mill material.

Table 5.13: Summary of the mine production scheduling results

PERIOD	MINED TONNAGE		STOCKPILE RECLAIM TONNAGE				PROCESSED TONNAGE		
	MILL	LEACH	1	2	3	4	CRUSHER	MILL	LEACH
1st 6months	896,243	10,095,839				7,918	11,000,000	904,161	10,095,839
2nd 6months	0	1,092,146	436,000				1,998,806	906,660	1,092,146
3	297,269	20,182,863		385,000	271,629	859,422	21,996,183	1,813,320	20,182,863
4	1,498,990	20,256,603			244,407		22,000,000	1,743,397	20,256,603
5	1,809,530	20,187,070			3,400		22,000,000	1,812,930	20,187,070
6	1,808,634	18,784,065			4,686		20,597,385	1,813,320	18,784,065
7	1,558,302	20,430,398			11,300		22,000,000	1,569,602	20,430,398
8	365,941	21,612,117			21,942		22,000,000	387,883	21,612,117
9	249,672	21,727,295			23,033		22,000,000	272,705	21,727,295

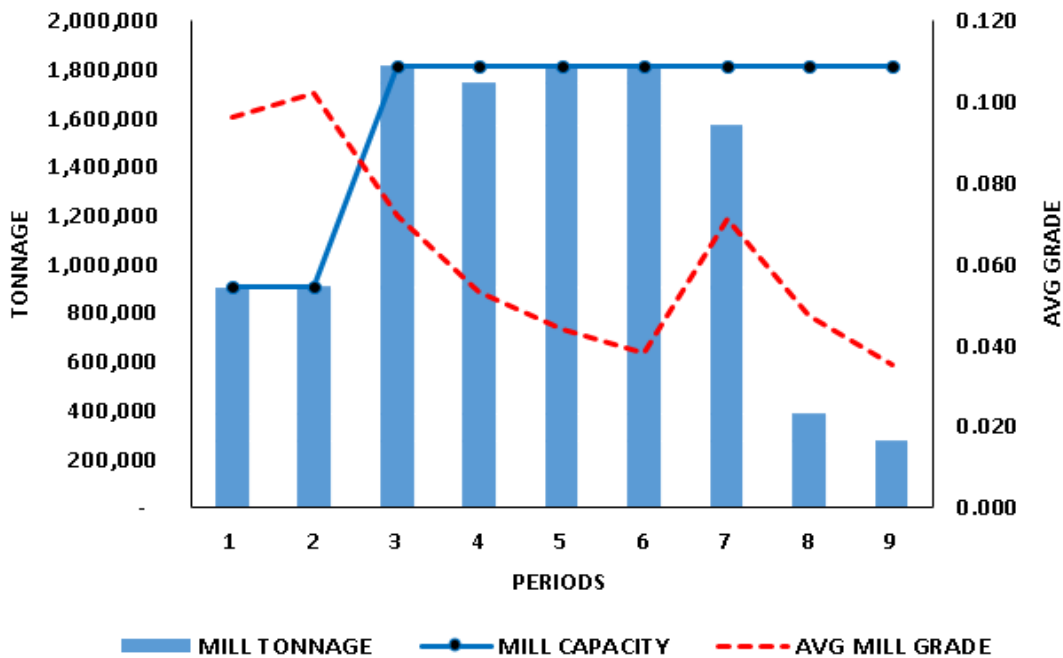


Figure 5.17: Yearly processed tonnage and the average grade at the mill

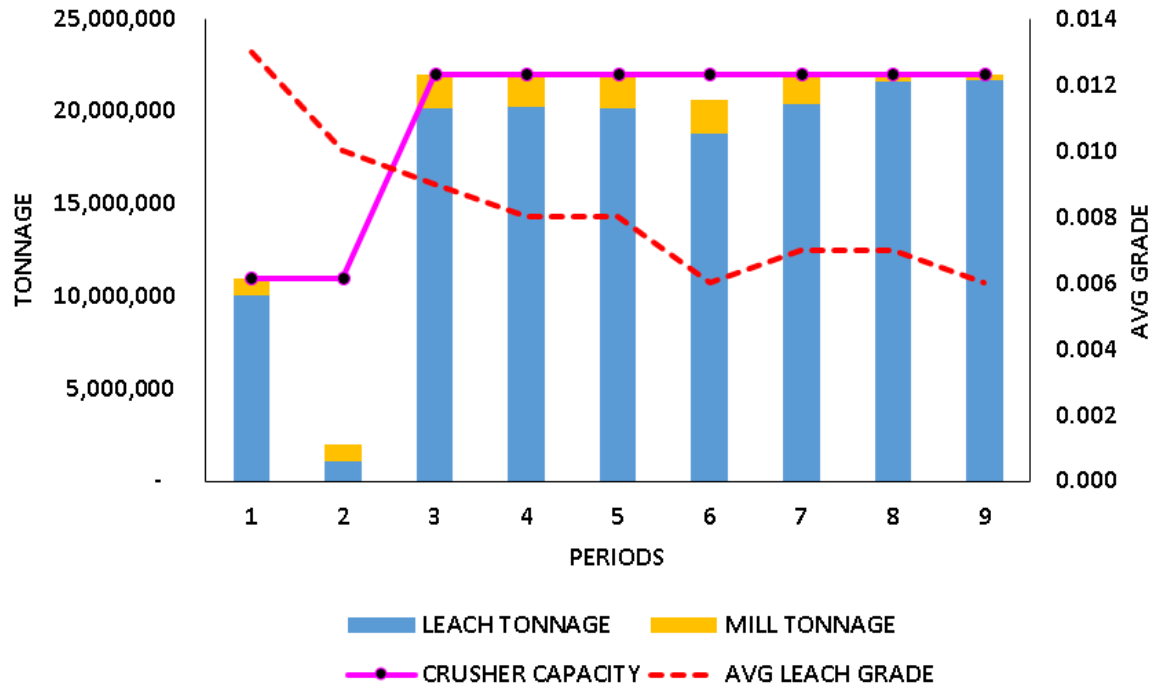


Figure 5.18: Yearly processed mill and leach tonnage and the average grade at the leach pad

As shown in Table 5.14, the theoretical upper bound of this problem is found to be \$778,426,000 whereas the integer solution is \$774,108,000 . The optimality gap is 0.55% which is achieved in 33 minutes. Since the optimality gap is so small, the quality of the integer solution is successfully verified.

Table 5.14: Summary of the results

LP NPV @ 7%	\$778,426,000
IP NPV @ 7%	\$774,108,000
OPTIMALITY GAP %	0.55%
SOLUTION TIME	33 min

The yearly schedules are also illustrated on a plan view in Figure 5.19, also on a cross section from East-27700 location in Figure 5.20 and North-55200 location in Figure 5.21. The colors represent the time period when each block is mined.

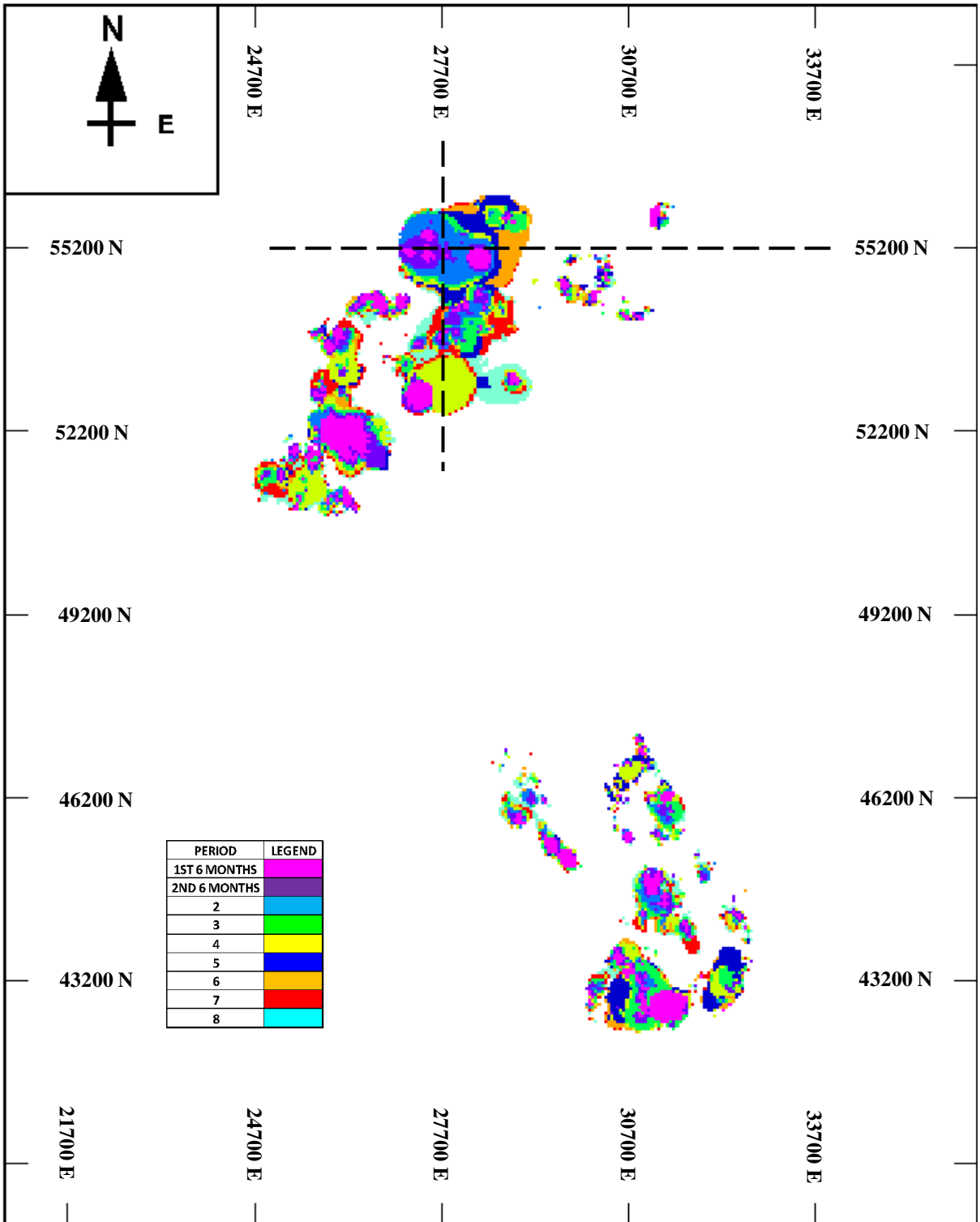


Figure 5.19: Yearly production schedules shown on a Plan View

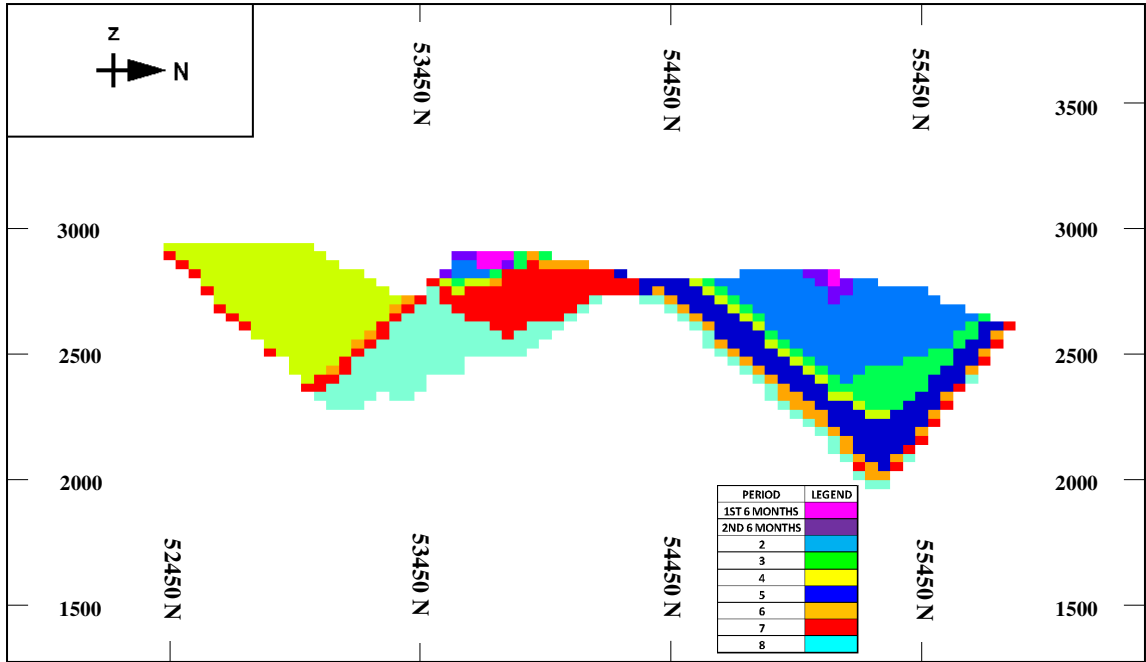


Figure 5.20: Yearly production schedules shown on East-27700 cross section

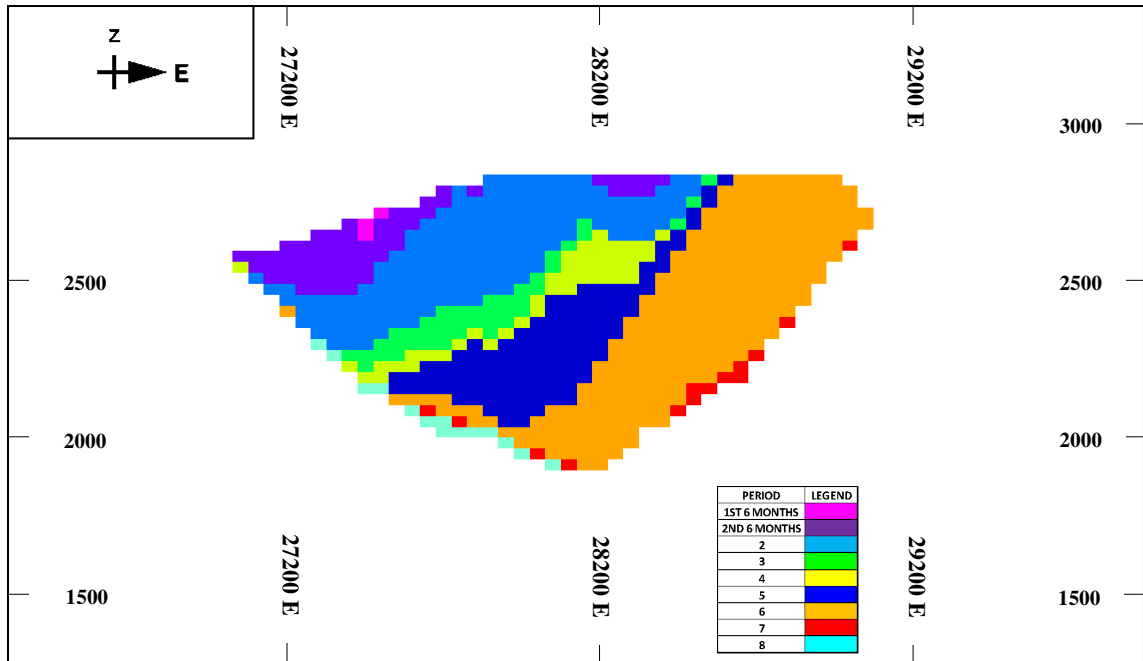


Figure 5.21: Yearly production schedules shown on North-55200 cross section

The case studies demonstrate the behavior of the components of the mining system, their interactions with each other, and how the targets of system production, grades and risks can be realized with a true optimization technique. The strength of the new integer solution algorithm will be further supported by the results obtained from the implementation on various types of mine production scheduling problems. Table 5.15 presents the results generated by scheduling the McLaughlin deposit with pre-determined destinations, on various ranges of time horizons and different uniform slope angle requirements by enforcing process capacity, grade blending and risk blending constraints. The optimality gaps range from 0.0003% to 1.3%. The next Table 5.16 illustrates the results obtained by scheduling the McLaughlin deposit for 3, 5 and 10 years using a dynamic cutoff grade strategy technique which allows the optimizer to pick the best destination for the blocks and with uniform 45⁰ slope angle requirements. The mine plan is constrained with mill and leach capacity, mill and leach grade blending, and risk blending constraints. The total number of variables range from 2.2 million to 7.3 million and the resulting optimality gaps range from 0.01% to 0.05%. Table 5.17 demonstrates the solutions obtained for the multi pit Gold deposit that consists of 17 geotechnical zones. Both mine plans are scheduled for 9 years and subject to the same total mining capacity, mill capacity and crusher constraints, however they are modeled under different slope angle requirements which changes the schedules drastically. The optimality gaps of 0.18% and 0.55% are again significantly small.

So far, 15 large scale open pit mining problems have been scheduled by implementing the new integer solution algorithm. The problems were subject to various pit slope requirements, multi destinations and multiple capacity and blending constraints with upper and lower bounds. The results demonstrate that for 11 out of 15 problems, the optimality gaps are less than 0.2%, three problems have a gap between 0.2% and 0.7% and only one problem ended up with an optimality gap of 1.3%. Although it cannot be proven that the integer solutions are true optimal integer solutions, we know that the optimality gap generated between the true optimal LP and the true optimal IP solution of the small 2D integer example in the previous chapter was 3.75%. Moreover, it was also mentioned a couple times by the developers of the BZ algorithm that the integrality gaps between the linear relaxation of the problems and the integer programs are often small. Hence, the fact that resulting gaps are so small may show that the integer solution to the problem may be the true optimal integer solution, if not the tightness of the gaps proves the quality of the integer solutions.

Table 5.15: Summary of the results for scheduling the McLaughlin Deposit with predetermined destinations together with the set of constraints enforced

MCLAUGHLIN DEPOSIT-FIXED DESTINATION-9X9X9 SLOPE PATTERN					
PERIODS	TOTAL VARIABLES	LP NPV	IP NPV	GAP	TIME
3YRS v1	874,989	\$2,089,565,000	\$2,076,065,000	0.65%	0.5h
3YRS v2	874,989	\$2,138,260,000	\$2,138,140,000	0.01%	0.5h
3YRS v3	874,989	\$2,087,750,000	\$2,087,425,000	0.02%	0.5h
3YRS v4	874,989	\$1,901,165,000	\$1,896,085,000	0.27%	0.5h
5YRS	1,458,315	\$1,682,040,000	\$1,682,020,000	0.001%	2h
7YRS	2,041,641	\$1,729,345,000	\$1,729,180,000	0.01%	2.5h
9YRS	2,624,967	\$1,589,660,000	\$1,589,410,000	0.02%	5.8h
10YRS	2,916,630	\$1,569,635,000	\$1,549,255,000	1.30%	5.8h
MCLAUGHLIN DEPOSIT-FIXED DESTINATION-45 DEGREE SLOPE					
3YRS	762,567	\$2,132,655,000	\$2,132,590,000	0.003%	1h
5YRS	1,270,945	\$1,714,060,000	\$1,714,055,000	0.0003%	4h

CONSTRAINTS	BOUNDS
MILL CAPACITY	<=
MILL AVG GRADE	>=
PROP. INFERRED	<=
PROP. INDICATED	>=
PROP. MEASURED	>=

Table 5.16: Summary of the results for scheduling the McLaughlin Deposit with dynamic cutoff grade strategy and the regarding set of constraints

MCLAUGHLIN DEPOSIT-MULTI DESTINATIONS-45 DEGREE SLOPE					
PERIODS	TOTAL VARIABLES	LP NPV	IP NPV	GAP	TIME
3YRS	2,210,553	\$2,070,265,000	\$2,069,205,000	0.05%	0.7h
5YRS	3,684,255	\$1,925,585,000	\$1,925,045,000	0.03%	2h
10YRS	7,368,510	\$1,581,250,000	\$1,581,085,000	0.01%	5.5h

CONSTRAINTS	BOUNDS
MILL CAPACITY	<=
LEACH CAPACITY	<=
MILL AVG GRADE	>=
LEACH AVG GRADE	>=
PROP. INFERRED	<=
PROP. INDICATED	>=
PROP. MEASURED	>=

Table 5.17: Summary of the results for scheduling the multi pit Gold Deposit subject to 17 slope zones and the corresponding set of constraints

GOLD DEPOSIT-MULTI PIT-FIXED DESTINATION-17 SLOPE ZONES					
PERIODS	TOTAL VARIABLES	LP	IP	GAP	TIME
9	412,704	\$793,187,000	\$791,776,000	0.18%	1h
9	419,265	\$778,426,000	\$774,108,000	0.55%	0.5h

CONSTRAINTS	BOUNDS
MINING CAPACITY	<=
CRUSHER CAPACITY	<=
MILL CAPACITY	<=

CHAPTER 6.

CONCLUSIONS

The production scheduling problem is a critical component of open pit mine planning whose solution impacts the cash flows of the operations. As stated in the introductory paragraph of this thesis, an open pit mine production schedule that maximizes the profitability of the mining operation subject to operational constraints is a very important issue for any mining problem. A complex mine production scheduling system requires a mathematical model in order to integrate all the factors characterizing the interactions between the blocks and the operational mining system. So far there is no known algorithm, either commercially available or presented in the literature, that can provide an optimal integer solution to the open pit mine production scheduling problem with capacity constraints together with lower and upper bound blending constraints.

The major contributions from the research discussed in this thesis are listed below:

1. Developed a novel integer solution algorithm which can solve the block by block mine production scheduling problems modeled with multi capacities, grade blending, grade uncertainty, stockpiles, variable pit slopes, multi destinations and truck hours.
2. Developed a program in C++ platform to implement the new integer solution algorithm to the large scale open pit mining problems.
3. The new integer solution algorithm achieved optimality gaps less than 1.3% in all the large scale open pit mine production scheduling problems tested which is the best ever reached.
4. Developed algorithm can be used as a phase design tool that serves as a practical guide for mine planning.
5. Developed a new cone pattern generation scheme which can handle variable pit slope angles based on complex geotechnical zones or multiple azimuths with any size block dimensions.
6. Introduced a new variable substitution methodology to formulate the open pit mine production scheduling problem with significantly reduced number of sequencing constraints.

7. Developed new refining rules to orthogonalize the set of partitions between the consecutive iterations of the new integer solution algorithm which is different than the orthogonalization procedure given in Bienstock Zuckerberg (2009).

The major contributions are discussed further in detail in the following paragraphs. These contributions should have a significant impact in the field of mine planning and a major enhancement to industry in their desire to maximize profitability.

In this thesis, a novel integer solution algorithm which can solve the mine production scheduling problems modeled with multi capacities, grade blending, grade uncertainty, stockpiles, variable pit slopes, multi destinations and truck hours was developed. Since the new integer solution algorithm also solves the LP relaxation of the mine production scheduling problem as part of the solution process, the theoretical upper bound is provided together with the integer solution which will allow the user to assess the quality of the integer solutions from the resulting optimality gap.

A program was developed on a C++ platform to implement the new integer solution algorithm. The program was tested on 15 large scale open pit mining problems subject to various pit slope requirements, multi destinations and multiple capacity and blending constraints with upper and lower bounds. 11 out of 15 problems, the optimality gaps were less than 0.2% and the highest optimality gap encountered was 1.3%. Furthermore, the strength of the proposed integer solution algorithm is highlighted by the ability of solving open pit mine production scheduling problems that have more than 7 million variables as an integer problem with an optimality gap as small as 0.01 % within 5 hours 30 minutes as an example.

The developed software in this thesis can also function as a phase design tool that serves as a practical guide for mine planning. Since the underlying algorithm generates optimal plans; the pushback generation with the new integer solution algorithm will honor all sorts of productional requirements which makes the solution strategy far more advanced than the traditional phase design methods with parametrization techniques.

A new cone pattern generation scheme was developed in this thesis which can handle variable pit slope angles based on complex geotechnical zones or multiple azimuths with any size block dimensions. The technique will minimize the number of arcs required to represent the actual

pit slope angles on a block level, which will allow the new integer solution algorithm to work with fast solution times.

A new mathematical model was developed in this thesis by introducing the dual path double “by” variable “ u_{bd}^t ” by summing the single path “by” variable “ w_{bd}^t ” across the destinations. The mathematical model incorporates the production and processing capacities, grade blending constraints, risk blending constraints, multi destinations, stockpiles and variable pit slope angles on a block by block level. The derived model represents a generic form which will make it easier for the operator to adjust the model based on the needs of the operation.

As the new integer solution algorithm requires master problem to be formulated with the set of orthogonal columns, new refining rules were developed in order to orthogonalize each integer feasible “orthogonal” column from the current iteration’s partition set with each one of the “orthogonal” partitions from the previous iterations’s partition set.

6.1 Recommended Future Work

The new integer solution algorithm developed in this thesis will contribute to the ongoing research in mine production scheduling as well as to the general fields of operations research.

The traditional approach towards solving a mine planning problem is limited from a deterministic point of view which neglects the level of uncertainty involved in a mining operation at the expense of not meeting the production targets promised to the shareholders. As the commodity price and the ore grade are the major uncertain components of the complex mining system, instead of running a plan by incorporating their estimated value, a wide range of possible values should be generated by simulations and the simulated values should be considered in the mine plan. This requires a stochastic modeling which has a limited implementation in the mine planning due to the massive number of variables attained by the simulations. The currently adapted solution methods are heuristic methods which are not fundamentally mathematically sophisticated so as to define the optimality gap. On the other hand, we know that the decomposition algorithms can guarantee the optimality of the linear relaxation solutions. Therefore, research into the implementation of the new integer solution algorithm to solve the large scale stochastic mine production scheduling algorithm is suggested.

Another important direction of future research relates to the development of a methodology to generate practical mineable phases which is the widely accepted mine planning approach today. Using the new integer solution algorithm provided in this thesis will greatly improve this approach to mine planning. Incorporation of the minimum mining widths to the new integer solution algorithm should be further investigated.

So far, the new integer solution algorithm has been tested on 15 large scale open pit mine production scheduling problems subject to various type of constraints and the optimality gaps were all less than 1.3%. More testing is suggested in order to demonstrate and confirm the tightness of the optimality gap over a wide range of large-scale mine production scheduling problems and operations.

The implementation of the new solution algorithm is not limited to the mining industry. In fact, any precedence constrained production scheduling type problem can be solved with the new solution algorithm. Therefore, there seems to be an advantage in testing the new integer solution algorithm on production scheduling problems in general. We know that the totally unimodular structure of the subproblems allows fast exploitation of the underlying network structure during the integerizing process. However, there could be an efficient way to generate integer feasible solution columns by solving the subproblems which do not constitute a totally unimodular structure. This should be further investigated.

REFERENCES

- Akaike, A. (1999). Strategic planning of long-term production schedule using 4d network relaxation method. Ph.D. Thesis, Colorado School of Mines, Golden, CO.
- Amaya, J., Goycoolea, M., Moreno, E., Prevost, T., and Rubio, E. (2009). A scalable approach to optimal block scheduling. 34th APCOM, 567–575.
- Barnes, R.J., and Johnson, T.B. (1982). Bounding techniques for the ultimate pit limit problem. 17th APCOM, Colorado School of Mines, Golden, CO.
- Bienstock, D., and Zuckerberg, M. (2009). A new LP algorithm for precedence constrained production scheduling. *Industrial Engineering*, 1–33.
- Bienstock, D., and Zuckerberg, M. (2010). Solving LP relaxations of large-scale precedence constrained problems. *Lecture Notes in Computer Science*, 1-14.
- Bienstock, D., and Zuckerberg, M. (2015). A new LP algorithm for precedence constrained production scheduling. *Industrial Engineering*, 1–53.
- Boland, N., Dumitrescu, I., and Froyland, G. (2008). A multistage stochastic programming approach to open pit mine production scheduling with uncertain geology. *Operations Research*, 1–33.
- Chandran, B. G., & Hochbaum, D. S. (2009). A computational study of the pseudoflow and push-relabel algorithms for the maximum flow problem. *Operations Research*, 57(2), 358–376.
- Chen, T. (1976). 3D pit design with variable wall slope capabilities. *Proceedings of the 14th Symposium on the application of computers and operations research in the mineral industries (APCOM)*, 615-625.
- Chicoisne, R., Espinoza, D., Goycoolea, M., Moreno, E., and Rubio, E. (2012). A new algorithm for the open-pit mine production scheduling problem. *Operations Research*, 60(3).
- Cullenbine, C., Wood, R. K., and Newman, A. (2011). A sliding time window heuristic for open pit mine block sequencing. *Optimization Letters*, 5(3), 365–377.
- Cullenbine, C. A. (2011). Theoretical and computational advances for open pit block sequencing and network diversion. Ph.D. Thesis, Colorado School of Mines, Golden, CO.

- Dagdelen, K., and Francois-Bongarcon D. (1982). Towards the complete double parameterization of recovered reserves in open pit mining. 17th APCOM, 288- 296.
- Dagdelen, K. (1985). Optimum multi-period open pit mine production scheduling. Ph.D. Thesis, Colorado School of Mines, Golden, CO.
- Dantzig, G. B. (1987). Origins of the simplex method. *A History of Scientific Computing*, 13.
- Everett, H. (1963). Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 399-417.
- Gauthier, F.J. and Gray, K. G. (1971). Pit design by computer at Gaspe Copper Mines, Limited. *CIM Bulletin*, 95-102.
- Gershon, M. E. (1983). Mine scheduling optimization with mixed integer programming. *Mining Engineering*, (April), 351–354.
- Gershon, M. E. (1983). Optimal mine production scheduling: evaluation of large scale mathematical programming approaches. *International Journal of Mining Engineering*, 1(4), 315–329.
- Gershon, M. (1987). Heuristic approaches for mine planning and production scheduling. *International Journal of Mining and Geological Engineering*, 5(1), 1–13.
- Giannini, L. M. (1990). Optimum design of open pit mines. Ph.D. Thesis, Curtin University of Technology, AU.
- Gilani, S. O., and Sattarvand, J. (2015). A new heuristic non-linear approach for modeling the variable slope angles in open pit mine planning algorithms. *Acta Montanistica Slovaca*, 20(4), 251–259.
- Gilbert, James W. (1966). A mathematical model for the optimal design of open pit mines. M.Sc. Thesis, Department of Industrial Engineering, University of Toronto.
- Goycoolea, M., Moreno, E., and Rivera, O. (2015). Comparing new and traditional methodologies for production scheduling in open pit mining. *APCOM 37th*, 352–359.
- Hochbaum, D. S. (2009). The Pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Operations Research*, 56(4), 992–1009.
- Johnson, T.B. (1968). Optimum open pit mine production scheduling. Ph.D. Thesis, University of California, Berkeley, CA.

- Kawahata, K. (2006). A new algorithm to solve large-scale mine production scheduling problems by using the Lagrangian relaxation method. Ph.D. Thesis, Colorado School of Mines, Golden, CO.
- Khalokakaie, R. (1999). Computer aided optimal open pit design with variable slope angles. Ph.D. Thesis, Department of Mining and Mineral Engineering, The University of Leeds, UK.
- Khalokakaie, R., Dowd, P.A., and Fowell, R.J. (2000). A Windows program for optimal open pit design with variable slope angles. *International Journal of Surface Mining, Reclamation and Environment*, 14(4), 261-275.
- King, B. W. (2016). Using integer programming for strategic underground an open pit to underground scheduling. Ph.D. Thesis, Colorado School of Mines, Golden, CO.
- Lambert, W.B. (2012). Techniques to reduce the solution time of the open pit block sequencing problem. Ph.D. Thesis, Colorado School of Mines, Golden, CO.
- Lambert, W. B., and Newman, A. M. (2013). Analyzing solutions of the openpit block sequencing problem obtained via Lagrangian techniques. *Mining Engineering*, 65 (2), 37–43.
- Lambert, W. B., and Newman, A. M. (2014). Tailored Lagrangian Relaxation for the open pit block sequencing problem. *Annals of Operations Research*, 222, 419–438.
- Lamghari, A., and Dimitrakopoulos, R. (2012). A diversified Tabu search approach for the open-pit mine production scheduling problem with metal uncertainty. *European Journal of Operational Research*, 222(3), 642–652.
- Lamghari, A., Dimitrakopoulos, R., and Ferland, J. A. (2014). A variable neighborhood descent algorithm for the open-pit mine production scheduling problem with metal uncertainty. *Journal of the Operational Research Society*, 65(9), 1305–1314.
- Lamghari, A., and Dimitrakopoulos, R. (2015). Network-flow based algorithms for scheduling production in multi-processor open-pit mines accounting for metal uncertainty. *European Journal of Operational Research*, 250(1), 273–290.
- Lerchs, H. and Grossmann, I.F. (1965). Optimum design of open-pit mines. *Transactions on CIM LXVII*, 47–54.
- Lipkewich, M.P. and Borgman, L. (1969). Two and three-dimensional pit design optimization techniques. *A Decade of Digital Computing in the Mineral Industry*, 505-523.
- Meagher, C., and Dimitrakopoulos, R., and Avis, D. (2014). Optimized open pit mine design, pushbacks and the gap problem-a review. *Journal of Mining Science*, 50(3), 508–526.

- Osanloo, M., Gholamnejad, J., and Karimi, B. (2008). Long-term open pit mine production planning: a review of models and algorithms. *International Journal of Mining, Reclamation and Environment*, 22(1), 3–35.
- Pana, M.T. (1965). The simulation approach to open pit design. *Proceedings of the 5th APCOM Symposium, Tucson, Arizona, zz1-zz24*.
- Ramazan, S. (1996). A new push back design algorithm in open pit mining. M.Sc. Thesis, Colorado School of Mines, Golden, CO.
- Ramazan, S. and Dagdelen, K. (1998). A new push back design algorithm in open pit mining. *Mine Planning and Equipment Selection*, 119-124.
- Ramazan, S. (2001). Open pit mine scheduling based on fundamental tree algorithm. Ph.D. Thesis, Colorado School of Mines, Golden, CO.
- Ramazan, S., and Dimitrakopoulos, R. (2004). Recent applications of operations research and efficient MIP formulations in open pit mining. *Transactions, Society for Mining, Metallurgy and Exploration*, 316(3), 73–78.
- Seymour, F. (1995). Pit limit parametrization from modified 3D Lerchs – Grossmann algorithm. *Society for Mining, Metallurgy and Exploration*, 95-96.
- Soleymani, M. and Sattarvand, J. (2012). Modeling of accurate variable slope angles in open-pit mine design using spline interpolation. *Archives of Mining Sciences*, 57(4), 921-932.
- Somrit, C. (2011). Development of a new open pit mine phase design algorithm using mixed integer linear programming. Ph.D. Thesis, Colorado School of Mines, Golden, CO.
- Tachefine, B., and Soumis, F. (1997). Maximal closure on a graph with resource constraints. *Computers & Operations Research*, 24(10), 981–990.
- Underwood, R., and Tolwinski, B. (1998). A mathematical programming viewpoint for solving the ultimate pit problem. *European Journal of Operational Research*, 107(1), 96–107.
- Van-Dunem, A. (2016). Open-pit mine production scheduling under grade uncertainty. Ph.D. Thesis, Colorado School of Mines, Golden, CO.
- Yegulalp, T.M. and Arias, J.A. (1992). A fast algorithm to solve ultimate pit limit problem. *Proceedings of the 23rd International Symposium on the Application of Computers and Operations Research in The Mineral Industries*, 391 – 398.
- Whittle, J. (1998). Beyond optimization in open pit design. *Proceedings of the First Canadian Conference on Computer Applications in the Mineral Industry*, 331 – 337.

Zhao, Y. (1992). Algorithms for optimum design and planning of open pit mines. Ph.D. Thesis, University of Arizona, Tucson, AZ.

Zhao, Y. and Kim, Y.C. (1992). A new optimum pit limit design algorithm. Proceedings of the 23rd International Symposium on the Application of Computers and Operations Research in the Mineral Industries, 423 – 434.

APPENDIX A.

MODELING WITH VARIABLE PIT SLOPE ANGLES

The iterative nature of the proposed open pit mine production scheduling solution algorithm requires the dependencies between the blocks based on the required pit slope angles to be preprocessed. This is accomplished by generating arcs between the blocks and storing them in map containers when programmed with a C++ coding language. Then the arcs between the blocks are transformed into sequencing constraints at every iteration of the algorithm. Hence, there is a strong correlation between the number of arcs generated versus the processing time of the sequencing constraints and the memory allocated to store these arcs. In order to achieve fast computing times, the number of arcs generated per block should be optimized. Therefore, a new cone pattern generation scheme is needed to eliminate the redundant arcs while minimizing the deviation from the required pit slope angle for block models with any size block dimensions.

Pit slope design is a fundamental element of the mining system where the slope stability is ensured for the life of the mine, which may extend beyond closure. The safe slope angles are determined to maximize the safety of the operating personnel and equipment. The pit may have multiple slope angles depending on the presence of faults, joints, water and the strength of a rock together with the slope governing boundary conditions and rock mass failure modes (Gilani and Sattarvand, 2015). The economic contributions of the pit slope angles to the mining operation are also significant. Pit slope angles form the domain that delineates the boundary of the mining system. As the pit proceeds deeper, the magnitude of the pit outline expansion will be dictated by the pit slopes. The operators tend to work with the safest steep slope angles to maximize the economic benefits. The steepened slopes will enable the production to proceed with less stripping. Moreover, there could be potential additional ore recovered because of the horizontal and vertical expansion at the pit bottom due to steeper slopes. (Figure A.1)

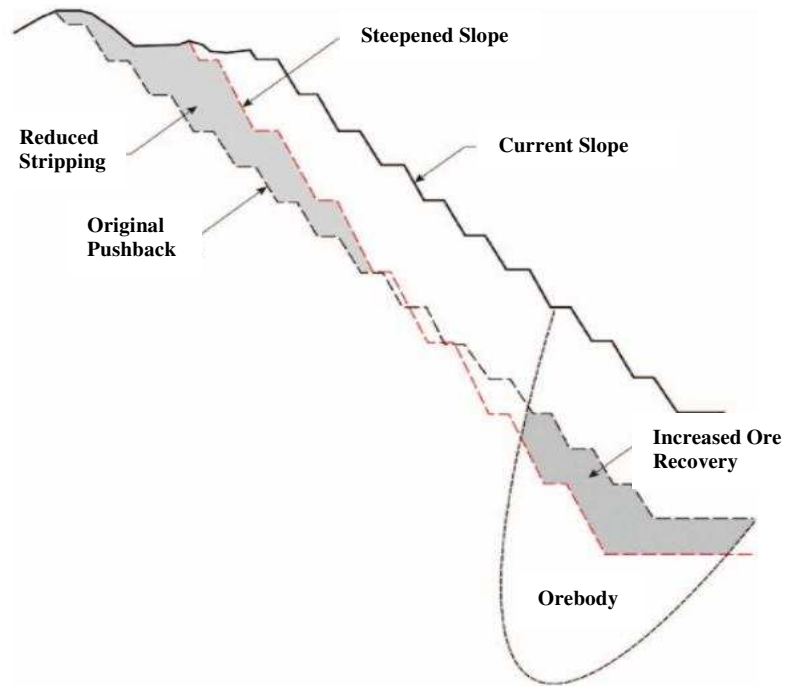


Figure A.1: Potential impacts of slope steepening (Read & Stacey, 2010)

Once the safe steep slope angles are determined, the challenge arises in integrating the pit slope angles into the mine production plans on a block by block basis. Mine schedules generated for a given period need to honor the pit slope angles on a block level. This is accomplished by creating dependencies between the target block and the blocks that should be removed until the required pit slope is achieved. The dependencies can be formed by connecting the target block by arcs to each one of the preceding blocks. Figure A.2 shows the directed graph $G = (V, A)$ where V represents the set of vertices (blocks) and A represents the set of arcs that impose the block dependencies based on the pit slope.

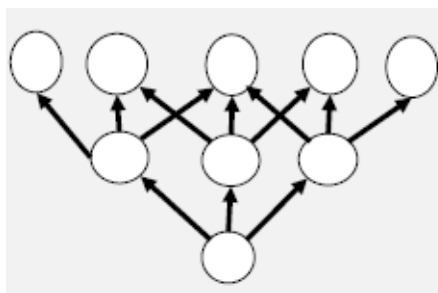


Figure A.2: Block dependencies on a directed graph

A.1 Background on Variable Pit Slope Angles

The arc generating concept that represents the pit slopes on a block level became a compelling subject among researchers as the geotechnical properties impose variations on the pit slopes from one azimuth direction to another also in vertical directions. Researchers have investigated various methods that would generate an extraction pattern to minimize the deviations from the required pit slope angles. The earliest research focused on generating a block extraction pattern based on repetitive implementation of certain block configurations where the block size is assumed to be cubical. Gilbert (1966) was first to show that given a cubic block model, 1:5:9 configuration gives the best approximation for a 45° pit slope as shown in Figure A.3. Johnson (1968) illustrated that if the pit is operating with a single slope angle different than 45° , a block configuration pattern can still be implemented by adjusting the size of the blocks. Lipkewich and Borgman (1969) proposed the knight's move pattern for a cubical block model to accomplish 45° slope angles. The method generates arcs from the target block to the 5 blocks on the level above and the 8 blocks on two levels above. The selection of the 8 blocks from the second level follows the movement pattern of the knight in a chess game. This method imposes a minimum search pattern of 13 blocks. Giannini (1990) showed that repeatedly applying a knight's move pattern will result with slope angles ranging from 41.8° to 45° . Unfortunately, the block configuration patterns are limited by hard assumptions on cubical block sizes and 45° pit slope angles. It is practically infeasible to represent the multiple variable pit slope angles and non-cubical block models with any block configuration patterns.

The shortcomings of the block configuration patterns shifted the direction of the research towards cone-based templates. The idea behind the cone-based templates is, once the cone is constructed on a base block, it is projected to the surface. If the midpoint of any block appears within the extraction cone, the block must be extracted before removing the base block. Moreover, the cone-based search methods can handle variable pit slopes with non-cubical block models. Chen (1976) implemented the cone-based search pattern to the predetermined pairs of azimuths and dips. The author used linear interpolation technique to smooth the pit slopes between the azimuths. Giannini (1990) proposed the idea of a minimum search pattern to remove the redundant arcs generated during the cone construction process. The search pattern is generated as follows:

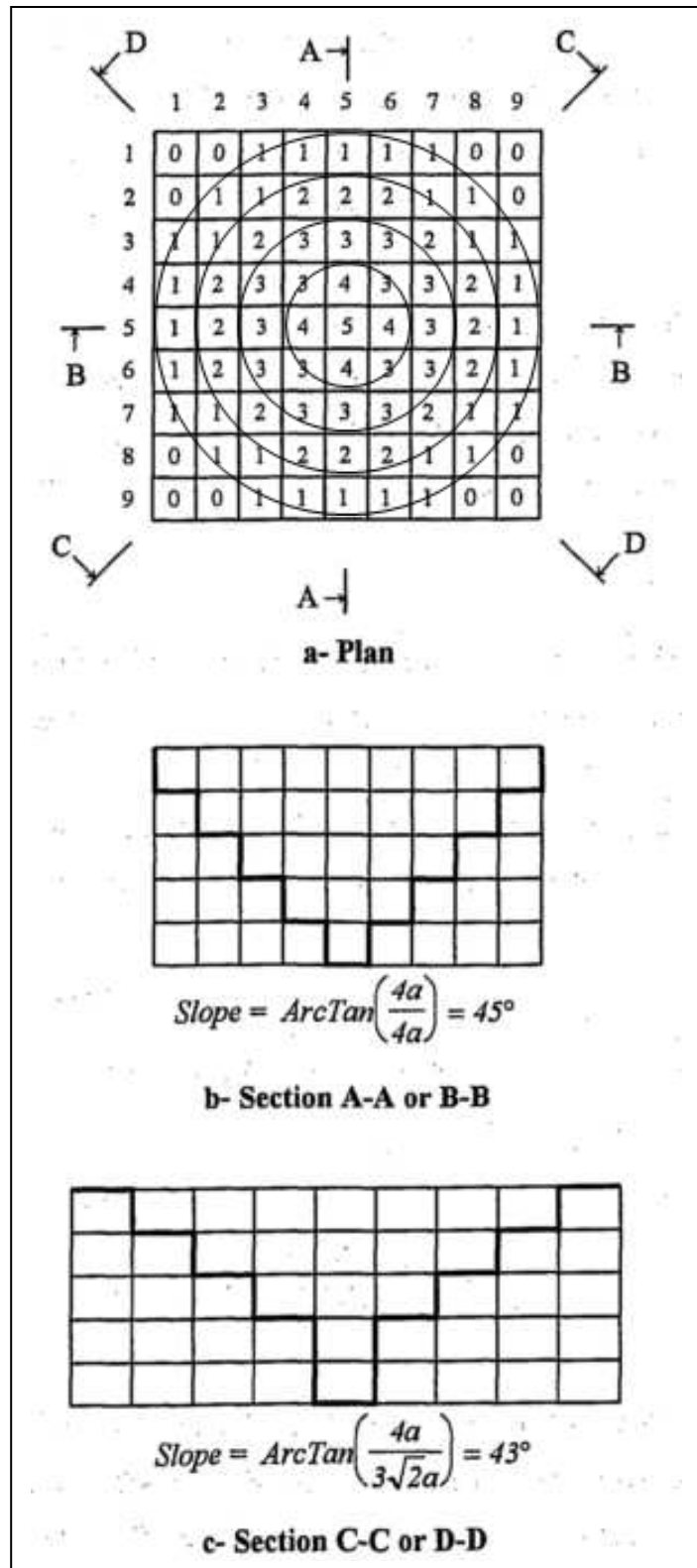


Figure A.3: 1:5:9 block pattern applied to a cubic revenue block model (Khalokakaie, 1999)

Given a vertex “ i ” representing the identification of the base block, the geometric pattern “ P ” that obeys the pit slope s_r on various azimuths B_r can be created by forming the set $A_i = \{b_1, b_2, b_3, \dots, b_k\}$ that holds the blocks to be removed when the block with vertex “ i ” is extracted. For each $b_j \in A_i$ there exists the preceding blocks selected with the same geometric pattern “ P ” captured in set A_j . In other words, for every block in set A_i from b_1 to b_k , there exists a set of preceding blocks from A_1 to A_k determined by the geometric pattern “ P ”. Again, there may be more preceding sets for every block within the sets of A_1 to A_k . The blocks from each one of the preceding sets are identified until the top level is reached and stored in a set “ T ”. If all the blocks extracted from the set “ T ” generates a conical pit that has the slope angle equal to s_r for each r on azimuth B_r , then A_i is called a search pattern for the block “ i ”. A pattern generated by minimum number of elements is called minimum search pattern (Giannini, 1990). If the pit consists of multiple azimuths, the author implements the linear spline interpolation technique to calculate the slope angles for different directions. Then, the search pattern is generated for different tolerance level inputs by the user. The number of blocks included in the pattern depends on the tolerance level. The author suggests that achieving 10% of the required wall slope angle may yield efficient and practical results. Khalokakaie (1999) proposed a cone method that considers variable slope angles on four principal directions (East, West, North, South). For a given base block, the author calculates the radius of the cone on four principal directions per level. Then for each direction for each level, the author calculates the number of blocks that the cone will mine by simply dividing the radius of the cone by the width of the block. The next step is to figure out if the remaining blocks will be included in the extraction cone. The author accomplishes it by using the ellipse equation. Eventually, the upper area of the extraction cone is split into four regions where each region falls between the consecutive principal directions and the regions are smoothed into an ellipse shape. Shishvan and Sattarvand (2012) investigated the spline interpolation techniques to smooth the pit walls for any number of slope angles in any direction. The authors compared quadratic, cardinal and cubic splines and concluded that the cubic spline generates smoother and more representative curves. The authors first identify the coordinates of the points where the slope line for a given slope direction intersects the blocks on each level and name them as corner points. Then, the authors generate spline interpolation matrices and solves them for the corner points on each level. The resulting tangent vectors are inserted into a cubical polynomial function which is solved by changing the value of “ t ” from zero to one in order to calculate the

coordinates of the points on each spline segment which the author refers to as border points. Then, for each level, the distance from the center of mass of each block to the cone apex is compared to the distance from the border points for the same direction in order to determine whether the block should be included in the cone template. Gilani and Sattarvand (2015) proposed a non-linear interpolation technique to construct a cone template for variable pit slope angles. The authors first calculate the radius of the cone for the predefined azimuths in all levels. Then, the authors implement an inverse distance law to interpolate the radius on azimuths from 0° to 360° with the increments of 10° . Lastly, the authors calculate the distance from the center of the base block to the center of the intended block and compare it against the radius of the cone along the direction of the intended block in order to specify whether or not the intended block is inside the cone template.

So far, the cone template generation techniques presented in the literature have been able to address the shortcomings of the block configuration techniques but smoothing the pit walls with a cone pattern generation technique that reduces the number of preceding blocks while minimizing the deviation from the pit slope angles still remains to be perfected. Chen (1976), Giannini (1990) and Khalokakaie (1999) used linear interpolation techniques to smooth the pit slopes from one azimuth direction to another. The linear interpolation approach results with polygonal shaped pits with sharp intersections (Figure A.4a) while nonlinear interpolation techniques achieve smoother pit walls (Figure A.4b). Moreover, the cone template generated by Khalokakaie (1999) was restricted with variable pit slopes for only principal directions that limits the practical implementation. Gilani and Sattarvand (2015) emphasized the shortcomings of the spline interpolation technique proposed by Shishvan and Sattarvand (2012) based on the modeling difficulties of the cone template whenever the pit slope consists of few slope directions. Although the technique proposed by Gilani and Sattarvand (2015) can handle any number of slope angles, the authors did not mention any pattern that can reduce the number of preceding blocks included in the cone of the base block.

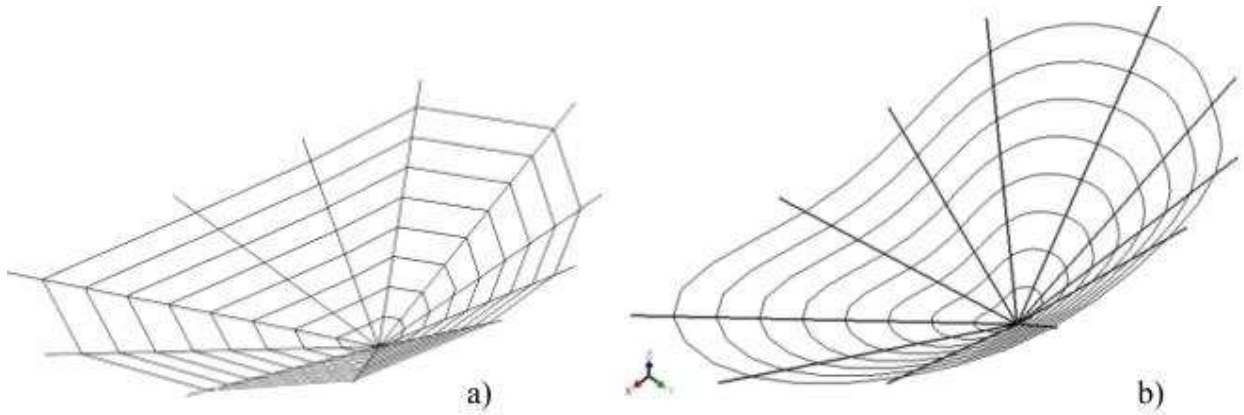


Figure A.4: Cone template generated by a) linear and b) non-linear interpolation (Shishvan and Sattarvand, 2012)

A.2 Constant Slope Angle Cone Pattern Generation

Initially a cone pattern generation technique will be shown for the pits modeled with constant slope angle for every direction for any size block. Assume b_i needs to be mined and due to the sequencing relationship enforced by the slope angle, the set $B_i = \{b_1, b_2, b_3, \dots, b_j\}$ which holds all the preceding blocks of b_i must also be mined. Depending on the location of the b_i in the pit, the number of preceding blocks in set B_i could be immense. The introduced cone pattern template will cut the size of the set B_i significantly. Let the set $C_i = \{b_1, b_2, b_3, \dots, b_s\}$ hold the preceding blocks of b_i identified by the cone pattern template. If the blocks b_1 to b_s within the cone pattern set C_i is replaced recursively such as $C_i = \{C_1, C_2, C_3, \dots, C_s\}$, then the set C_i becomes equal to the set B_i which proves that the cone pattern template will have sufficient blocks to represent the sequencing relationship.

The algorithm initially seeks a pattern on principal (N-S, E-W) and diagonal directions (NE-SW, NW-SE), and then the remaining directions within the boundary of the principal and diagonal patterns will be investigated. The boundary is essentially the highest level the pattern reached on principal and diagonal directions. Given i, j, k coordinate system, “ i ” represents the northing, “ j ” represents easting and k represents the elevation. The following illustrates the steps of the cone pattern generation algorithm. The cone apex is located on coordinate $(0,0,0)$ and expands towards the direction of the “ j ” axis. The pattern generated for the “ j ” axis applies to the “ i ” axis as well; since the pit slope is assumed to be constant for all directions. The following

steps can be repeated to generate a pattern for diagonal directions. For the diagonal directions, “ z ” axis which is located on the “ i-j ” plane will be used. The algorithm steps will only concentrate on the first quadrant of the Cartesian coordinate system. Working with a constant pit slope angle for all the directions makes it easier to reproduce the pattern generated on the first quadrant for the rest of the quadrants by simply reflecting the pattern over the “ i ” axis and “ j ” axis.

The pattern search is limited with 8 levels which is a user defined value. Moreover, the blocks are assumed to have a square base. Let`s assume the base block b_{ijk} is located on coordinates $i=0, j=0$ and $k=0$.

STEP 0

Calculate the radius of the cone for 8 levels. Calculate the horizontal distance from the center of the base block for 8 increments on principal and diagonal directions.

slope ← *required pit slope angle*

radius_l ← *radius of the cone pattern at level l*

CMP_j ← *horizontal distance from the center of the base block on principle direction*

CMD_z ← *horizontal distance from the center of the base block on diagonal direction*

$$radius_l = \frac{l \times block\ height}{\tan(slope)} \quad for\ l = 1\ to\ 8 \quad (A.1)$$

$$CMP_j = j \times block\ width \quad for\ j = 1\ to\ 8 \quad (A.2)$$

$$CMD_z = z \times block\ width \times \sqrt{2} \quad for\ z = 1\ to\ 8 \quad (A.3)$$

STEP 1

As the cone expands on “ j ” axis, the level at which the first block is captured with the cone pattern at $j=1$ becomes the starting pattern of the algorithm.

spatpk ← *the level at which the cone pattern starts on principal directions*

spatpj ← *the j coordinate at which the cone pattern starts on principal directions*

for l ← 1 to 8 *do*

if radius_l × 1.1 ≥ *CMP₁*

spatpk ← *l*

exit for loop

end for
spatpj ← 1

STEP 2

As the cone expands from $j=1$ to $j=8$, the level at which the first block is captured will be stored as a candidate for the end pattern. The decision to choose the end pattern among the set of candidate levels will be based on the minimum percentage deviation from the pit slope angle.

epatpk ← *the level at which the cone pattern ends on principle directions*

epatpj ← *the j coordinate at which the cone pattern ends on principle directions*

$$\text{percentage}_l = \frac{CMP_l}{\text{radius}_l} \times 100 \text{ for } l = 1 \text{ to } 8 \quad (\text{A.4})$$

for l ← *spatpk to 8 do*

for j ← 2 to 8 do

if $\text{radius}_l \times 1.1 \geq CMP_j$

*candidate*_{*l*} ← *j*

*calculate percentage*_{*l*} *using equation (A.4)*

end for

end for

l ← $\min(1 - \text{percentage}_l)$ *for* $l = 1$ to 8

j ← *candidate*_{*l*}

epatpk ← *l*

epatpj ← *j*

STEP 3

As the cone is constructed from “*spatpj*” to “*epatpj*”, the level at which the first block is captured needs to be determined for each “*j*”.

prinlvl_j ← *the level at which the cone pattern captures the first block on j coordinate for principle directions*

for l ← *spatpk to epatpk do*

for j ← 2 to *epatpj* – 1 do

if $\text{radius}_l \times 1.1 \geq CMP_j$

$prinlvl_j \leftarrow l$

end for

end for

$prinlvl_1 \leftarrow spatpk$
 $prinlvl_{epatpj} \leftarrow epatpk$

STEP 4

Apply the steps from STEP 0 to STEP 3 for the diagonal directions and generate $dialvl_j$ array.

$epatdk \leftarrow$ *the level at which the cone pattern ends on diagonal directions*
 $epatdz \leftarrow$ *the z coordinate at which the cone pattern ends on diagonal directions*
 $spatdk \leftarrow$ *the level at which the cone pattern starts on diagonal directions*
 $spatdz \leftarrow$ *the z coordinate at which the cone pattern starts on diagonal directions*
 $dialvl_z \leftarrow$ *the level at which the cone pattern captures the first block on z coordinate for diagonal directions*

STEP 5

Once the arrays $prinlvl_j$ and $dialvl_j$ are determined, the boundary of the cone template is constructed. For the remaining directions within the cone boundary, the level when the blocks need to be extracted will be determined.

$lvl \leftarrow$ *level at which the target block is captured by cone pattern*
 $store \leftarrow$ *stores the i, j and level of the blocks captured by cone pattern*
 $CM \leftarrow$ *distance from the center of mass of a block to a base block*

$$CM \leftarrow \sqrt{(ixwidth)^2 + (jxwidth)^2} \tag{A.5}$$

if $epatpk > epatdk$
 for $i \leftarrow 1$ *to* $epatpj$ *do*
 for $j \leftarrow 1$ *to* $epatpj$ *do*
 calculate CM using equation (A.5)
 $lvl \leftarrow CM/radius_1$
 $lvl \leftarrow$ *Round Up*
 if $lvl \leq epatpk$
 $store \leftarrow i, j$ *and* lvl

```

        end for
    end for

else
    for i ← 1 to epatdz do
        for j ← 1 to epatdz do
            calculate CM using equation (A.5)
            lvl ← CM/radius1
            lvl ← Round Up
            if lvl ≤ epatdk
                store ← i, j and lvl
            end for
        end for
    end for

```

Once the steps of the cone pattern generation algorithm are completed, the pattern on the first quadrant is reflected over “ i ” and “ j ” axis to complete the cone template. The steps of the algorithm will be illustrated with an example below.

Example

Assume a block model with a block size of 50x50x35ft. The cone pattern template will be generated for 40° pit slope angle.

STEP 0

Calculate $radius_l$ for $l = 1$ to 8 by using equation (A.1)

Calculate CMP_j for $j = 1$ to 8 by using equation (A.2)

Calculate CMD_z for $z = 1$ to 8 by using equation (A.3)

Table A.1: Summary of the cone radius and the distance from the center of the mass of a block calculated for the principal and diagonal directions for 8 increments

Levels	$radius_l$	CMP_j	CMD_z
1	41.7	50	70.7
2	83.4	100	141.4
3	125.2	150	212.1
4	166.9	200	282.8
5	208.6	250	353.6
6	250.3	300	424.3
7	292.0	350	494.9
8	333.7	400	565.7

STEP 1

The pattern always starts at $j=1$ and $z=1$.

$$spatpj = 1$$

$$spatdz = 1$$

According to the table A.1:

$$1.1radius_1 \leq CMP_1$$

$$1.1radius_2 \geq CMP_1 \quad \text{level 2 is selected to start the pattern for principle direction}$$

$$spatpk = 2$$

$$1.1radius_2 \geq CMD_1 \quad \text{level 2 is selected to start the pattern for diagonal direction}$$

$$spatdk = 2$$

STEP 2

Identify the candidate end pattern coordinates and levels for the principal and diagonal directions.

Principal Direction

The levels where the cone radius becomes greater than the distance from the center of mass of the base block are illustrated in Table A.2. The percentage deviations of CMP_j from the $radius_l$ are shown in Table A.3. The candidate with the minimum deviation is selected as an end pattern.

$candidate_3 = 2, candidate_4 = 3, candidate_5 = 4$
 $candidate_6 = 5, candidate_7 = 6, candidate_8 = 7$

Table A.2: Candidates for the end pattern levels and j locations on principal direction shown with similar colors

Increments	1. $1 \times radius_l$	CMP_j
2	91.8	100
3	137.7	150
4	183.6	200
5	229.4	250
6	275.3	300
7	321.2	350
8	367.1	400

Table A.3: The best candidate end pattern selected on principal direction based on the minimum percent deviation

Candidates	Percent Deviations (%)
3	20.1
4	10.1
5	4.1
6	0.12
7	2.7
8	4.9

$candidate_6 = 5$ has the minimum percent deviation

$epatpk = 6$

$epatpj = 5$

Diagonal Direction

The same steps for generating the candidate patterns for the principal direction will be followed.

$candidate_4 = 2, candidate_5 = 3$

$candidate_7 = 4, candidate_8 = 5$

Table A.4: Candidates for the end pattern levels and z locations on diagonal direction shown with similar colors

Increments	1. $1xradius_l$	CMD_z
2	91.8	141.4
3	137.7	212.1
4	183.6	282.8
5	229.4	353.6
6	275.3	424.3
7	321.2	494.9
8	367.1	565.7

Table A.5: The best candidate end pattern selected on diagonal direction based on the minimum percent deviation

Candidates	Percent Deviations (%)
4	15.2
5	1.7
7	3.1
8	5.9

$candidate_5 = 3$ has the minimum percent deviation

$epatdk = 5$

$epatdz = 3$

STEP 3

The start and end patterns of the cone for principal and diagonal directions are determined. For the principal directions, as the cone is constructed from “ $spatpj$ ” to “ $epatpj$ ”, the level at which the first block is captured needs to be determined for each “ j ”. For the diagonal directions, the cone is constructed from “ $spatpz$ ” to “ $epatpz$ ”, hence the level at which the first block is captured will be determined for each “ z ”.

Principal Direction

The calculations are performed for each level to determine the level and j location of the first block included in the cone template and presented in Table A.6.

Table A.6: The block levels and j locations on principal direction shown with similar colors

Increments	1. $1xradius_l$	CMP_j
2	91.8	100
3	137.7	150
4	183.6	200
5	229.4	250

$prinlvl_1 = 2, prinlvl_2 = 3, prinlvl_3 = 4$

$prinlvl_4 = 5, prinlvl_5 = 6,$

Diagonal Direction

This time, the level and z location of the first block included in the cone template are determined and presented in Table A.7.

Table A.7: The block levels and z locations on diagonal direction shown with similar colors

Increments	1. $1xradius_l$	CMD_z
2	91.8	141.4
3	137.7	212.1
4	183.6	282.8

$$dialvl_1 = 2, \quad dialvl_2 = 4, \quad dialvl_3 = 5$$

So far, the patterns that governs the cone boundary are accomplished as it is demonstrated in Figure A.5. The next step will seek the levels where the first block is captured for the blue shaded region. The white shaded region will not be considered since it is on the diagonal direction where the start and end patterns are accomplished.

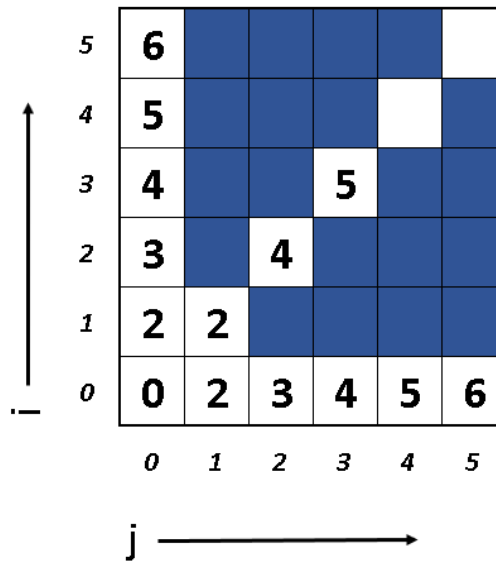


Figure A.5: Cone template generated for principal and diagonal directions

STEP 4

STEP 1 to STEP 3 included the calculations for the diagonal direction, therefore STEP 4 will be skipped and proceed with STEP 5 of the algorithm.

CM is calculated by using equation (A.5) for each one of the remaining blocks within the cone boundary and presented in Table A.8.

Table A.8: The blocks that are included in the cone template colored with blue

Coordinates	<i>CM</i>	<i>CM/radius₁</i>
(1,2)	111.8	3
(1,3)	158.1	4
(1,4)	206.2	5
(1,5)	254.9	7
(2,1)	111.8	3
(2,3)	180.3	5
(2,4)	223.6	6
(2,5)	269.3	7
(3,1)	158.1	4
(3,2)	180.3	5
(3,4)	250	6
(3,5)	291.6	7
(4,1)	206.2	5
(4,2)	223.6	6
(4,3)	250	6
(4,5)	320.2	8
(5,1)	254.9	7
(5,2)	269.3	7
(5,3)	291.6	7
(5,4)	320.2	8

The cone template reached to a maximum level 6 on principal direction ($prinlvl_5$), which is higher than the level attained on diagonal direction ($dialvl_3$). Hence, any block that has higher level will not be included in the cone template. The final outline of the cone template is shown in Figure A.6. This template only reflects the cone pattern on the first quadrant of the Cartesian coordinate system. Reflecting the pattern over j and i axis will generate the full cone template representing all four quadrants as shown in Figure A.7.

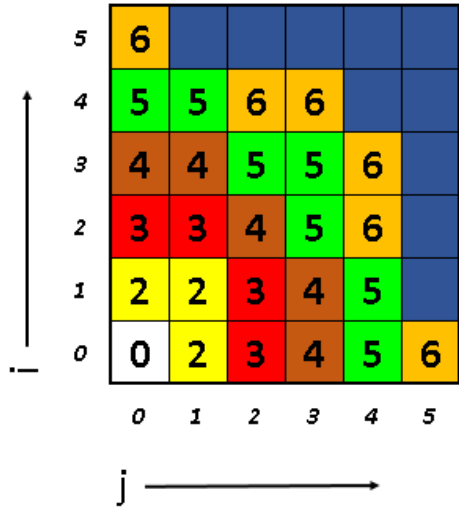


Figure A.6: Cone template generated for the 1st Quadrant



Figure A.7: Final cone pattern template for 40° pit slope angle, 50x50x35ft blocks

The generated cone pattern template can be used to generate an ultimate pit or run the proposed production scheduling solution algorithm for a pit with a single 40° slope angle and 50x50x35ft block dimensions. Based on the location of the block in the pit, this cone pattern will generate at most 80 arcs from a single block to the preceding blocks to create the dependencies that will reflect

the 40° slope angle. The maximum elevation of the search pattern is level 6, which means that the first 5 benches will have even less than 80 arcs connecting the target blocks. The block size plays a significant role to determine the extent of the cone pattern. If the cone pattern is generated with 20x20x15ft block dimensions for the same 40° slope angle, the cone template would look like the one demonstrated in Figure A.8.

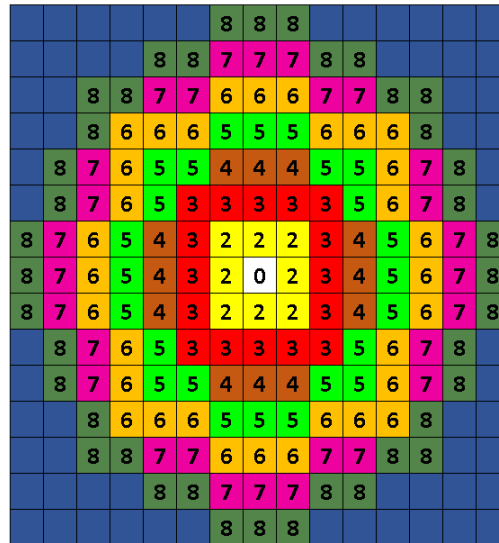


Figure A.8: Final cone pattern template for 40° pit slope angle, 20x20x15ft blocks

Another important factor that governs the pattern stopping criteria is the minimum deviation of the radius from the center of mass of the block closest to the side of the cone. Since the blocks are discrete objects, the side of the cone will assume to pass from the center of the block which may shorten or extend the radius of the cone representing the true slope angle. The proposed algorithm will select the level at which the center of the block is closest to the side of the cone. If the actual slope angle of the cone template for 50x50x35ft block size is measured for principal and diagonal directions, it is found that on principal directions the actual slope is 40° and on diagonal directions the actual slope is 39.5°. For 20x20x15ft block size, the actual slope is 40.6° and on diagonal directions the actual slope is 40.3°.

A.2.1 Complex Slope Angles Based on Geotechnical Zone

The pit may consist of a set of geotechnical domains where each domain inheres a unique behavior which will lead to a complex slope design process. These domains can also be named

zones. A medium size open pit mine can have couple dozen geotechnical zones. The zones may be altered both horizontally and vertically. The geotechnical properties of each zone will determine the steepest slope angle at which mining can occur without risking pit slope failure. Depending on the behavior of the rock type, a medium scale open pit mine may have slope angles varying from 30° to 58° from one zone to another. The schedules generated by mine production plans must honor these complex slope angles on a block level. If the complex slope angles are not reflected precisely in the production plans, the resulting NPVs may be either underestimated or overestimated depending on how shallow or deep the pit slopes are generated. So far, it hasn't been shown in the published literature how to integrate the vertical alteration of the slope zones into the cone generation techniques.

As was mentioned before, the dependencies between the blocks are formed by connecting the target block by arcs to each one of the preceding blocks. The arcs must impose the variation of the slope angles from one zone to another. It is important to keep the number of arcs formed as low as possible in order to minimize the computing time and the memory usage. The proposed cone pattern generation algorithm aims to minimize the number of the arcs generated as well as the deviations from the required slope angles.

The geotechnical properties of each zone may lead to a simplification of the possible slope angles within a domain to a unique slope angle. In other words, slope angle will be constant for all directions within the domain. This interpretation empowers the use of cone pattern generating algorithm proposed for the constant pit slope angles. If an open pit mine consists of n number of domains with n number of distinct slope angles, then n number of different cone pattern templates will be generated which means that each domain will have its own cone pattern template. The arcs will connect the base block to the dependent block addressed by the cone pattern template developed for the domain of the base block. If the dependent block does not share the same domain with the base block, then the arc generation stops. When the dependent block becomes the new base block, the arc generation will continue by using a new cone pattern template reflecting the domain of the new base block. The proposed technique can handle any number of geologic domains.

A.3 Cone Pattern Generation for Complex Slopes Based on Multiple Azimuths

The cone pattern generation algorithm presented in Appendix A.2 will be extended to incorporate slope angles based on multiple azimuths. One of the main problems with the previous methods is the linear interpolation techniques used for smoothing the pit walls from one azimuth direction to another resulted in polygonal shaped pits with sharp intersections. This problem can be addressed by utilizing non-linear interpolation techniques. Gilani and Sattarvand (2015) proposed a non-linear interpolation method to smooth the pit walls which is also motivated this research in order to generate cone pattern templates for multi-azimuth directions. The non-linear interpolation method benefits from the inverse distance law. However, the implementation of the non-linear interpolation technique is different than the original since the goal is to minimize the number of arcs generated while obeying the slope angles.

A cone template with various azimuths can be shown in Figure A.9, where the azimuth directions are represented with red colored arrows and the principal and diagonal directions are shown with black colored arrows. There may be different pit slope requirements on each azimuth direction which emphasizes the necessity of pit wall smoothing. The non-linear interpolation technique will take the radius on each azimuth as input and interpolate the radius on the principal and diagonal directions. This will lead to a pattern generation on eight directions for at most eight levels. The pattern generation technique will follow similar steps as before. In addition to that, the variables will have one more dimension to incorporate eight directions which was not required before since the slopes were constant for all directions.

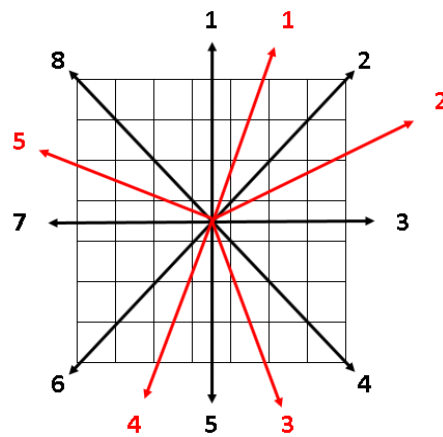


Figure A.9: Cone template with principal, diagonal and azimuth directions

Given i, j, k coordinate system, “ i ” represents the northing, “ j ” represents easting and k represents the elevation. For the diagonal directions, “ z ” axis which is located on the “ i-j ” plane will be used. The algorithm steps will show how to generate patterns on both principal and diagonal directions.

Let`s assume the base block b_{ijk} is located on coordinates $i=0, j=0$ and $k=0$.

STEP 0

Calculate the radius of the cone for each one of the azimuth directions.

Calculate the horizontal distance from the center of the base block for 8 levels on principal and diagonal directions.

$N \leftarrow$ number of azimuths defined

$slope_a \leftarrow$ required pit slope angle on azimuth a

$radius_a \leftarrow$ radius of the cone on azimuth a

$CMP_j \leftarrow$ horizontal distance from the center of the base block on principle direction

$CMD_z \leftarrow$ horizontal distance from the center of the base block on diagonal direction

$$radius_a = \frac{\text{block height}}{\tan(\text{slope}_a)} \text{ for } a = 1 \text{ to } N \quad (\text{A.6})$$

$$CMP_j = j \times \text{block width} \text{ for } j = 1 \text{ to } 8 \quad (\text{A.7})$$

$$CMD_z = z \times \text{block width} \times \sqrt{2} \text{ for } z = 1 \text{ to } 8 \quad (\text{A.8})$$

STEP 1

Non-linear interpolation technique is used to interpolate the radius for all the principal and diagonal directions. First, the directions are numbered every 45° starting from 1 at North (0°) to 8 at Northwest (315°). Each direction will have a predecessor and successor azimuths as shown in Figure A.10. The following equation will be used to calculate the radius for each one of the 8 directions.

$rad_{l,d} \leftarrow$ the radius for direction d on level l

$$rad_{l,d} = \frac{\beta^2}{\beta^2 + \alpha^2} \times radius_p + \frac{\alpha^2}{\beta^2 + \alpha^2} \times radius_s \quad (\text{A.9})$$

```

for d ← 1 to 8 do
    identify predecessor and successor azimuths for direction d
    calculate  $rad_{1,d}$  by using eqn (A.9)
end for
 $rad_{l,d} = rad_{1,d} \times l$  for  $l = 1$  to 8,  $\forall d$ 

```

(A.10)

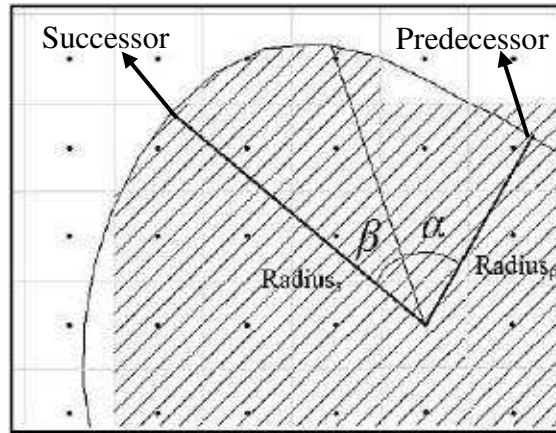


Figure A.10: Parameters needed for non-linear interpolation (Gilani and Sattarvand, 2015)

STEP 2

The starting pattern of the cone will be calculated for 8 directions.

$spatpk_d \leftarrow$ the level at which the cone pattern starts on principle direction d

$spatpj_d \leftarrow$ the j coordinate at which the cone pattern starts on principle direction d

$spatdk_d \leftarrow$ the level at which the cone pattern starts on diagonal direction d

$spatdz_d \leftarrow$ the z coordinate at which the cone pattern starts on diagonal direction d

Principal Directions

```

for d ← 1 to 8 do
    for l ← 1 to 8 do
        if  $rad_{l,d} \times 1.1 \geq CMP_1$ 
             $spatpk_d \leftarrow l$ 
            exit for loop
        end for
    end for
end for

```

$d \leftarrow d + 2$
end for

Diagonal Directions

for $d \leftarrow 2$ to 8 do
 for $l \leftarrow 1$ to 8 do
 if $rad_{l,d} \times 1.1 \geq CMD_1$
 $spatdk_d \leftarrow l$
 exit for loop
 end for
 $d \leftarrow d + 2$

end for
 $spatpj_d \leftarrow 1$ for $d \leftarrow$ principle directions
 $spatdz_d \leftarrow 1$ for $d \leftarrow$ diagonal directions

STEP 3

Generate a set of candidate levels for the end pattern for each one of the 8 directions. Then, select the best candidate that will minimize the deviation from the required slope angle.

$epatpk_d \leftarrow$ the level at which the cone pattern ends on principle direction d
 $epatpj_d \leftarrow$ the j coordinate at which the cone pattern ends on principle direction d
 $epatdk_d \leftarrow$ the level at which the cone pattern ends on diagonal direction d
 $epatdz_d \leftarrow$ the z coordinate at which the cone pattern ends on diagonal direction d

$$percentage_{l,d} = \frac{CMP_l}{rad_{l,d}} \times 100 \text{ for } l = 1 \text{ to } 8, d \leftarrow \text{principle directions} \quad (A.11)$$

$$percentage_{l,d} = \frac{CMD_l}{rad_{l,d}} \times 100 \text{ for } l = 1 \text{ to } 8, d \leftarrow \text{diagonal directions} \quad (A.12)$$

Principal Directions

for $d \leftarrow 1$ to 8 do
 for $l \leftarrow spatpk_d$ to 8 do
 for $j \leftarrow 2$ to 8 do
 if $rad_{l,d} \times 1.1 \geq CMP_j$

$candidate_{d,l} \leftarrow j$
 calculate percentage using equation (A.11)

end for

end for

$d \leftarrow d + 2$

end for

Diagonal Directions

for $d \leftarrow 2$ *to* 8 *do*

for $l \leftarrow spatdk_d$ *to* 8 *do*

for $z \leftarrow 2$ *to* 8 *do*

if $rad_{l,d} \times 1.1 \geq CMD_z$

$candidate_{d,l} \leftarrow z$

calculate percentage using equation (A.12)

end for

end for

$d \leftarrow d + 2$

end for

Best Candidate for the End Pattern

for $d \leftarrow 1$ *to* 8 *do*

if $d \leftarrow$ principle direction

$epatpk_d \leftarrow \min(1 - percentage_{l,d})$ *for* $l = 1$ *to* 8

$epatpj_d \leftarrow candidate_{d,l}$

if $d \leftarrow$ diagonal direction

$epatdk_d \leftarrow \min(1 - percentage_{l,d})$ *for* $l = 1$ *to* 8

$epatdz_d \leftarrow candidate_{d,l}$

end for

STEP 4

The level at which the first block is captured needs to be determined for each “j” on principal, for each “z” on diagonal directions.

$prinlvl_{d,j} \leftarrow$ the level at which the cone pattern captures the first block on increment
j on principle directions

$dialvl_{d,z} \leftarrow$ the level at which the cone pattern captures the first block on increment
z on diagonal directions

Principal Directions

for $d \leftarrow 1$ to 8 do

for $l \leftarrow spatpk_d$ to $epatpk_d$ do

for $j \leftarrow 2$ to $epatpj_d - 1$ do

if $rad_{l,d}x1.1 \geq CMP_j$

$prinlvl_{d,j} \leftarrow l$

end for

end for

$prinlvl_{d,epatpj_d} \leftarrow epatpk_d$

$prinlvl_{d,1} \leftarrow spatpk_d$

$d \leftarrow d + 2$

end for

Diagonal Directions

for $d \leftarrow 2$ to 8 do

for $l \leftarrow spatdk_d$ to $epatdk_d$ do

for $z \leftarrow 2$ to $epatdz_d - 1$ do

if $rad_{l,d}x1.1 \geq CMD_z$

$dialvl_{d,z} \leftarrow l$

end for

end for

$dialvl_{d,epatdz_d} \leftarrow epatdk_d$

$dialvl_{d,1} \leftarrow spatdk_d$

$$d \leftarrow d + 2$$

end for

STEP 5

Once the arrays $prinlvl_{d,j}$ and $dialvl_{d,z}$ are determined, the boundary of the cone template is constructed. For the remaining directions within the cone boundary, the level when the blocks need to be extracted will be determined.

$lvl \leftarrow$ level at which the target block is captured by cone pattern

$store \leftarrow$ stores the i, j and level of the blocks captured by cone pattern

$CM \leftarrow$ distance from the center of mass of a block to a base block

$$CM \leftarrow \sqrt{(ixwidth)^2 + (j * width)^2} \quad (A.13)$$

$Angle \leftarrow$ azimuth of the blocks in the cone template

$$Angle \leftarrow \arctan(j/i) \quad (A.14)$$

$nrad \leftarrow$ interpolated radius for the remaining directions in the cone template

1st Quadrant of the Coordinate System

if $epatdk_2 > epatpk_1$ AND $epatdk_2 > epatpk_3$

for $i \leftarrow 1$ to $epatdz_2$ do

for $j \leftarrow 1$ to $epatdz_2$ do

calculate CM using equation (A.13)

calculate $Angle$ using equation (A.14)

if $Angle < 45$

$$nrad = \frac{(Angle)^2}{(Angle)^2 + (45 - Angle)^2} xradius_2 + \frac{(45 - Angle)^2}{(Angle)^2 + (45 - Angle)^2} xradius_1$$

if $Angle > 45$

$$nrad = \frac{(Angle - 45)^2}{(Angle - 45)^2 + (90 - Angle)^2} xradius_3 + \frac{(90 - Angle)^2}{(Angle - 45)^2 + (90 - Angle)^2} xradius_2$$

$lvl \leftarrow CM/nrad$

$lvl \leftarrow$ Round Up

if $lvl \leq epatdk_2$

store $\leftarrow i, j, lvl$ and d

```

        end for
    end for

else
    for i ← 1 to epatpj1 do
        for j ← 1 to epatpj3 do
            calculate CM using equation (A.13)
            calculate Angle using equation (A.14)
            if Angle < 45
                
$$nrad = \frac{(Angle)^2}{(Angle)^2 + (45 - Angle)^2} xradius_2 + \frac{(45 - Angle)^2}{(Angle)^2 + (45 - Angle)^2} xradius_1$$

            if Angle > 45
                
$$nrad = \frac{(Angle - 45)^2}{(Angle - 45)^2 + (90 - Angle)^2} xradius_3 + \frac{(90 - Angle)^2}{(Angle - 45)^2 + (90 - Angle)^2} xradius_2$$

            lvl ← CM/nrad
            lvl ← Round Up
            if lvl ≤ epatpk1 OR lvl ≤ epatpk3
                store ← i, j, lvl and d
            end for
        end for
    end for

```

By making the appropriate variable substitutions, the remaining quadrants can be also determined similarly to complete the cone pattern template.

Example

The below example will demonstrate the implementation of the cone pattern generation technique for complex slopes based on multiple azimuths where only the first quadrant will be considered. Assume a block model with a block size of 50x50x35ft. Figure A.11 shows the azimuths and slope angles on given regions. Also, black colored lines show the principal direction 1 (N-S), diagonal direction 2 (NE-SW) and principal direction 3 (E-W).

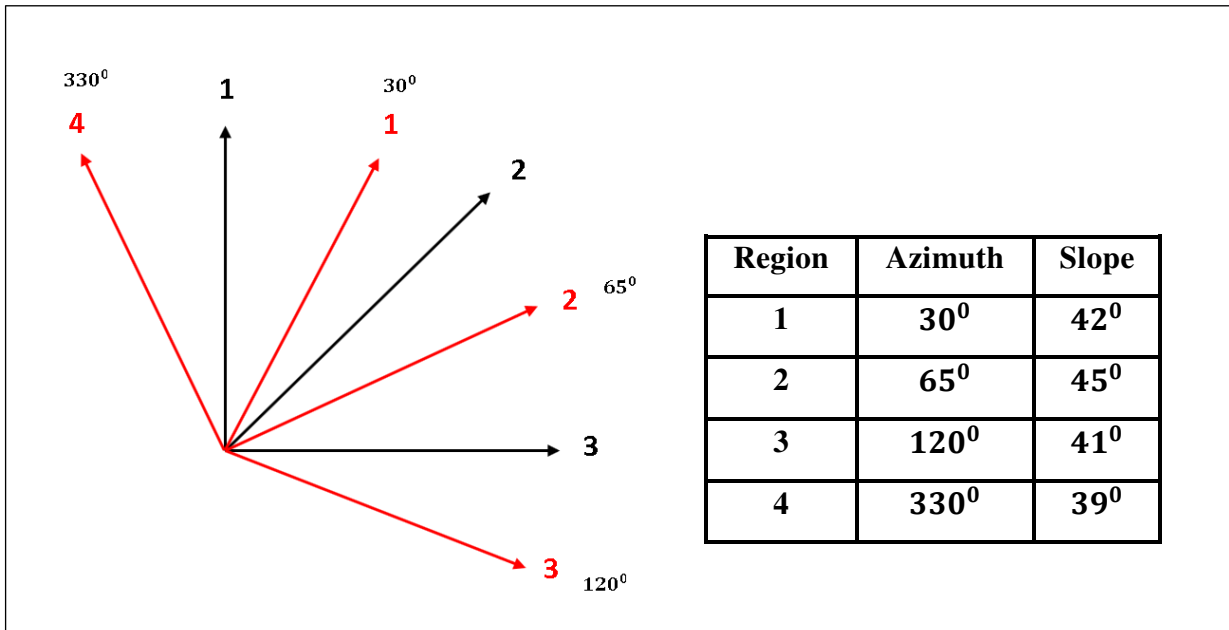


Figure A.11: Azimuths and slope angles given for each region on the first quadrant. Azimuths shown with red, principal and diagonal directions shown with black color

STEP 0

Number of azimuths = 4

Calculate $radius_a$ for $a = 1$ to 4 by using equation (A.6)

Calculate CMP_j for $j = 1$ to 8 by using equation (A.7)

Calculate CMD_z for $z = 1$ to 8 by using equation (A.8)

Table A.9: Cone radius calculated for each region

Region	$radius_a$
1	35
2	32.6
3	29.4
4	37.5

Table A.10: Summary of the distance from the center of the mass of a block calculated for the principal and diagonal directions for 8 increments

Levels	CMP_j	CMD_z
1	50	70.7
2	100	141.4
3	150	212.1
4	200	282.8
5	250	353.6
6	300	424.3
7	350	494.9
8	400	565.7

STEP 1

The non-linear interpolation technique shown in equation (A.9) will be used to interpolate the radius for principal directions 1 and 3 and diagonal direction 2.

Figure A.12 shows the successor and principal azimuths for each principal and diagonal direction. Let's demonstrate the calculation of the radius for the principal direction 1 for level 1. We know that the angle between principal direction 1 and the azimuth 4 is $\beta = 30^\circ$ and the angle between principal direction 1 and the azimuth 1 is $\alpha = 30^\circ$ as shown in Figure A.12. By using equation (A.9):

$$rad_{1,1} = \frac{30^2}{30^2 + 30^2} \times 35 + \frac{30^2}{30^2 + 30^2} \times 37.5 = 36.3$$

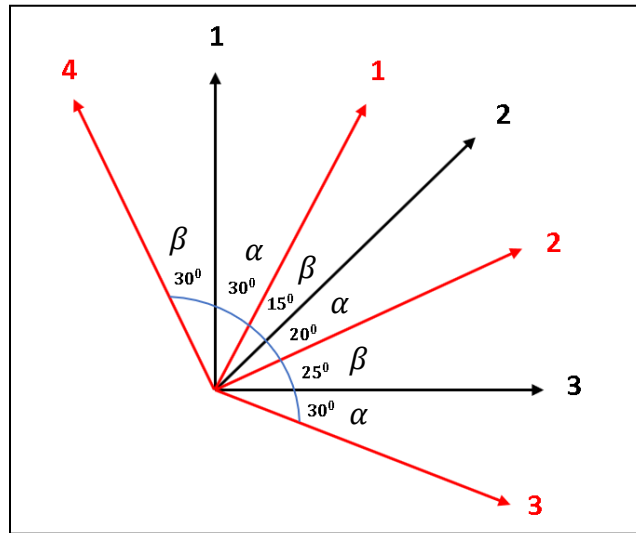


Figure A.12: Successor and predecessor azimuths with α and β angles

The radiuses are interpolated for each direction and each level similarly by using equation (A.9) and presented in Table A.11.

Table A.11: Summary of the radius calculated for each level on each principal and diagonal direction

Levels	Radius		
	Direction 1	Direction 2	Direction 3
1	36.3	36.6	31.3
2	72.5	73.3	62.6
3	108.8	109.9	93.9
4	145.1	146.6	125.2
5	181.3	183.2	156.5
6	217.6	219.8	187.8
7	253.9	256.5	219.1
8	290.1	293.1	250.4

STEP 2

The pattern always starts at $j=1$ and $z=1$.

$spatpj_1 = 1$ for principal direction 1

$spatdz_2 = 1$ for diagonal direction 2

$spatpj_3 = 1$ for principal direction 3

According to the table A.12:

For principal direction 1 and 3

$1.1xrad_{1,1} \leq CMP_1$

$1.1xrad_{2,1} \geq CMP_1$ level 2 is selected to start the pattern for principle direction 1

$1.1xrad_{1,3} \leq CMP_1$

$1.1xrad_{2,3} \geq CMP_1$ level 2 is selected to start the pattern for principle direction 3

$spatpk_1 = 2$

$spatpk_3 = 2$

For diagonal direction 2

$1.1xrad_{2,2} \geq CMD_1$ level 2 is selected to start the pattern for diagonal direction 2

$spatdk_2 = 2$

Table A.12: Start pattern level selected for each direction. Yellow color shows the radius on the principal directions greater than CMP_j , blue color shows the radius on the diagonal directions greater than CMD_z

Increments	Radius			CMP_j	CMD_z
	Direction 1	Direction 2	Direction 3		
1	36.27	36.64	31.30	50	70.7
2	72.53	73.28	62.60	100	141.4
3	108.80	109.92	93.89	150	212.1

STEP 3

The levels where the cone radius becomes greater than the distance from the center of mass of the base block for principal directions 1 and 3 and diagonal direction 2 are illustrated in Table A.13.

Table A.13: Candidates for the end pattern levels and j locations on principal and diagonal directions shown with similar colors

Levels	<i>radiusx1.1</i>	CMP_j	<i>radiusx1.1</i>	CMD_z	<i>radiusx1.1</i>	CMP_j
	Direction 1		Direction 2		Direction 3	
1	38.9	50	40.3	70.7	34.4	50
2	79.8	100	80.6	141.4	68.9	100
3	119.7	150	120.9	212.1	103.3	150
4	159.6	200	161.2	282.8	137.7	200
5	199.5	250	201.5	353.6	172.1	250
6	239.4	300	241.8	424.3	206.6	300
7	279.3	350	282.1	494.9	241	350
8	319.1	400	322.4	565.7	275.4	400

The candidates are:

Principal direction 1:

$$candidate_{1,2} = 1, candidate_{1,3} = 2, candidate_{1,4} = 3$$

$$candidate_{1,6} = 4, candidate_{1,7} = 5, candidate_{1,8} = 6$$

Diagonal direction 2:

$$candidate_{2,2} = 1, candidate_{2,4} = 2$$

$$candidate_{2,6} = 3, candidate_{2,8} = 4$$

Principal direction 3:

$$candidate_{3,2} = 1, candidate_{3,3} = 2, candidate_{3,5} = 3$$

$$candidate_{3,6} = 4, candidate_{3,8} = 5$$

The percentage deviations of CMP_j and CMD_z from the cone radius on principal and diagonal directions are shown in Table A.14. The candidates with the minimum deviation is selected as an end pattern. For the diagonal direction, the deviations are equal, so any candidate can be selected.

Table A.14: The best candidate end pattern selected on principal and diagonal directions based on the minimum percent deviation

Direction 1		Direction 2		Direction 3	
Candidates	Deviation (%)	Candidates	Deviation (%)	Candidates	Deviation (%)
2	31.1	2	3.5	2	20.1
3	8.1	4	3.5	3	6.1
4	3.3	6	3.5	5	4.2
6	8.1	8	3.5	6	6.1
7	1.5			8	0.2
8	3.3				

For principal direction 1:

$$candidate_{1,7} = 5 \quad \text{where} \quad epatpk_1 = 7 \quad \text{and} \quad epatpj_1 = 5$$

For diagonal direction 2:

$$candidate_{2,8} = 4 \quad \text{where} \quad epatdk_2 = 8 \quad \text{and} \quad epatdz_1 = 4$$

For principal direction 3:

$$candidate_{3,8} = 5 \quad \text{where} \quad epatpk_3 = 8 \quad \text{and} \quad epatpj_3 = 5$$

STEP 4

The start and end patterns of the cone for principal and diagonal directions are determined. As the cone is constructed from the start pattern coordinate ($spatpj_d$) to end pattern coordinate ($epatpj_d$) for the principal directions, and from the start pattern coordinate ($spatdz_d$) to end pattern coordinate ($epatdz_d$) for the diagonal directions, the level at which the first block is captured needs to be determined for each “j” and “z”. Figure A.13 highlights the levels for each direction on the coordinate based on CMP_j and CMD_z with the same color.

Increments	Direction 1	CMP_j
2	79.8	100
3	119.7	150
4	159.6	200
5	199.5	250
6	239.4	300

Increments	Direction 2	CMD_z
2	80.6	141.4
3	120.9	212.1
4	161.2	282.8
5	201.5	353.6
6	241.8	424.3

Increments	Direction 3	CMP_j
2	68.9	100
3	103.3	150
4	137.7	200
5	172.1	250
6	206.6	300

Figure A.13: The block levels and j locations on principal direction and z locations on diagonal direction shown with similar colors

For principal direction 1:

$$princlvl_{1,1} = 2, \quad prinl_{1,2} = 3, \quad prinl_{1,3} = 4$$

$$princlvl_{1,4} = 6, \quad prinl_{1,5} = 7$$

For diagonal direction 2:

$$dialvl_{2,1} = 2, \quad dialvl_{2,2} = 4$$

$$dialvl_{2,3} = 6, \quad dialvl_{2,4} = 8$$

For principal direction 3:

$$princlvl_{3,1} = 2, \quad prinl_{3,2} = 3, \quad prinl_{3,3} = 5$$

$$princlvl_{3,4} = 6, \quad prinl_{3,5} = 8$$

So far, the cone boundary governed by the cone pattern is accomplished and demonstrated in Figure A.14. The white shaded region will not be considered since it is beyond the end pattern on the diagonal direction. The next step will seek the levels where the first block is captured for the blue shaded region.

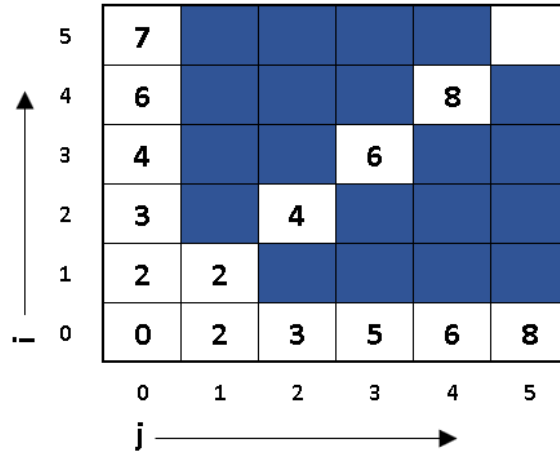


Figure A.14: Cone template generated for principal and diagonal directions

STEP 5

This time the non-linear interpolation technique will be used to calculate the radius passing from the midpoint of the blocks on the blue shaded region shown in Figure A.14. In the following, a calculation for the block that has a coordinate of $i = 1$ and $j = 2$ will be demonstrated.

Figure A.15 shows that the successor radius is on diagonal direction 2 with $\beta = 18.5^\circ$ and the predecessor radius is on principal direction 3 with $\alpha = 26.5^\circ$. By using equation (A.9):

$$nrad_{1,2} = \frac{18.5^2}{18.5^2 + 26.5^2} \times 31.3 + \frac{26.5^2}{18.5^2 + 26.5^2} \times 36.6 = 34.9$$

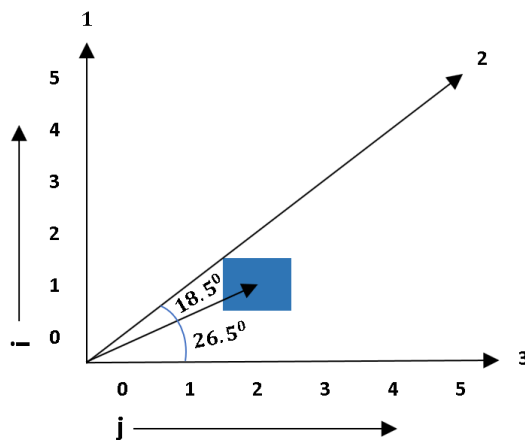


Figure A.15: Successor and predecessor radius for coordinate (1,2) with β and α angles.

The rest of the blue shaded region shown in Figure A.14 will be calculated similarly and the summary of the results are shown in Table A.15. Since the highest level reached is 8, any level above 8 which is colored white will not be part of the cone pattern template.

Table A.15: The blocks that are included in the cone template colored with blue

Coordinates	nrad	CM	CM/nrad
(1,2)	34.9	111.8	4
(1,3)	33	158.1	5
(1,4)	32.2	206.2	7
(1,5)	31.8	254.9	8
(2,1)	36.5	111.8	4
(2,3)	6.1	180.3	5
(2,4)	34.9	223.6	7
(2,5)	33.8	269.3	8
(3,1)	36.4	158.1	5
(3,2)	36.6	180.3	5
(3,4)	36.4	250	7
(3,5)	35.7	291.6	9
(4,1)	36.3	206.2	6
(4,2)	36.5	223.6	7
(4,3)	36.6	250	7
(4,5)	36.5	320.2	9
(5,1)	36.3	254.9	7
(5,2)	36.4	269.3	8
(5,3)	36.6	291.6	8
(5,4)	36.6	320.2	9

The final outline of the cone template derived for the first quadrant is shown in Figure A.16.

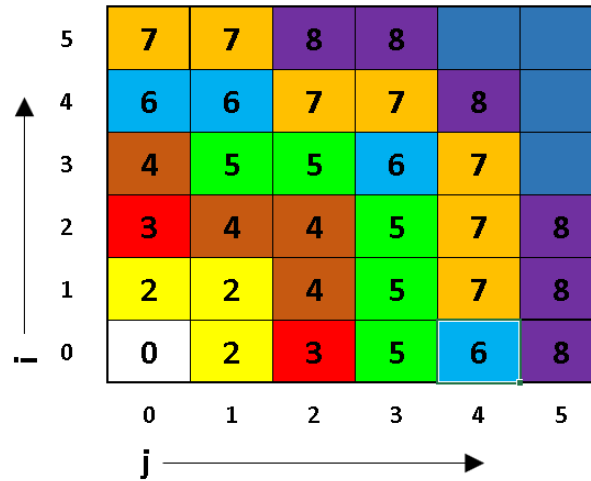


Figure A.16: Cone template generated for the 1st Quadrant

The proposed cone pattern generation algorithm can handle any number of azimuths with any size of block dimensions. The pattern is aimed to stop on such a level that the difference between the distance from the center of a mass of the base block to the side of the cone and the cone radius on that level is minimum. This property will enable the algorithm to generate slope angles closest to the actual slope angle on a given direction. If the slope angles change from one direction to another, the non-linear interpolation technique will generate smoothed pit walls. The size of the cone template brings another key advantage to the mine production scheduling problems by preventing the generation of the redundant arcs. The proposed cone pattern algorithm adds a strength to the goal of this thesis by enabling the proposed mine production solution algorithm to work with fast solution times as well as reflecting the actual pit slopes on a block level.