



UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA DE MINAS
ESCUELA PROFESIONAL DE INGENIERÍA DE MINAS



**IMPLEMENTACIÓN EN PYTHON DEL ALGORITMO DE
LERCHS Y GROSSMANN PARA LA OPTIMIZACIÓN DEL PIT
FINAL EN MINAS A CIELO ABIERTO EN YACIMIENTOS DE
COBRE TIPO SKARN**

TESIS

PRESENTADA POR:

Bach. YOVANI ACHATA FLORES

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO DE MINAS

PUNO – PERÚ

2023



NOMBRE DEL TRABAJO

IMPLEMENTACIÓN EN PYTHON DEL ALGORITMO DE LERCHS Y GROSSMANN PARA LA OPTIMIZACIÓN DEL PIT FINAL EN M

AUTOR

Yovani Achata Flores

RECuento DE PALABRAS

26028 Words

RECuento DE CARACTERES

132748 Characters

RECuento DE PÁGINAS

126 Pages

TAMAÑO DEL ARCHIVO

4.0MB

FECHA DE ENTREGA

Sep 12, 2023 6:00 AM GMT-5

FECHA DEL INFORME

Sep 12, 2023 6:02 AM GMT-5

● **5% de similitud general**

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos

- 4% Base de datos de Internet
- Base de datos de Crossref
- 2% Base de datos de trabajos entregados
- 2% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● **Excluir del Reporte de Similitud**

- Material bibliográfico
- Material citado
- Bloques de texto excluidos manualmente
- Material citado
- Coincidencia baja (menos de 10 palabras)

Universidad
Nacional
del Altiplano 
Firmado digitalmente por DURANT
BRODEN Jorge Gabriel FAU
20145496170 hard
Motivo: Doy V° B°
Fecha: 12.09.2023 11:27:19 -05:00

Universidad
Nacional
del Altiplano 

Firmado digitalmente por SALAS
URVICLA Fernando Benigno FAU
20145496170 soft
Motivo: Soy el autor del documento
Fecha: 12.09.2023 12:45:46 -05:00



DEDICATORIA

Querido padre y querida madre, Humberto Achata Mamani y Francisca Avelina Flores Alata, respectivamente; no puedo encontrar las palabras adecuadas para agradecer todo lo que han hecho por mí a lo largo de mi vida, su sacrificio, persistencia, apoyo incondicional y amor han sido fundamentales para mi formación personal y profesional. Gracias por motivarme, por ser ejemplo de fortaleza y por estar siempre a mi lado, incluso en los momentos más difíciles. Esta tesis es solo una pequeña muestra de todo lo que he logrado gracias a Ustedes. Dedico con todo mi corazón y mente este logro, en reconocimiento a su infinito amor y dedicación.

También quiero agradecer a mis maestros Ing. Ronald Benito López, quien me inculco la pasión por la informática en minería y Ing. Marco Echenique Medina, quien me enseñó Operaciones Mina, por su apoyo constante, por su motivación incondicional y por la confianza que siempre han tenido en mí. Gracias por ayudarme a alcanzar mis objetivos, por inculcarme valores importantes y por enseñarme a insistir y persistir en todo momento.

Por último, quiero dedicar esta tesis a mis maestros de la universidad: Mrs. Aza Gates, Arturo Rafael Chayña y Jorge Durant Broden, quienes me inspiraron a estudiar. Agradezco su motivación, disciplina, los cuales me ayudaron a formar una base sólida en mi formación académica y personal. Sin el apoyo y la motivación de todas estas personas especiales, este logro no hubiera sido posible. Por eso, les dedico con todo mi corazón esta tesis.

Yovani Achata Flores,



AGRADECIMIENTOS

Quiero expresar mi profundo agradecimiento a Dios por haberme brindado la oportunidad de estudiar en la Universidad y por haberme bendecido con el éxito académico.

También quisiera agradecer a la Universidad Nacional del Altiplano y a la Facultad de Ingeniería de Minas por su excelente educación y por brindarme las herramientas necesarias para mi formación profesional.

Agradezco a mis maestros de la universidad: Aza Gates, Arturo Rafael Chayña y Durant Broden, quienes me inspiraron a estudiar. Agradezco su motivación, disciplina, los cuales me ayudaron a formar una base sólida en mi formación académica y personal. Sin el amor, el apoyo y la motivación de todas estas personas especiales, este logro no hubiera sido posible. Por eso, les dedico con todo mi corazón esta tesis.

Por último, quiero expresar mi gratitud a la Universidad Nacional del Altiplano (VRIUNAP) y al MINEDU por su valioso financiamiento en el desarrollo de este proyecto.

Gracias por hacer posible la realización de este trabajo.

Yovani Achata Flores,



ÍNDICE GENERAL

DEDICATORIA

AGRADECIMIENTOS

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

ÍNDICE DE ACRÓNIMOS

RESUMEN 14

ABSTRACT..... 15

CAPÍTULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA..... 16

1.2. FORMULACIÓN DEL PROBLEMA 17

1.2.1. Problema General 17

1.2.2. Problemas Específicos 17

1.3. HIPÓTESIS DE LA INVESTIGACIÓN 17

1.3.1. Hipótesis General..... 17

1.3.2. Hipótesis Específicas 17

1.4. JUSTIFICACIÓN DE LA INVESTIGACIÓN 18

1.5. OBJETIVOS DE LA INVESTIGACIÓN..... 19

1.5.1. Objetivo General..... 19

1.5.2. Objetivos Específicos 19

CAPÍTULO II

REVISION DE LITERATURA

2.1. ANTECEDENTES DE INVESTIGACIÓN..... 20



2.2. MARCO TEÓRICO	25
2.2.1. Planificación minera	25
2.2.1.1. La planificación minera a largo plazo	25
2.2.1.2. La planificación minera a mediano plazo	25
2.2.1.3. La planificación minera a corto plazo	26
2.2.2. Métodos de optimización de límites económicos en minas a cielo abierto	27
2.2.2.1. Algoritmo de cono flotante	28
2.2.2.2. Algoritmo de Korobov	38
2.2.2.3. Algoritmo de Lerchs y Grossmann	48
2.2.3. Determinación del pit final	57
2.2.4. Métodos de evaluación económica	59
2.2.4.1. Valor actual neto	60
2.2.4.2. Tasa interna de retorno	60
2.2.4.3. Periodo de retorno de la inversión	61
2.2.5. Diseño de taludes en minería a cielo abierto	62

CAPÍTULO III

MATERIALES Y METODOS

3.1. MATERIAL DE ESTUDIO	66
3.1.1. Modelo de Bloques	66
3.1.2. Valorización del Modelo	70
3.1.2.1. Recuperación metalúrgica	71
3.1.2.2. Net Smelting Return	72
3.1.2.3. Parámetros geotécnicos	74
3.2. METODOLOGÍA	75
3.3. TIPO DE INVESTIGACIÓN	76
3.4. DISEÑO DE INVESTIGACIÓN.....	76



3.5. VARIABLES	76
3.5.1. Variables Independientes	76
3.5.2. Variables Dependientes	77
3.6. POBLACIÓN Y MUESTRA.....	77
3.6.1. Población	77
3.6.2. Muestra	77

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. IMPLEMENTACIÓN DEL ALGORITMO ORIGINAL DE LG EN PYTHON.....	79
4.2. IMPLEMENTACIÓN DE ANGULO VARIABLE.....	86
4.2.1. Restricciones de acceso y minado	87
4.2.2. Métodos no basados en conos:.....	88
4.2.3. Métodos basados en conos.....	89
4.2.4. Algoritmo De Lerchs Y Grossmann Con Angulo Variables	89
4.2.5. Angulo variable.....	90
4.3. Software	96
4.3.1. Requerimientos del sistema	96
4.3.2. Estructura del software	96
4.3.3. Clases del software:	97
4.3.3.1. Clase Importa_Data:	97
4.3.3.2. Clase AlgoritmoLG:	98
4.3.3.3. Clase Graphical Unit Interface (GUI).....	98
4.3.4. Funciones del algoritmo original de Lerchs y Grossmann	102
4.3.4.1. Función HallarCamino:	102
4.3.4.2. Función RevisarEstadoNodo:	103
4.3.4.3. Función AgregarPrecedencia:.....	103



4.3.4.4. Función Normalizar:	103
4.3.4.5. Función EliminarArco:	104
4.3.5. Requerimientos de Entrada (Input):.....	104
4.3.6. Resultados (Output):	105
4.4. OPTIMIZACIÓN DEL PIT FINAL CON LA LIBRERÍA DE LG.....	105
4.4.1. Aspectos generales de la mina Constanca – Hudbay.....	105
4.4.1.1. Ubicación y acceso	105
4.4.1.2. Descripción del proyecto	106
4.4.2. Mineralización del Tajo Pampacancha	107
4.4.3. Modelo de bloques 3D	108
4.4.4. Valorización económica con Script en Python	108
4.4.5. Optimización del Pit Final	108
V. CONCLUSIONES.....	112
VI. RECOMENDACIONES	114
VII. REFERENCIAS BIBLIOGRÁFICAS.....	115
ANEXOS.....	118

Área : Ingeniería de Minas

Tema : Diseño y Planeamiento en Minería

FECHA DE SUSTENTACIÓN: 15 de setiembre del 2023



ÍNDICE DE TABLAS

Tabla 1 Categorización de arcos en un grafo dirigido	53
Tabla 2 Modelo de bloques de Tajo Pampacancha 10 x 10 x 15m.....	67
Tabla 3 Modelo de bloques rebloqueado de Tajo Pampacancha 20 x 20 x 15m.....	67
Tabla 4 Variables del modelo de bloques de Tajo Pampacancha	68
Tabla 5 Modelo de recuperación metalúrgica.....	71
Tabla 6 Costo de Minado	72
Tabla 7 Costo de Procesos y Gastos Administrativos	73
Tabla 8 Costo de venta de concentrado de cobre y molibdeno.....	73
Tabla 9 Costo de tratamiento	73
Tabla 10 Costo de refinación por metal	74
Tabla 11 Precio base de los metales	74
Tabla 12 Variables Dependientes	77
Tabla 13 Reservas probadas y probables de Tajo Pampacancha.....	107



ÍNDICE DE FIGURAS

Figura 1 Actividades para la Optimización con la Metodología Tradicional.....	27
Figura 2 Flujograma de algoritmo de Cono Flotante.....	29
Figura 3 Cono Flotante: Matriz de bloques	30
Figura 4 Cono Flotante: Primer cono	31
Figura 5 Cono Flotante: Segundo cono	31
Figura 6 Cono Flotante: Tercer cono	32
Figura 7 Cono Flotante: Cuarto cono	32
Figura 8 Cono Flotante: Diseño final del tajo	33
Figura 9 Cono Flotante: Problema 1, matriz de bloques	34
Figura 10 Cono Flotante: Primer cono cuya extracción no se realizaría.....	34
Figura 11 Cono Flotante: Segundo cono cuya extracción no se realizaría.....	35
Figura 12 Cono Flotante: Diseño optimo	35
Figura 13 Cono Flotante: Segundo problema, matriz de bloques	36
Figura 14 Cono Flotante: Cono de gran tamaño para el bloque 3, 3	36
Figura 15 Cono Flotante: Cono menor para el bloque de la fila 2, 2	37
Figura 16 Korobov: Valores del modelo en una sección vertical.....	39
Figura 17 Korobov: Primer cono	39
Figura 18 Korobov: Primer cono	40
Figura 19 Korobov: Segundo cono con valor +2.....	40
Figura 20 Korobov: Segundo cono con valor +2.....	41
Figura 21 Korobov: Tercer cono con un valor de +4	42
Figura 22 Korobov: Tercer cono	42
Figura 23 Korobov: Quinto cono con un valor de +4.....	43
Figura 24 Korobov: Ejemplo el cual no tiene una solución optima	43
Figura 25 Korobov: Ejemplo el cual no tiene una solución optima	44
Figura 26 Korobov: Ejemplo el cual no tiene una solución optima	44
Figura 27 Korobov: Ejemplo el cual no tiene una solución óptima, primer cono.....	45
Figura 28 Korobov: Ejemplo el cual no tiene una solución optima	45
Figura 29 Korobov: Límite de pit con valor 0.....	46
Figura 30 Korobov: Ejemplo de algoritmo Korobov corregido	47
Figura 31 Korobov: Ejemplo de algoritmo Korobov corregido	48
Figura 32 Cierre de un grafo.....	49



Figura 33	Grafo Ficticio y Arcos.....	50
Figura 34	Pasos del algoritmo de Lerchs y Grossmann	55
Figura 35	Seudocódigo de tiempo de computación de algoritmo de LG	56
Figura 36	Secuencias de extracción Best case y Worst case	58
Figura 37	Vista en Planta del Modelo de Bloques en 3D.....	69
Figura 38	Vista lateral Izquierdo del modelo de bloques en 3D	69
Figura 39	Vista lateral Derecho del modelo de bloques en 3D	70
Figura 40	Sectores geotécnicos del Tajo Pampacancha	75
Figura 41	Ejemplo base para implementación de algoritmo de LG	79
Figura 42	Ejemplo base LG: Grafo normalizado	80
Figura 43	Ejemplo base: Selección y creación de arco (x_5, x_1).....	80
Figura 44	Ejemplo base LG: Etiquetado de los arcos.....	81
Figura 45	Ejemplo base LG: Adición de arco (x_5, x_2) y eliminación de (x_0, x_2).....	81
Figura 46	Ejemplo base LG: Etiquetado de los arcos.....	82
Figura 47	Ejemplo base LG: Adición de arco (x_5, x_3) y eliminación de (x_0, x_3).....	83
Figura 48	Ejemplo base LG: Adición de arco (x_6, x_3) y eliminación de (x_0, x_1).....	84
Figura 49	Configuración de método no basado en conos.....	88
Figura 50	Modelo de bloques con sistema de coordenadas.....	90
Figura 51	Cono de extracción con envolventes.....	91
Figura 52	Área superior del cono en el nivel superior.....	92
Figura 53	Sección A-A Norte-Sur	92
Figura 54	Sección A-A Este-Oeste.....	93
Figura 55	Vista en Planta de un bloque en el nivel superior	95
Figura 56	Comando para ejecución de software en CMD.....	98
Figura 57	GUI: Barra de herramientas	99
Figura 58	GUI: Ventana principal	100
Figura 59	GUI: Ventana UPL Optimizer.....	101
Figura 60	GUI: Ventana 3D Vizualisation	102
Figura 61	Ubicación Mina Constancia	106
Figura 62	Vista 3D-Planta: Pit Final del tajo Pampacancha.....	109
Figura 63	Vista 2D-Seccion A-A': Pit Final del tajo Pampacancha	109
Figura 64	Vista Planta: Pit Final-Modelo de bloques.....	110
Figura 65	Vista Sección A-A': Pit Final-Modelo de bloques.....	110
Figura 66	Vista Sección B-B': Pit Final-Modelo de bloques	111



INDICE DE ANEXOS

Anexo 1. Script para Valorización Del Modelo Geológico.....	118
Anexo 2. Plano De Ubicación, Tajo Constancia Y Pampacancha	127



ÍNDICE DE ACRÓNIMOS

Cu	: Ley de Cobre (%)
Ag	: Ley de Plata (ppm)
Au	: Ley de Oro (ppm)
LG	: Algoritmo de Lerchs y Grossman
Ppm	: Unidad de medida en partes por millón
g	: Unidad de medida en gramos
ton	: Unidad de medida en tonelada
m	: Unidad de medida metro
USD o \$: Dólares americanos
DBS	: Direct Block Scheduling (Agendamiento Directo de Bloques)
%	: Porcentaje
NSR	: Retorno Neto después de Fundición (Net Smelting Return)
Python	: Lenguaje de programación de alto nivel
Grafo	: Conjunto de nodos y arcos
Pit Final	: Límite máximo de minado
Cerradura Máxima	: Conjunto de nodos con valor positivo
Software	: Conjunto de programas
Programa	: Consiste básicamente de una función



RESUMEN

El objetivo de esta investigación es la optimización del pit final en minas a cielo abierto haciendo uso del algoritmo de Lerchs and Grossman modificado con ángulos variables implementado en Python; por medio del cual se maximizará el valor neto del negocio minero. La investigación es de tipo aplicada, con un diseño experimental y enfoque cuantitativo que busca la implementación del algoritmo mediante la revisión de literatura, luego la implementación del algoritmo de LG en código Python haciendo uso de técnicas de programación. Finalmente, se incorporará la metodología de Cono Plantilla para la implementación de ángulo variable. Existe una variedad de algoritmos desarrollados, a la fecha, para resolver el problema del pit final; enfoques basados en la teoría de grafos, programación lineal, programación dinámica, inteligencia artificial, y métodos heurísticos. Entre todos ellos el algoritmo de Lerchs y Grossmann es el más conocido y usado. La evaluación de los resultados en términos de valor neto; diferencia entre el valor total de mineral extraído y el costo total de extracción del mineral y desmonte, indica que se obtuvo el diseño de un pit final óptimo con un valor neto de USD 0.54 millones, el cual cumplió con la restricción de los parámetros económicos, técnicos y geotécnicos para el caso de estudio desarrollado. La conclusión indica que la implementación del algoritmo de LG es viable en uno de los lenguajes de programación más usados y de fácil aprendizaje, Python, haciendo uso de matemáticas discretas (teoría de grafos) y técnicas de programación, además de la implementación de una interfaz gráfica para cargar el modelo de bloques con sus respectivos parámetros y visualizar los resultados de la optimización. Esto para medir la viabilidad económica en etapas tempranas del proyecto como estudios a nivel conceptual.

Palabras Clave: Python, pit final, grafo, cerradura máxima, algoritmo recursivo, programación dinámica.



ABSTRACT

The objective of this research is the optimization of the final pit in open pit mines using the modified Lerchs and Grossman algorithm with variable angles implemented in Python; by means of which the net value of the mining business will be maximized. The research is of an applied type, with an experimental design and quantitative approach that seeks the implementation of the algorithm through literature review, then the implementation of the LG algorithm in Python code making use of programming techniques. Finally, the Cone Template methodology will be incorporated for the variable angle implementation. There are a variety of algorithms developed, to date, to solve the final pit problem; approaches based on graph theory, linear programming, dynamic programming, artificial intelligence, and heuristic methods. Among them, the Lerchs y Grossmann algorithm is the best known and most widely used. The evaluation of the results in terms of net value; the difference between the total value of extracted mineral and the total cost of mineral extraction and clearing, indicates that the design of an optimal final pit with a net value of U\$D 0.54 million was obtained, which complied with the restriction of the economic, technical and geotechnical parameters for the developed case study. The conclusion indicates that the implementation of the LG algorithm is feasible in one of the most used and easy to learn programming languages, Python, making use of discrete mathematics (graph theory) and programming techniques, in addition to the implementation of a graphical interface to load the block model with its respective parameters and visualize the optimization results. This to measure the economic feasibility in early stages of the project as conceptual level studies.

Keywords: Python, ultimate pit, graph theory, maximum closure, recursive algorithm, dynamic programming.



CAPÍTULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

El problema en el diseño del pit final radica en la dificultad de establecer de manera precisa el límite máximo de minado en minas a cielo abierto, con el objetivo de maximizar el valor neto del pit final. La importancia de delimitar el pit final y dimensionar las fases de minado en etapas tempranas de cualquier proyecto es relevante; para dimensionar la mina y justificar la inversión del capital, por supuesto, se necesita tener certeza de la existencia de reservas para sustentar la vida de la mina. Así como también para la localización de los componentes de la mina: botaderos, planta de recuperación metalúrgica, rutas de acarreo, etc. El algoritmo de Lerchs y Grossmann se ha convertido en un estándar globalmente reconocido y respaldado por instituciones financieras y organismos técnicos como AUSIMM, SAIMM, SME, entre otros, en el proceso de optimización de minas a cielo abierto. Sin embargo, en el mercado actual, existen limitaciones significativas en cuanto a la disponibilidad de herramientas gratuitas y la flexibilidad de software para resolver el problema del pit final de forma eficiente y sin costo alguno. Finalmente, dado que la literatura sobre la implementación del algoritmo de Lerchs y Grossmann original e implementación de ángulos variables en el algoritmo de LG es escasa, surge la necesidad de proporcionar una explicación detallada de la metodología para llevar a cabo estas implementaciones, a fin de cubrir este vacío en el conocimiento y facilitar su uso en la industria minera.

1.2. FORMULACIÓN DEL PROBLEMA

1.2.1. Problema General

¿Cuál es la mejor metodología para resolver eficientemente el problema de optimización del pit final en minas a cielo abierto en yacimientos de cobre tipo skarn?

1.2.2. Problemas Específicos

- ¿Cuáles son los desafíos clave en la implementación del algoritmo original de LG en Python y cómo pueden superarse para la optimización del pit final en minas a cielo abierto?
- ¿Cuáles son los desafíos y beneficios asociados con la implementación de ángulos variables en el algoritmo de Lerchs y Grossmann en Python con el fin de mejorar la optimización de minas a cielo abierto?
- ¿Cuál es la metodología para optimizar el pit final utilizando la librería de Lerchs y Grossmann en Python, e incorporando conceptos de ángulos variables en minas a cielo abierto?

1.3. HIPÓTESIS DE LA INVESTIGACIÓN

1.3.1. Hipótesis General

La mejor metodología para resolver eficientemente el problema del pit final es el desarrollo de un aplicativo que implemente el algoritmo de Lerchs y Grossmann en el lenguaje de programación Python.

1.3.2. Hipótesis Específicas

- La implementación exitosa del algoritmo original de Lerchs y Grossmann en Python puede presentar desafíos técnicos, pero permitirá abordar de manera efectiva la delimitación precisa del límite máximo de minado en minas a cielo abierto, lo que a su vez contribuirá a maximizar el valor neto del pit final.



- La implementación de ángulos variables en el algoritmo de Lerchs y Grossmann en Python presenta desafíos técnicos, pero ofrece beneficios significativos en términos de una optimización más precisa y eficiente de minas a cielo abierto en comparación con la implementación del algoritmo original de LG en el mismo lenguaje.
- Es posible optimizar el pit final utilizando la librería de Lerchs y Grossmann en Python, con la incorporación de ángulos variables; lo que resultará en la maximización del valor neto del pit en minas a cielo abierto de yacimientos tipo skarn.

1.4. JUSTIFICACIÓN DE LA INVESTIGACIÓN

- La importancia de delimitar el pit final y dimensionar las fases de minado en etapas tempranas de cualquier proyecto es relevante; para dimensionar la mina y justificar la inversión del capital, por supuesto, uno necesita tener certeza de que hay reservas para sustentar la vida de la mina. Así como también para la localización de los componentes de la mina: botaderos, planta de recuperación metalúrgica, rutas de acarreo, etc. Puesto que el uso del algoritmo de Lerchs y Grossmann es un estándar a nivel mundial, aprobado por instituciones financieras y está aprobado por la mayoría de los organismos técnicos como AUSIMM, SAIMM, SME, etc., en el proceso de optimización de minas a cielo abierto, Por lo tanto, es justificado, aun, el uso de este método para la optimización de minas a cielo abierto.



1.5. OBJETIVOS DE LA INVESTIGACIÓN

1.5.1. Objetivo General

Implementar el algoritmo de Lerchs y Grossmann y ángulo variable en el lenguaje de programación Python para el diseño óptimo y maximización del pit final en minas a cielo abierto en yacimientos de cobre skarn.

1.5.2. Objetivos Específicos

- Implementar exitosamente el algoritmo de Lerchs y Grossmann original en el lenguaje de programación Python como una solución para optimizar el pit final.
- Implementar exitosamente ángulo variable al algoritmo de Lerchs y Grossmann original en el lenguaje de programación Python.
- Determinar el diseño del pit final maximizando el valor neto de este mismo en minas a cielo abierto mediante el uso de la librería de Lerchs y Grossmann en Python.



CAPÍTULO II

REVISION DE LITERATURA

2.1. ANTECEDENTES DE INVESTIGACIÓN

El planeamiento de minado estratégico es realizado con el enfoque tradicional; es decir, el uso del algoritmo de Lerchs y Grossmann con sus variaciones y modificaciones implementadas en software comercial: MineSight, Datamine, Whittle, Vulcan, etc. Los cuales son cajas negras; es decir que no se conoce como es el funcionamiento de la implementación del algoritmo de Lerchs y Grossmann (L&G), simplemente se aceptan los resultados como óptimos y verdaderos, pero que en realidad son sub-óptimos y más aún cuando se tiene taludes complejos porque la mayoría del software desarrollados no detallan la implementación o el algoritmo usado. Uno de los aspectos más importantes en el planeamiento estratégico es la delimitación del pit final o envolvente económica, el cual varía en función del tiempo; cada vez que se obtiene más información técnica y económica, (Kim, 1979). Kim subraya que se le ha dado mucha importancia a la delimitación del pit final. El justifica que solo se debe solucionar el problema del pit final en algunas circunstancias: El pit alcanza su última fase, el depósito es relativamente pequeño o muy angosto, el yacimiento es muy irregular con mineralización por zonas. Luego, (Mathieson, 1982) refuta diciendo que si es de suma importancia: “Para dimensionar el complejo de recuperación y justificar la inversión del capital”, por supuesto, uno necesita tener certeza de que hay reservas para sustentar la vida de la mina, por lo tanto, la definición del pit final basado en las mejores estimaciones económicas y de taludes es de suma importancia así como también para la localización de los componentes de la mina: botaderos, planta de recuperación metalúrgica, rutas de acarreo, etc.



Este problema ha sido abordado por más de 50 años desde la gloriosa publicación del algoritmo por Lerchs y Grossmann y la primera implementación de L&G por Jeff Whittle quien no da detalles de la implementación. Este problema puede ser abordado como: un problema de investigación de operaciones (programación lineal y flujo de redes) y matemáticas discretas (teoría de grafos). Picard demuestra que encontrar una solución al problema del máximo closure (cerradura) de un grafo es el equivalente de resolver el problema de flujo máximo en una red formada por un grafo (Picard 1976). Todos estos son algoritmos los cuales se clasifican en heurísticos y rigurosos; La palabra "heurística" se usa para describir un algoritmo que funciona en casi todos los casos pero que carece de una prueba matemática rigurosa, (Kim 1979), la experiencia demuestra que funcionan satisfactoriamente, aunque no poseen demostración matemática que permitan asegurar su validez, por ejemplo, como flotante. Además, estos conducen a error como el caso de superposición. El segundo tiene una completa demostración matemática, por ejemplo, Lerchs y Grossmann. El algoritmo de Lerchs y Grossmann ha sido utilizado por más de 50 años por la industria minera, pero este cuenta con un problema; el valor del dinero en el tiempo no es tomado en cuenta en este algoritmo, cosa que no debería ser así, puesto que se asume que el yacimiento será extraído en un solo periodo. Es más, este problema en todos los casos es resuelto castigando los bloques con una tasa de descuento, pero al final esto conduce a resultados sub-óptimos.

Los pushbacks o fases de minado son la forma en la que se va extraer las reservas de un yacimiento de la mejor forma económicamente posible la cual está en función del ritmo de producción de planta, mina, pad de lixiviación, etc. Estos, también, están en función de la política de la empresa; es decir se desea obtener la más rápida recuperación del capital invertido o prolongar la vida de la mina con ganancias constantes; Política de extracción e/m (estéril/mineral) descendente, e/m creciente, e/m constante. El principal



objetivo de la generación de las fases es obtener la secuencia de extracción óptima con todas las limitaciones: capacidad planta, capacidad de minado, blending, ancho mínimo de minado entre otros. La secuencia que genera el máximo NPV es el que debe ser seleccionado, (Gershon 1987). Una vez que se logró obtener los datos relevantes del proyecto minero: límites de minado, secuencia de extracción, cuadro de reservas, espacios y lugares de acopio; aun es necesario definir el plan de producción flexible y dinámico en el tiempo buscando que se pueda aplicar en tiempo real. Cousenz (1979) propone 5 reglas generales para el planeamiento a largo plazo: Se debe mantener los objetivos claros porque se trabaja con estimaciones de leyes, proyecciones en geología y pronósticos de parámetros económicos. Se debe estar dispuesto al cambio, debemos estar comunicados. Si el planeamiento no está claro para aquellos quienes toman las decisiones ni para los que las ejecutan, entonces el planeamiento será malentendido o ignorado, debemos recordar que estamos trabajando con volumen de material que deben ser movidos en secuencia. Geometría es tan importante como aritmética para un planeador, debemos recordar que se trabaja con tiempo. Los volúmenes deben ser movidos para cumplir los objetivos de producción. El uso productivo del tiempo determinara eficiencia y efectividad en los costos, debemos buscar la aceptación de nuestros planes tal que se conviertan los objetivos de la empresa y no simplemente las ideas de los planeadores. Además, entre otros autores se tienen a Henry Chura Torres (2017), en su tesis: “Optimizar el planeamiento a largo plazo de la mina a tajo abierto Tacaza – CIEMSA empleando distintos software mineros GEMCOM y MineSight”, Universidad Nacional del Altiplano, el cual tiene por objetivo evaluar la valorización de la mina Tacaza habiendo obtenido el planeamiento a largo plazo acorde a la realidad de la mina. Concluye, que la valorización de la mina Tacaza es óptima, no solo por conciliar las restricciones operativas y las condiciones de mercado con la interpretación geológica,



sino que también por la adición de conceptos y aseveraciones de otros autores que mejoran el desarrollo del planeamiento a largo plazo en post de la obtención de una mejor valorización, también concluye con una rentabilidad económica de la mina Tacaza mejorada, con un VAN operativo mejorado, y un TIR de 18% (Chura 2019).

Fernando Peirano Opazo (2011), en su tesis: “Definición del pit final capacitado bajo incertidumbre”, Universidad de Chile, el cual tiene por objetivo principal desarrollar una metodología y una herramienta que permita definir el pit final incorporando las restricciones de capacidad y las condiciones de incertidumbre asociadas al recurso geológico. Las conclusiones más importantes de este estudio son: las formas de los pit finales generados en cada caso muestran una fuerte dependencia con la restricción capacitaría impuesta, es decir una capacidad mayor genera mayor valor en el flujo de caja; ya que los bloques se minan en el menor tiempo posible, y una capacidad menor implica lo contrario. La segunda conclusión relevante es la capacidad de incorporar la incertidumbre dentro de las condiciones del problema, lo cual es enriquecedor para la línea de pensamiento que busca tomar en cuenta la incertidumbre intrínseca en la industria minera (Opazo 2011).

Tomas Gonzales Paihua (2010), en su tesis: “Diseño de minas a tajo abierto”, investigada en la Universidad Nacional de Ingeniería, examinó en detalle las diferentes etapas requeridas para llevar a cabo el diseño de una mina y calcular las reservas minerales en un yacimiento específico. El estudio se centró en identificar y analizar las conclusiones más significativas. Una de las conclusiones clave de su investigación es que, para desarrollar y evaluar activos mineros de manera efectiva, es esencial contar con una base de conceptos clara y una terminología estandarizada que respalde los prospectos de exploración, recursos y reservas minerales. Además, se destacó la importancia de consideraciones geométricas relacionadas con el tamaño del yacimiento, así como



consideraciones geomecánicas, el ratio de minado, el tamaño de los equipos y otros factores relacionados con la geometría de los depósitos minerales. Una correcta estrategia de ley de corte permite mejorar el VAN del proyecto y reducir el tiempo de retorno de la inversión (Paihua 2010).

Ronald Benito López (2013), en su tesis: “Determinación de límites óptimos de minado con aplicación de bases de datos”, Universidad Nacional del Altiplano, el cual tiene por objetivo la determinación de los límites máximos de minado con la aplicación del algoritmo de Lerchs y Grossmann con base de datos. Concluye que almacenar y administrar datos es para producir información útil de manera simple y efectiva, la cual es usada para hacer decisiones claves de negocio (Benito 2013).

Eduardo Latorre Nanjarí (2015), en su tesis: “Incorporación del valor del dinero en el tiempo para la determinación de límites económicos de un open pit”, Pontificia Universidad Católica de Chile, el cual tiene por objetivo la incorporación del valor del dinero en el tiempo como variable relevante para resolver el problema de límites de pit por medio de la implementación de un nuevo algoritmo que combina programación dinámica y heurística. Concluye que efectivamente, bajo ciertas condiciones, se genera una diferencia de límites de minado al cambiar la variable a maximizar; beneficio neto y valor presente neto (Nanjari 2015).

Por último, Dowd y Fowell, en su trabajo de investigación: “Algoritmo de Lerchs y Grossmann con ángulo variable”, el cual tiene por objetivo implementar ángulo variable al algoritmo original de Lerchs y Grossmann a través de la metodología de cono invertido. Concluye, que el método implementado supera la limitación del algoritmo original de LG. Además, el método es capaz de generar límites de minado óptimos tanto para modelo de bloques cúbicos y rectangulares (Khalokakaie, R., Dowd, P. and Fowell 2000).



2.2. MARCO TEÓRICO

2.2.1. Planificación minera

La planificación minera es el procedimiento que posibilita la identificación y posterior extracción de recursos en el momento oportuno y con el menor gasto posible, con el propósito de alcanzar los objetivos establecidos por la empresa o el propietario de la mina. Cualquier proceso de planificación demanda que se haya definido con anterioridad una declaración de la misión empresarial, la cual puede incluir cualquiera de los siguientes objetivos: maximizar la rentabilidad, minimizar los costos de producción, optimizar la producción de cobre, garantizar la seguridad de los trabajadores, cumplir con las regulaciones ambientales, minimizar el impacto en las comunidades locales, entre otros. La misión empresarial debe ser clara y estar alineada con los valores y objetivos a largo plazo de la compañía minera.

En función del horizonte temporal de evaluación, la planificación se divide en largo, mediano y corto plazo (Osanloo, Gholamnejad y Karimi, 2008).

2.2.1.1. La planificación minera a largo plazo

En la planificación minera a largo plazo, se busca definir la secuencia de extracción de los recursos a lo largo de la vida útil del yacimiento. Esto implica determinar la ubicación de las operaciones mineras, los diseños de las fases de extracción y la secuencia de producción. Además, se considera la planificación de la infraestructura necesaria, como caminos, instalaciones de procesamiento y áreas de almacenamiento.

2.2.1.2. La planificación minera a mediano plazo

En la planificación minera a mediano plazo, se detallan los planes operativos que se implementarán en un horizonte temporal más cercano,



generalmente de uno a cinco años. Esto incluye la programación detallada de la extracción y el procesamiento de los recursos, la asignación de equipos y personal, y la planificación de la producción.

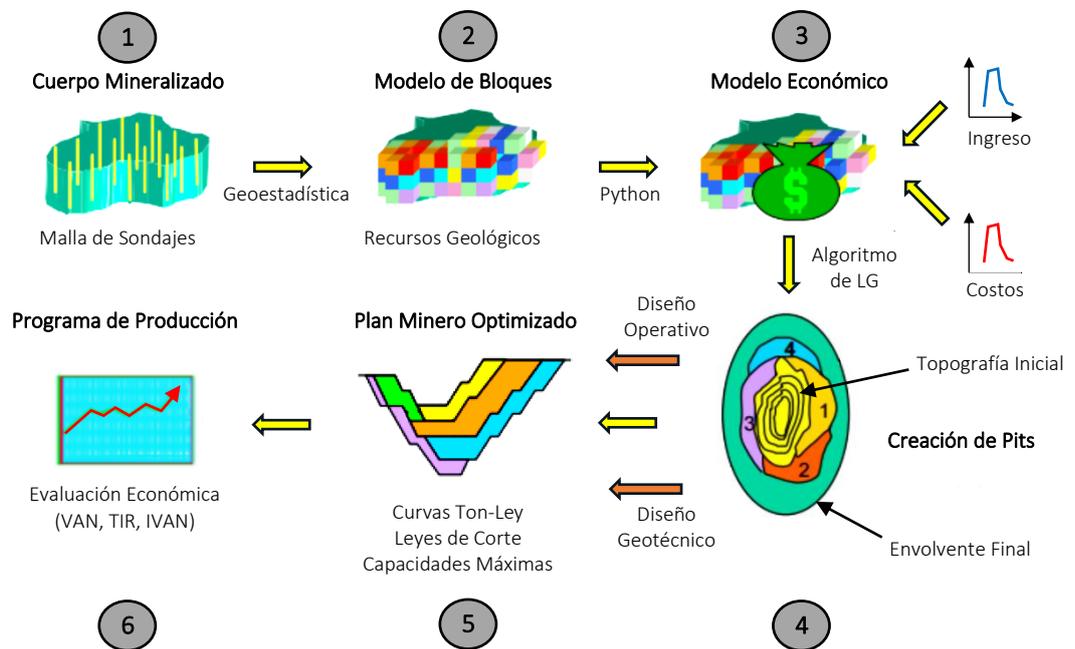
2.2.1.3. La planificación minera a corto plazo

En la planificación minera a corto plazo, se establecen los planes diarios o semanales para la extracción de los recursos. Esto implica la asignación de equipos y personal en función de las condiciones operativas y la optimización de la producción en el corto plazo.

En cada etapa de la planificación, se deben considerar diversos factores, como la geología del yacimiento, la calidad y cantidad de los recursos, los costos de extracción y procesamiento, la disponibilidad de equipos y personal, y los requerimientos ambientales y de seguridad. Estos deben ser capaces de hacer sinergias entre sí de tal forma de lograr optimizar el proceso global, como se muestra en la Figura 1. Además, es importante tener en cuenta la incertidumbre geológica y los escenarios económicos, ya que pueden tener un impacto significativo en la rentabilidad del proyecto.

Figura 1

Actividades para la Optimización con la Metodología Tradicional



Fuente: Figuras extraídas de (Morales et al, 2015)

2.2.2. Métodos de optimización de límites económicos en minas a cielo abierto

La evaluación de un depósito minero a menudo representa un desafío complejo que la mayoría de los ingenieros de minas deben abordar al planificar eficazmente la secuencia de extracción y los límites de explotación en un proyecto minero. Esto se hace con el objetivo de maximizar los ingresos generados. La razón detrás de esta complejidad radica en la necesidad de comprender una amplia gama de variables posibles para poder hacer una estimación precisa de los beneficios derivados de la extracción con un alto grado de confiabilidad.

Existen distintos métodos para definir los límites económicos de una mina a cielo abierto, que sin duda cada uno aporta un concepto útil y que en muchos casos se combinan para generar otro método. Estos se dividen en heurísticos y rigurosos; en los métodos heurísticos la experiencia demuestra que funcionan



satisfactoriamente, aunque no poseen de demostraciones matemáticas que permitan asegurar su validez, por ejemplo, el método del cono flotante además que estos conducen a error. Por el otro lado los métodos rigurosos los cuales tiene una completa demostración matemática como el algoritmo de Lerchs y Grossmann, y Pseudoflow.

Todos estos elementos contribuyen a la determinación del límite óptimo en minas a cielo abierto, estableciendo ese límite en el punto en el que se logra la maximización de los ingresos y definiendo la secuencia en la que se extraerá el material que se encuentra dentro de los límites máximos de minado.

Estas técnicas se emplean extensamente en yacimientos masivos y diseminados, y se aplican de manera limitada en yacimientos estratificados, especialmente cuando estos tienen una orientación horizontal o subhorizontal.

Dado que la delimitación de los límites máximos de minado en una operación minera a cielo abierto es un factor crucial en su diseño, estos enfoques se utilizan como fundamentos en software de simulación y optimización minera ampliamente adoptados en todo el mundo.

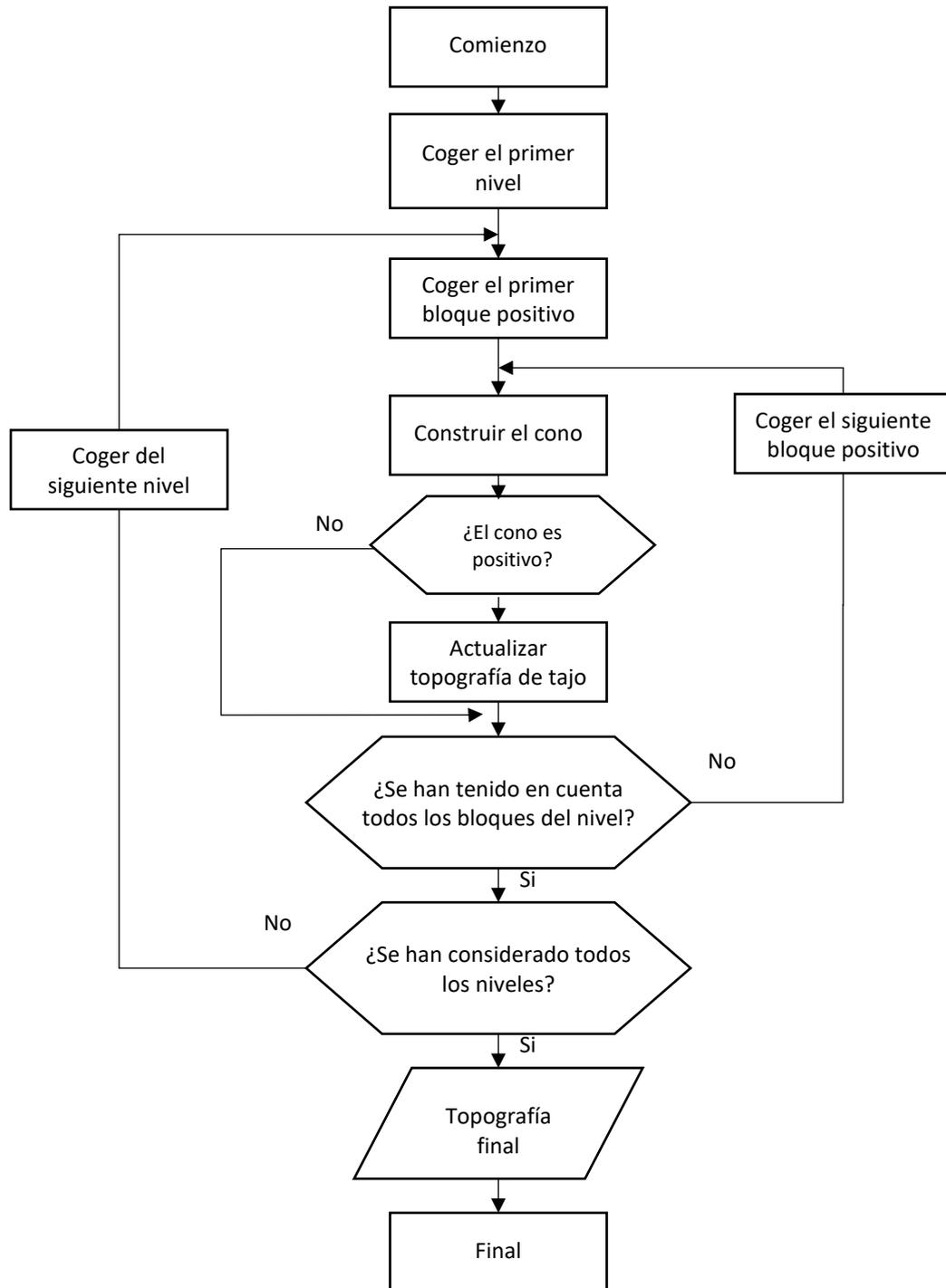
2.2.2.1. Algoritmo de cono flotante

La teoría del cono flotante, que se utiliza para establecer los límites económicos de una mina a cielo abierto, se originó en la década de 1960. Este enfoque implica una rutina que evalúa si es conveniente extraer un bloque y su sobrecarga correspondiente. En este proceso, el algoritmo convencional se ubica en cada bloque con valor económico positivo en el modelo de bloques y crea un cono invertido, donde la superficie lateral del cono representa el ángulo de inclinación. Si el beneficio neto obtenido del cono es igual o mayor que un

beneficio deseado, se procede a extraer el cono; de lo contrario, se mantiene en su posición original, como se puede ver en mayor detalle en Figura 2.

Figura 2

Flujograma de algoritmo de Cono Flotante



En la Figura 3 siguiente, se muestra un perfil de un modelo de bloques que ha sido sometido al algoritmo del cono flotante. En este modelo, cada bloque tiene un valor económico, es decir, el beneficio que se obtendría al extraerlo. De esta manera, los bloques con valor negativo representan el material estéril y su correspondiente costo de extracción (-1, -2 y -3), mientras que los bloques de mineral se identifican por el beneficio total que se obtendría de su extracción (Beneficio Total = Ingresos - Costos = 800).

Figura 3

Cono Flotante: Matriz de bloques

	1	2	3	4	5	6	7
1	-1	-1	-1	-1	-1	+1	-1
2		-2	-2	+4	-2	-2	
3			+7	+1	-3		

1.- En la fila inicial, encontramos un bloque con un valor económico positivo. Dado que no hay bloques por encima de él, su extracción conduciría a resultados positivos, y el valor del cono sería igual al valor del bloque (+1), Figura 4.

Figura 4

Cono Flotante: Primer cono

	1	2	3	4	5	6	7
1	-1	-1	-1	-1	-1	+1	-1
2		-2	-2	+4	-2	-2	
3			+7	+1	-3		

2.- El bloque siguiente se determina a partir del bloque ubicado en la fila 2 y columna 4, que tiene un valor de (+4). El valor del cono se calcula como $-1 - 1 - 1 + 4 = +1$. Dado que el valor del cono es positivo, se procedería a extraerlo, Figura 5.

Figura 5

Cono Flotante: Segundo cono

	1	2	3	4	5	6	7
1	-1	-1	-1	-1	-1		-1
2		-2	-2	+4	-2	-2	
3			+7	+1	-3		

3.- Luego, el bloque siguiente que se evaluaría es aquel ubicado en la fila 3 y columna 3, que tiene un valor de (+7). El valor del cono generado por este bloque se calcula como $-1 - 1 - 2 - 2 + 7 = +1$. Una vez más, dado que el valor del cono es positivo, este bloque también se extraería, Figura 6.

Figura 6

Cono Flotante: Tercer cono

	1	2	3	4	5	6	7
1	-1	-1					-1
2		-2	-2		-2	-2	
3			+7	+1	-3		

4.- Finalmente, el último cono se determina a partir del bloque ubicado en la fila 3 y columna 4, que tiene un valor de (+1). La extracción de este cono daría como resultado el siguiente valor: $-2 + 1 = -1$. En esta ocasión, el valor es negativo, lo que significa que este cono no se extraería., Figura 7.

Figura 7

Cono Flotante: Cuarto cono

	1	2	3	4	5	6	7
1							-1
2					-2	-2	
3				+1	-3		

El diseño del pit final sería el que se muestra en la Figura 8. El valor total del pit vendría dado por: $-1 -1 -1 -1 -1 +1 -1 -1 +4 +7 = +3$

Figura 8

Cono Flotante: Diseño final del tajo

	1	2	3	4	5	6	7
1	-1	-1	-1	-1	-1	+1	-1
2		-2	-2	+4	-2	-2	
3			+7	+1	-3		

Sin embargo, este método de optimización no garantiza siempre la configuración ideal, ya que pueden surgir diversas circunstancias problemáticas.

En particular, se pueden identificar dos posibles escenarios:

1.- La primera dificultad surge cuando se examinan bloques con valores positivos de manera individual. En este caso, un solo bloque puede no justificar la extracción del material circundante, pero cuando se combinan con otros bloques que se superponen, pueden generar valores positivos. Johnson (1973) ha denominado a esta situación como el problema del soporte mutuo. En la Figura 9, podemos observar esta circunstancia. El cono que se forma a partir del bloque ubicado en la fila 3 y columna 3(+10) tiene un valor de: $-1 -1 -1 -1 -1 -2 -2 -2 +10 = -1$

Figura 9

Cono Flotante: Problema 1, matriz de bloques

	1	2	3	4	5	6	7
1	-1	-1	-1	-1	-1	-1	-1
2		-2	-2	-2	-2	-2	
3			+10	-3	+10		

Dado que el resultado del cono es negativo, no se extraería el tajo de la

Figura 10.

Figura 10

Cono Flotante: Primer cono cuya extracción no se realizaría

	1	2	3	4	5	6	7
1	-1	-1	-1	-1	-1	-1	-1
2		-2	-2	-2	-2	-2	
3			+10	-3	+10		

De manera similar, el valor del cono generado a partir del bloque ubicado en la fila 3 y columna 5 (+10) sería: $-1 -1 -1 -1 -1 -2 -2 -2 +10 = -1$, lo que significa que en este caso tampoco se procedería a extraerlo, Figura 11.

Figura 11

Cono Flotante: Segundo cono cuya extracción no se realizaría

	1	2	3	4	5	6	7
1	-1	-1	-1	-1	-1	-1	-1
2		-2	-2	-2	-2	-2	
3			+10	-3	+10		

Por lo tanto, mediante el análisis simple del cono flotante, no se seleccionaría la extracción de ningún bloque. No obstante, debido a la superposición o interacción mutua que existe entre ambos conos, la combinación de estos dos conos generaría un valor positivo: $-1 -1 -1 -1 -1 -1 -1 -2 -2 -2 -2 -2 +10 +10 = +3$. Esto demuestra que este diseño representa la verdadera optimización, como se muestra en la, Figura 12. Situaciones como esta pueden surgir con frecuencia en yacimientos reales, y el método simple de cono flotante no las tiene en cuenta. Por lo tanto, considerar la técnica iterativa mencionada anteriormente es la única forma de abordar eficazmente este tipo de situaciones.

Figura 12

Cono Flotante: Diseño optimo

	1	2	3	4	5	6	7
1	-1	-1	-1	-1	-1	-1	-1
2		-2	-2	-2	-2	-2	
3			+10	-3	+10		

2.- La segunda problemática se presenta cuando el método incorpora bloques que no aportan beneficios al diseño final, lo cual puede disminuir el valor neto de la operación. Considerando la matriz que se muestra en la Figura 13, el cono relacionado con el bloque ubicado en la fila 3 y columna 3 generaría un valor de $-1 -1 -1 -1 -1 +5 -2 -2 +5 = +1$.

Figura 13

Cono Flotante: Segundo problema, matriz de bloques

	1	2	3	4	5
1	-1	-1	-1	-1	-1
2		-5	-2	-2	
3			+5		

Figura 14

Cono Flotante: Cono de gran tamaño para el bloque 3, 3

	1	2	3	4	5
1	-1	-1	-1	-1	-1
2		+5	-2	-2	
3			+5		

La circunstancia de que el valor del cono sea positivo no necesariamente implica que deba ser extraído. Como se puede apreciar en la Figura 14, el bloque ubicado en la fila 2 y columna 2 tendría un valor de (+2), que representa el valor

óptimo del diseño. Esto se debe a que, una vez extraído este bloque, el siguiente generaría resultados negativos (fila 3 y columna 3): $-1 -1 -1 -2 -2 +5 = -1$. En este caso, el valor del cono más pequeño supera al del cono más grande.

Figura 15

Cono Flotante: Cono menor para el bloque de la fila 2, 2

	1	2	3	4	5
1	-1	-1	-1	-1	-1
2		+5	-2	-2	
3			+5		

Lemieux (1979) propuso una serie de técnicas para superar estos problemas, pero las diversas formas del método del cono flotante siguen siendo heurísticas debido a la falta de demostración matemática. A pesar de los desafíos mencionados anteriormente, hay varios aspectos favorables que contribuyen a que esta técnica sea ampliamente utilizada:

1.- Puesto que el método es una informatización de las técnicas manuales, los usuarios pueden utilizarla, entender lo que están haciendo y sentirse satisfechos con los resultados.

2.- Desde una perspectiva de diseño, el algoritmo es bastante sencillo, lo que permite su implementación en un software de computadora de manera rápida y sencilla.

3.- Ofrece resultados lo bastante fiables como para inspirar la confianza necesaria al optimizar una operación minera a cielo abierto.

2.2.2.2. Algoritmo de Korobov

Se trata de un algoritmo basado en conos desarrollado originalmente por Korobov (1974) y descrito por David, Dowd y Korobov (1974), y Dowd y Onur (1992 y 1993). El primer paso del algoritmo es construir, para cada bloque de mineral en el modelo de bloques de valor neto o descontado, un cono de extracción con lados paralelos a los ángulos de inclinación del tajo. El segundo paso consiste en asignar valores positivos dentro de cada cono de extracción a valores negativos dentro del cono hasta que no quede ningún bloque negativo o hasta que se haya asignado el valor total de los bloques positivos. Si el valor del bloque sobre el que se construye un cono de extracción sigue siendo positivo después de que se haya completado la asignación, entonces este cono de extracción se incluye como miembro del conjunto de soluciones óptimas. Cuando se añade un cono de extracción no vacío al conjunto de soluciones, el algoritmo vuelve a empezar desde el principio con los valores de bloque originales restaurados para los bloques aún no extraídos del modelo de bloques. Si un cono de extracción está vacío, el bloque positivo se añade a la solución y el algoritmo continúa con el siguiente bloque mineral. Este proceso se repite hasta que no queda ningún bloque positivo en el modelo de bloques.

Para demostrar cómo funciona el algoritmo de Korobov, se aplica al ejemplo simple bidimensional que se muestra en la Figura 16, en el que la pendiente del tajo es de 45° y los bloques son cuadrados. Para determinar los límites óptimos de pit a partir de este algoritmo se llevan a cabo los siguientes procedimientos:

Figura 16

Korobov: Valores del modelo en una sección vertical

	1	2	3	4	5	6	7
1	-1	-1	-1	-1	-1	+1	-1
2	-2	+2	-2	-2	+3	-2	-2
3	-3	-3	+7	-3	+1	-3	-3

1.- El algoritmo parte del nivel superior. Sólo hay un bloque positivo en este nivel, que se elimina y se incluye en el conjunto de soluciones Figura 17.

Figura 17

Korobov: Primer cono

	1	2	3	4	5	6	7
1	-1	-1	-1	-1	-1		-1
2	-2	+2	-2	-2	+3	-2	-2
3	-3	-3	+7	-3	+1	-3	-3

2.- En el segundo nivel hay dos bloques positivos, los bloques (2,2) y (2,5). La extracción del bloque (2, 2) incluye los bloques (1, 1), y (1, 2) y (1, 3). Un valor de +1 del bloque (2, 2) se asigna al bloque (1, 1), dejando el bloque (1, 1) con un valor de cero y el bloque (2, 2) con un valor de +1. El valor +1 restante del bloque (2, 2) se asigna entonces al bloque (1, 2), dejando ambos bloques con valor cero. De la misma manera, el cono de extracción del bloque (2, 5) comprende los

bloques (1, 4), (1, 5) y (1, 6) de los cuales el bloque (1, 6) ya ha sido eliminado.

El valor del bloque (2, 5) se imputa a los bloques (1, 4) y (1, 5), quedando ambos bloques con valor cero y el bloque (2, 5) con un valor residual de +1 Figura 18.

Figura 18

Korobov: Primer cono

	1	2	3	4	5	6	7
1	0	0	-1	0	0		-1
2	-2	0	-2	-2	+1	-2	-2
3	-3	-3	+7	-3	+1	-3	-3

Como este bloque sigue siendo positivo y su cono de extracción no está vacío, se añade al conjunto de soluciones óptimas, que ahora tiene un valor de +2, y el algoritmo empieza desde el principio con los valores originales de los bloques restaurados en todos los bloques no eliminados Figura 19.

Figura 19

Korobov: Segundo cono con valor +2

	1	2	3	4	5	6	7
1	-1	-1	-1				-1
2	-2	+2	-2	-2		-2	-2
3	-3	-3	+7	-3	+1	-3	-3

3.- Dos unidades del valor del bloque (2, 2) se asignan contra los bloques (1, 1) y (1, 2), quedando los tres bloques con valor cero. Consideremos ahora el bloque (3, 3). Su extracción contiene sólo los bloques (1, 3), (2, 3) y (2, 4) con valores negativos. Se asigna una unidad al bloque (1, 3) y cuatro unidades a los bloques (2, 3) y (2, 4). Después de la asignación Figura 20.

Figura 20

Korobov: Segundo cono con valor +2

	1	2	3	4	5	6	7
1	0	0	0				-1
2	-2	0	0	0		-2	-2
3	-3	-3	+2	-3	+1	-3	-3

El bloque (3, 3) sigue siendo positivo con un valor de +2. Este bloque y todos los bloques dentro de su extracción tienen un valor positivo. Este bloque y todos los bloques dentro de su cono de extracción se añaden al conjunto de soluciones. El valor neto de este cono de extracción es +2, y el valor neto del pit pasa a ser +4. Como el cono de extracción del bloque (3, 3) no está vacío, el algoritmo empieza desde el principio con los valores originales de los bloques restaurados a todos los bloques no extraídos Figura 21.

Figura 21

Korobov: Tercer cono con un valor de +4

	1	2	3	4	5	6	7
1							-1
2	-2					-2	-2
3	-3	-3		-3	+1	-3	-3

4.- Finalmente, sólo queda un bloque positivo, el bloque (3, 5). Su cono de extracción contiene los bloques (1, 7) y (2, 6) ambos con valores negativos. El valor del bloque (3, 5) se asigna contra el bloque (1, 7), quedando ambos bloques con valor cero. Como el cono de extracción de este bloque contiene un bloque con valor negativo, el bloque (2, 6), y se ha asignado el valor total del bloque positivo, el cono no puede añadirse al conjunto de soluciones Figura 22.

Figura 22

Korobov: Tercer cono

	1	2	3	4	5	6	7
1							0
2	-2					-2	-2
3	-3	-3		-3	0	-3	-3

Este algoritmo da la forma final del pit mostrada en la Figura 23 con un valor de +4.

Figura 23

Korobov: Quinto cono con un valor de +4

	1	2	3	4	5	6	7
1	-1	-1	-1	-1	-1	+1	0
2	-2	-2	-2	-2	+3	-2	-2
3	-3	-3	+7	-3	0	-3	-3

Este método es sencillo y el tiempo de cálculo necesario es inferior al del método de Lerchs-Grossmann. Aunque el algoritmo de Korobov supera algunos puntos débiles del método del cono flotante, en algunas circunstancias el método no proporciona una solución óptima real. Por ejemplo, consideremos el sencillo ejemplo bidimensional de la Figura 24.

Figura 24

Korobov: Ejemplo el cual no tiene una solución óptima

	1	2	3	4	5	6
1	-1	-1	-1	-1	-1	-1
2	-1	-1	-1	-1	-1	-1
3	-1	-1	+3	+7	-1	-1

Tomado de Dowd y Onur (1993). En el modelo de bloques de ingresos hay dos bloques positivos. El valor del primer bloque positivo, el bloque (3,3), se asigna a los bloques (1,1), (1,2) y (1,3), dejando los cuatro bloques con valor cero, Figura 25.

Figura 25

Korobov: Ejemplo el cual no tiene una solución óptima

	1	2	3	4	5	6
1	0	0	0	-1	-1	-1
2	-1	-1	-1	-1	-1	-1
3	-1	-1	0	+7	-1	-1

Consideremos ahora el segundo bloque positivo, el bloque (3,4). El cono de extracción de este bloque contiene seis bloques con valores negativos, cada uno de los cuales se convierte en cero tras la asignación del valor del bloque (3,4). Cuando se ha completado la asignación, este bloque sigue siendo positivo con un valor de +1, Figura 26.

Figura 26

Korobov: Ejemplo el cual no tiene una solución óptima

	1	2	3	4	5	6
1	0	0	0	0	0	0
2	-1	-1	0	0	0	-1
3	-1	-1	0	+1	-1	-1

Y, por lo tanto, este bloque y su cono de extracción se añaden al conjunto de soluciones. Como el valor neto de este cono es -1 , el valor neto del pozo también es -1 en esta fase. Como el cono de extracción del bloque $(3,4)$ no está vacío, el algoritmo empieza desde el principio con los valores originales del bloque restaurados para todos los bloques no eliminados Figura 27.

Figura 27

Korobov: Ejemplo el cual no tiene una solución óptima, primer cono

	1	2	3	4	5	6
1	-1					
2	-1	-1				-1
3	-1	-1	+3		-1	-1

Ahora hay dos bloques con valores negativos (-1) dentro del cono de extracción del bloque $(3,3)$, cada uno de los cuales se convierte en cero cuando se asigna el valor del bloque $(3,3)$. Tras la asignación, este bloque sigue siendo positivo con un valor de $+1$ Figura 28.

Figura 28

Korobov: Ejemplo el cual no tiene una solución óptima

	1	2	3	4	5	6
1	-0					
2	-1	0				-1
3	-1	-1	+1		-1	-1

Por lo tanto, este bloque y su extracción se añaden al conjunto de soluciones que da como resultado un pozo con un valor neto de cero Figura 29.

Figura 29

Korobov: Límite de pit con valor 0

	1	2	3	4	5	6
1	-1	-1	-1	-1	-1	-1
2	-1	-1	-1	-1	-1	-1
3	-1	-1	+3	+7	-1	-1

Por tanto, este algoritmo produce un pit con un beneficio neto de cero.

En el ejemplo anterior, los bloques que son comunes a ambos conos de extracción provocan el error. Dowd y Onur (1992 y 1993) identificaron el fallo en el algoritmo de Korobov y modificaron el algoritmo para producir un verdadero óptimo. Se refieren al proceso de asignación de valores de bloques positivos a valores de bloques negativos como el segundo "pagando" por el primero; un bloque negativo se "paga" cuando el proceso de asignación aumenta su valor a cero. Su corrección del algoritmo de Korobov se basa en la siguiente lógica:

"Si dos o más conos tienen bloques en común, los bloques no comunes deben pagarse primero; los bloques comunes sólo se pagan después de que se hayan pagado todos los bloques no comunes".

Por ejemplo, consideremos de nuevo el ejemplo bidimensional de la Figura 24. Los bloques (1,2), (1,3), (1,4), (1,5), (2,3) y (2,4) son comunes a los conos de extracción de los bloques (3,3) y (3,4). Los bloques (1,1) y (2,2), pertenecen al cono de extracción del bloque (3,3) pero no son miembros del cono

de extracción del bloque (3,4). Lo mismo ocurre con los bloques (1,6) y (2,5), que son miembros del cono de extracción del bloque (3,4).

Los bloques no comunes deben pagarse primero. Empezando por el cono de extracción del bloque (3,3), se asigna una unidad a cada uno de los bloques (1,1) y (2,2), dejando ambos bloques con un valor de cero y el bloque (3,3) con un valor residual de +1. Del mismo modo, se asignan un total de dos unidades, del bloque (3,4), contra los bloques (1,6) y (2,5), dejando ambos bloques con un valor de cero y el bloque (3,4) con un valor de +5 Figura 30.

Figura 30

Korobov: Ejemplo de algoritmo Korobov corregido

	1	2	3	4	5	6
1	0	-1	-1	-1	-1	0
2	-1	0	-1	-1	0	-1
3	-1	-1	+1	+5	-1	-1

Cuando se completa la asignación para los bloques no comunes, el algoritmo comienza de nuevo pagando por los bloques que son comunes a los conos de extracción de ambos bloques. Una unidad del bloque (3,3) se asigna al bloque (1,2), dejando ambos bloques con valor cero. Cinco unidades del bloque (3,4) se asignan a, (1,3), (1,4), (1,5), (2,3) y (2,4), dejando todos los bloques con valor cero Figura 31.

Figura 31

Korobov: Ejemplo de algoritmo Korobov corregido

	1	2	3	4	5	6
1	0	0	0	0	0	0
2	-1	0	0	0	0	-1
3	-1	-1	0	0	-1	-1

Tras la asignación, ninguno de los bloques positivos, sobre los que se construyen los conos de extracción, sigue siendo positivo, por lo que no se añade ningún bloque al conjunto de soluciones óptimas. Como no hay más bloques positivos, el algoritmo se detiene y la solución obtenida es no extraer ningún bloque. En la versión original del algoritmo, ambos conos de extracción formaban parte del conjunto de soluciones; en la versión corregida, ninguno de los conos de extracción pertenece al conjunto de soluciones.

Es evidente que la forma corregida del algoritmo Korobov produce buenos resultados y que el método es más rápido y sencillo que el algoritmo Lerchs-Grossmann. Sin embargo, debido a la falta de una demostración matemática rigurosa, el método no ha sido aceptado como método general para la determinación del pit óptimo.

2.2.2.3. Algoritmo de Lerchs y Grossmann

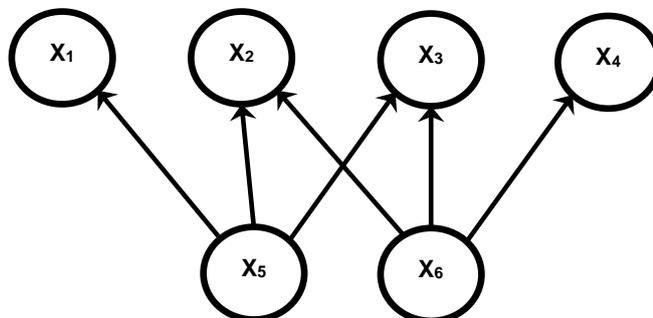
El primer algoritmo, publicado en el año 1965 por Lerchs y Grossmann ha dado inicio a una de las áreas más activas en investigación de operaciones en la industria minera. Este algoritmo se centra en definir el tajo final para maximizar la diferencia entre el valor total del mineral extraído y el costo total de extracción

del mineral y desmonte. Las únicas restricciones se refieren a la geometría del tajo, el talud de las paredes del tajo no debe superar determinados ángulos que pueden variar con la profundidad del tajo o con el material. Es necesario destacar, el algoritmo se basa en la construcción de tajos anidados con el uso de un factor de ingresos que modula el tamaño de los tajos penalizando el precio en la valoración del bloque Morales et al (2015).

La metodología propuesta por Lerchs & Grossmann define el tajo final usando la teoría de grafos donde un grafo dirigido $G = (V, A)$ tal que, un nodo en el grafo representa un bloque en el modelo de bloques y un arco se dirige desde el nodo $x_i \in V$ a un nodo $x_j \in V$, además el nodo x_i debe ser eliminado antes del bloque que representa el nodo x_j (debido a la precedencia del ángulo), así mismo, es posible asignar el peso c_i al nodo x_i , donde c_i es el valor económico del bloque que representa x_i , entonces, el problema de descubrir un tajo final es equivalente a acertar lo que se conoce como un cierre de grafo máximo en G (cierre de grafo), donde, es un subconjunto $V' \subset V$ del conjunto de nodos, tal que ninguna arista va de un nodo en V' a un nodo en $V - V'$.

Figura 32

Cierre de un grafo



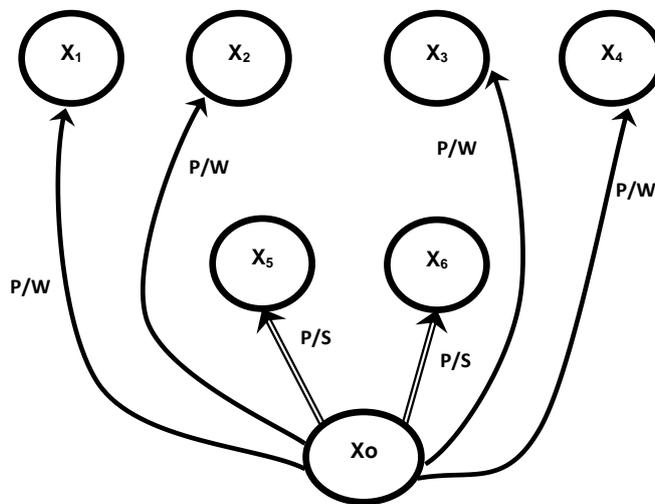
Nota. x_1, \dots, x_6 representa los bloques del modelo de recursos

La definición de un cierre del grafo en G figura un tajo físicamente factible considerando la restricción del ángulo o pendiente, a partir de un conjunto de

nodos dentro de los límites del tajo, entonces, se induce que existe una correspondencia uno a uno entre los diseños de límites de factibles y los cierres de grafos. Por lo tanto, el cierre de grafo de peso máximo es el cierre de grafo $V' \subset V$ tal que es maximizado. $\sum v \in V' cv$ que corresponde a los límites del tajo definitivo.

Figura 33

Grafo Ficticio y Arcos



Nota. Grafo G con el nodo ficticio x_0 y arcos adicionados desde el nodo ficticio x_0 a todos los demás nodos.

2.2.2.3.1. Definiciones para el algoritmo de Lerchs y Grossmann

Un grafo dirigido $G = (X, A)$ está definido por un conjunto X de elementos llamados los vértices de G , junto con un conjunto A de pares ordenados de elementos $a_i = (x, y)$, llamados los arcos de G . El grafo G también define una función denominada T mapeando los elementos de X en X y tal que: $(x, y) \in A \Leftrightarrow y \in T x$.

Un camino es una secuencia de arcos (a_1, a_2, \dots, a_n) tal que el nodo final de cada arco corresponde al vértice inicial del arco sucesor. Un circuito es un camino en el cual el vértice inicial coincide con el vértice final. Un borde, $e_i = [x, y]$, de

un grafo G es un conjunto de dos elementos tal que $(x, y) \in A$ o $(y, x) \in A$. Este concepto difiere del concepto de un arco, el cual tiene o implica orientación. Una cadena es una secuencia de bordes $[e_1, e_2, \dots, e_n]$ en el cual cada borde tiene un vértice en común con el borde sucesor. Un ciclo es una cadena en el cual el vértice inicial y final coinciden.

El subgrafo $G(Y)$ de G es un grafo (Y, A_y) definido por un conjunto de vértices de $Y \subset X$ y que contiene todos los arcos que conectan los vértices de Y en G . Un grafo parcial $G(B)$ de G es un grafo (X, B) definido por un conjunto de arcos $B \subset A$, y que contiene todos los vértices de G . Una cerradura de un grafo dirigido $G = (X, A)$ es un conjunto de vértices $Y \subset X$ tal que $x \in Y \rightarrow T x \in Y$. Si Y es una cerradura de G , entonces $G(Y)$ es un subgrafo cerrado de G . Por definición, el conjunto nulo, $Y = \emptyset$, es también una cerradura de G .

Un árbol es un grafo conectado y dirigido $T = (X, C)$ el cual no contiene ciclos. Un árbol raíz es un árbol con un vértice distinguido el cual es la raíz. El grafo obtenido al eliminar un arco a_i en un árbol raíz T tiene dos componentes. El componente $T_i = (X_i, A_i)$ el cual no contiene la raíz del árbol, este es llamado una rama de T . La raíz de la rama es el vértice de la rama que esta adyacente a el arco a_i . Una rama es un árbol en sí, y las ramas de una rama son llamadas ramitas.

2.2.2.3.2. *El problema*

Dado un grafo dirigido $G = (X, A)$ y para cada vértice X_i un valor numérico $m_i \geq 0$, llamado el valor de X_i , se encuentra una cerradura Y del G con valor máximo. En otras palabras, encontrar el conjunto de elementos $Y \subset X$ tal que:

$$x_i \in Y \rightarrow T x_i \in Y$$
$$M_y = \sum_{x_i \in Y} m_i \text{ es máximo}$$

Una cerradura con valor máximo también es llamada una cerradura máxima.

2.2.2.3.3. *El algoritmo*

Inicialmente al grafo G se agrega un nodo raíz x_0 y un arco (x_0, x_1) . El algoritmo empieza con la construcción de un árbol T^0 en G . Luego, T^0 es transformado en sucesivos arboles T^1, T^2, \dots, T^n , siguiendo las reglas dadas hasta que no existe transformación posible. La cerradura máxima luego está dada por el conjunto de ramas muy bien identificadas en el árbol final. Los árboles construidos durante el proceso iterativo están caracterizados por un número de propiedades.

Cada borde e_k (arco a_k) de un árbol T define una rama, denotado por $T_k = (X_k, A_k)$. También, es conveniente escribir x_k para la raíz de la rama T_k . Se dice que el borde e_k (arco a_k) soporta la rama T_k . El valor M_k de una rama T_k es la suma de los valores de todos los vértices de T_k . Este valor está asociado con el borde e_k (arco a_k) y decimos que el borde e_k (arco a_k) soporta un valor M_k .

En un árbol T con raíz X_0 , un borde e_k (rama T_k) esta caracterizado por la orientación del arco a_k con respecto a x_0 ; e_k es llamado un “p-edge” (plus-borde) si el arco a_k apunta hacia la rama T_k , eso es, si el vértice terminal de a_k es parte de la rama T_k . Entonces, T_k es llamado un “p-branch” (plus-rama). Si el arco a_k apunta en sentido contrario de la rama T_k , entonces e_k es llamado un “m-edge” (minus-borde) y T_k una “m-branch”. Similarmente, todas las ramitas de una rama pueden ser divididos en 2 clases: “p-twigs” (plus-ramita) y “m-twigs” (minus-ramita). También, se distingue entre bordes “Strong” y “Weak”. Un “p-edge” (plus-borde) es “Strong” si soporta un valor positivo; un “m-edge” es “Strong” si soporta un valor que es nulo o negativo. Todos los bordes que no son “Strong”

son denominados “Weak”. Se dice que un vértice x_i es strong si existe por lo menos un borde “Strong” en la cadena de T uniendo x_i a la raíz x_0 . Los vértices que no son “Strong” son denominados “Weak”. Finalmente, un árbol esta “Normalizado” si la raíz x_0 es común a todos los bordes “Strong”. Cualquier árbol T de un grafo G puede ser normalizado reemplazando el arco (x_k, x_l) de un “p-edge” “Strong” con un arco (x_0, x_l) , el arco (x_q, x_r) de un “m-edge” “Strong” con un arco temporal (x_0, x_q) y repetir el proceso hasta que todos los bordes strong tengan el nodo x_0 como uno de sus extremidades.

El árbol en la Figura 32 ha sido obtenido al normalizar el árbol en la Figura 33. Podemos ver que todos los bordes temporales son “p-edges”, todos los bordes strong de un árbol normalizado serán “p-edges”.

Tabla 1

Categorización de arcos en un grafo dirigido

Caso	Dirección	Soporte	Etiqueta
1	Plus	Positivo	Strong
2	Plus	Nulo o Negativo	Weak
3	Minus	Positivo	Weak
4	Minus	Nulo o Positivo	Strong

Nota. Si la dirección de un arco sale del nodo raíz, entonces es “Plus” de lo contrario es

“Minus”. Además, si el valor del nodo es positivo, entonces el arco es “Strong”

2.2.2.3.4. Pasos del algoritmo

Inicialmente se crea un árbol normalizado T^0 , con la ayuda de la tabla x, conectando todos los nodos al nodo raíz, luego:



1. Si existe un arco (x_k, x_l) tal que $x_k \in Y_t$, y $x_l \in X - Y_t$, entonces ir al paso 2, de lo contrario ir al paso 4.
2. Determinar x_m , la raíz del “strong branch” que contiene a x_k . Construir el árbol T^s reemplazando el arco (x_0, x_m) de T^t con el arco (x_k, x_l) . Ir al paso 3.
3. Normalizar T^s . Esto conduce a T^{tx1} . Ir al paso 1.
4. Terminar. Y^t es un “closure” máximo de G , figura 4.

El arco (X_0, X_m) soporta un ramal con peso >0 ($wB > 0$). La transformación sólo se hace en la cadena $[X_m, \dots, X_k, X_l, \dots, X_p, X_0]$. Se tienen las siguientes transformaciones:

Para un arco E_i en la cadena $[X_m, \dots, X_k]$: $W_{E_i} = wB - W_{E_0}$ (arcos cambian status, arcos rojos). W_{E_0} es el valor del arco antes de la transformación.

Para el arco $[X_k, X_l]$: $W_k = wB$ (el status es siempre “m”)

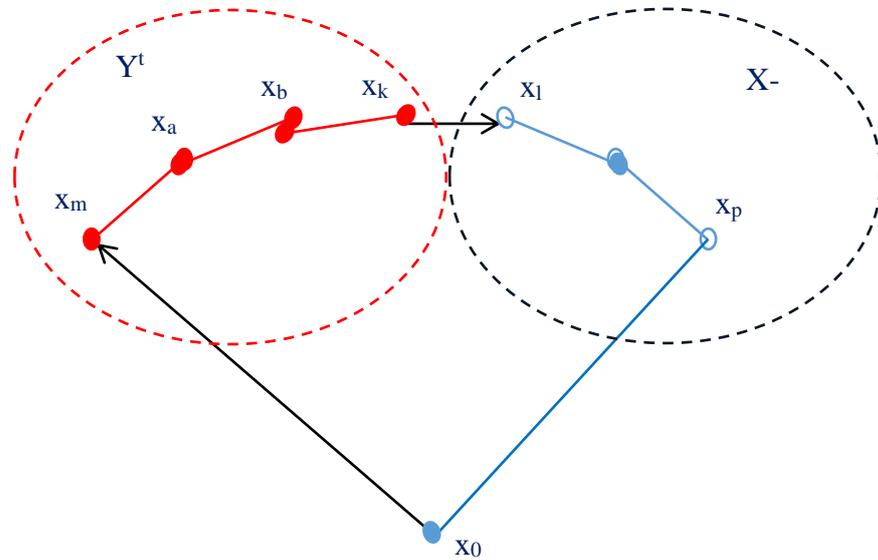
Para un arco E_i en la cadena $[X_l, \dots, X_p, X_0]$: $W_{E_i} = wB + W_{E_0}$ (arcos no cambian status, arcos azules)

Si se tienen arcos strong no conectados a la raíz (ejemplo arco $[X_a, X_b]$ en la figura anterior), eliminar tal arco y conectar el sucesor con la raíz con un nuevo arco. Como regla remover arcos strong uno por uno empezando del primer arco strong encontrado en la cadena $[X_m, \dots, X_0]$.

Para cada arco en la cadena $[X_b, \dots, X_0]$ (líneas verdes siguiente figura), actualizar los pesos usando la siguiente relación: $W_{E_i} = W_{E_0} - wM$. Luego buscar el siguiente arco strong en la cadena $[X_b, \dots, X_0]$ y repetimos el proceso hasta que el último arco strong se haya removido de la cadena.

Figura 34

Pasos del algoritmo de Lerchs y Grossmann



Fuente. Lerchs, H., and Grossmann, I.F. 1965, "Optimum design of open pit mines", Canadian Institute of Mining Transactions, Vol. 68, p. 17-24.

2.2.2.3.5. *Tiempo de computación*

No cabe duda de que el algoritmo de Lerchs-Grossmann requiere más tiempo de cálculo que otros métodos para alcanzar una solución; ya que este algoritmo tiene un Big O, $O(n^3)$, como se muestra en el siguiente pseudocódigo escrito en Python,

Figura 35*Seudocódigo de tiempo de computación de algoritmo de LG*

```
# 1.- Primer bucle –  $O(n)$ 
while not cerraduraFinal:
    cerraduraBanco = false
    # 2.- Segundo bucle -  $O(n)^2$ 
    while not cerraduraBanco:
        strongNodos = arcosTabla [  $x_0, x_n$ ]
        # 3.- Tercer bucle -  $O(n)^3$ 
        For each strongNodos:
            weakNodes = conoTemplate [  $x_0, \dots, x_n$ ]
        End For
        if weakNodes = []:
            cerraduraBanco = true
        else:
            funciónTrasformarArbol () #Paso 2 LG
    funciónNormalizarArbol () #Paso 3 LG
    nodosMinados ()
    if bancoActual = bancoInferior:
        cerraduraFinal = true
    else:
        bancoActual = bancoActual - 1
```

Debido a este problema, se han desarrollado muchos algoritmos alternativos para reducir el tiempo de cálculo. El mayor tiempo de cálculo es el precio que hay que pagar por un diseño de pit óptimo garantizado. Con la llegada de ordenadores más rápidos, el tiempo de cálculo se está volviendo irrelevante, siempre que el software se adapte al hardware mejorado. Los ordenadores modernos ya pueden ejecutar el algoritmo de Lerchs-Grossmann para modelos de bloques, que hace sólo unos años habrían requerido grandes ordenadores centrales.

2.2.2.3.6. Complejidad

Otra desventaja que se atribuye al método es la complejidad de las matemáticas y la programación necesarias. Este argumento sostiene implícitamente que ambas están fuera del alcance del ingeniero de minas medio. Sin embargo, ésta no es una razón válida para no utilizar el método. No es

necesario que el "ingeniero tenga un conocimiento detallado del algoritmo una vez que se ha codificado en un paquete de software validado. Pocos ingenieros, por no decir ninguno, entenderían los algoritmos empleados en los paquetes CAD o de contorneado, pero esto no impide su uso generalizado. No obstante, es esencial que el usuario conozca las limitaciones de las aplicaciones informáticas de este (o cualquier otro) algoritmo.

2.2.3. Determinación del pit final

Durante la vida de una mina, el plan para el pit final puede cambiar debido a la disponibilidad de más información, como resultados de sondajes, nuevos precios de los metales y avances tecnológicos que pueden reducir costos, entre otros factores. Por lo tanto, es importante estudiar diferentes escenarios económicos, desarrollar planes alternativos y considerar diferentes ritmos de extracción.

Uno de los criterios para determinar el pit final implica definir dos tipos de secuencias que representen el mejor y el peor caso posible de extracción, como se detalla en los siguientes párrafos:

- Extracción tipo best case:

La extracción tipo best case se refiere a una secuencia de extracción en la que se prioriza la obtención temprana del mineral, lo que genera altos retornos iniciales. Sin embargo, esta secuencia puede no ser factible debido a la falta de espacios óptimos para su desarrollo. A pesar de esto, desde el punto de vista del valor, esta es considerada la mejor forma de extracción. Es importante tener en cuenta que la viabilidad técnica y operativa también deben ser consideradas en la planificación minera, y no solo el aspecto económico.

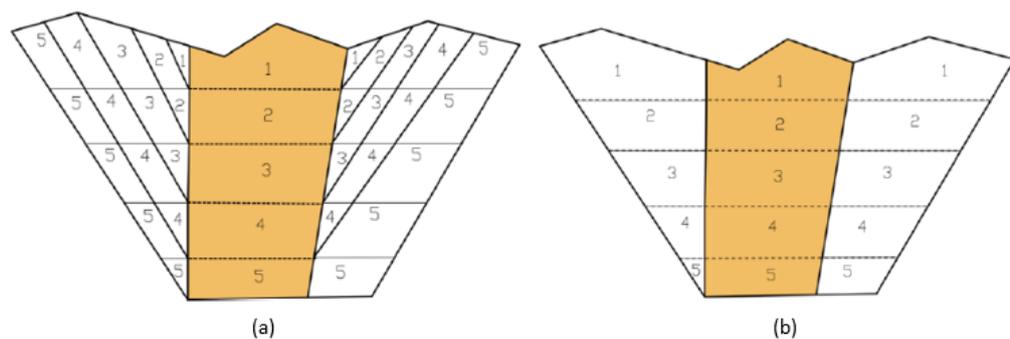
- Extracción tipo worst case:

La extracción tipo "Worst Case" es una secuencia en la planificación minera que implica extraer grandes cantidades de material estéril al principio, debido a la presencia de una gran cantidad de "ganga" o "desecho". Esto resulta en un retraso en la extracción de mineral y en una generación tardía de ingresos. Aunque esta secuencia es considerada la peor en términos de generación de ingresos iniciales, puede ser necesaria en situaciones donde no hay otras opciones factibles disponibles. Es importante evaluar cuidadosamente la viabilidad técnica y operativa de esta secuencia antes de implementarla.

Tengamos presentes que para una buena elección de "Pit final" se enfocará más que todo en los objetivos que tendrá cada compañía, ya que estos dos tipos de extracciones solamente nos servirá de guía para ver el mejor y peor valor que se pueda extraer. El plan siempre será obtener un óptimo escenario estableciendo delimitaciones operacionales.

Figura 36

Secuencias de extracción Best case y Worst case



Nota. (a) Best case y (b) Worst case asociado a los periodos de extracción. Fuente: Loor, 2020

Al analizar los dos casos se debe definir el tajo final, enfocada en los objetivos de la compañía y el criterio del planificador; enfocados principalmente en (Suarez, 2017):

- Alcanzar más óptimo valor de NPV (Net Present Value)



- Cantidad de material a ser extraído
- Razón de “waste” / “mineral”
- Vida de la mina
- Volatilidad de precios y costos
- Recuperaciones

2.2.4. Métodos de evaluación económica

La Evaluación Económica de Proyectos es un proceso que busca proporcionar una evaluación cuantitativa para tomar decisiones sobre la viabilidad de un proyecto. Incluye varios aspectos, como los técnicos, económicos, financieros y el riesgo de inversión. Estos aspectos son fundamentales para determinar si un proyecto es factible desde diferentes perspectivas. Los aspectos técnicos se refieren a la viabilidad técnica y operativa del proyecto, mientras que los aspectos económicos se centran en la rentabilidad y los beneficios económicos que se pueden obtener. El aspecto financiero se ocupa de la estructura financiera del proyecto y la capacidad de generar flujo de efectivo para cubrir los costos e inversiones. Por último, el riesgo de inversión evalúa los posibles riesgos y la probabilidad de éxito del proyecto.

El aspecto económico del análisis de proyectos se centra en el modelo económico de la inversión, que considera el valor temporal del dinero o valor cronológico del dinero. Este concepto reconoce que el dinero tiene un costo de utilización significativo, que puede manifestarse a través de intereses que deben pagarse o como el costo de oportunidad de invertir en otros proyectos. Normalmente, se trabaja con una tasa de interés que se expresa como un porcentaje del capital y se refiere a un período de tiempo determinado. Este

análisis permite evaluar la viabilidad económica de un proyecto al considerar los flujos de dinero positivos y negativos a lo largo del tiempo.

2.2.4.1. Valor actual neto

El Valor Actual Neto (VAN) es un concepto financiero utilizado para determinar la rentabilidad de una inversión, proyecto o negocio. Se calcula mediante el descuento de los flujos de caja futuros de la inversión al valor presente. El VAN se utiliza comúnmente en finanzas corporativas, como en banca de inversión y contabilidad, para tomar decisiones estratégicas. La fórmula del VAN implica calcular los flujos de caja, el número de períodos, la tasa de descuento y la inversión inicial. Un VAN positivo indica una oportunidad rentable, un VAN negativo sugiere un proyecto no rentable y un VAN cero significa que la inversión es neutral [1][2][3].

$$VAN = -I_0 + \sum_{t=1}^n \frac{F_t}{(1+k)^t} = -I_0 + \frac{F_1}{(1+k)^1} + \frac{F_1}{(1+k)^2} + \dots + \frac{F_n}{(1+k)^n}$$

Donde:

F_t es el flujo de dinero en cada periodo

I_0 es la inversión realizada al momento inicial ($t = 0$)

n , es el número de periodos de tiempo

k , es el tipo de descuento o tipo de interés exigido a la inversión

2.2.4.2. Tasa interna de retorno

La Tasa Interna de Retorno (TIR) es un indicador financiero que mide la rentabilidad de una inversión. Se define como la tasa de interés que iguala el valor presente de los flujos de efectivo futuros de un proyecto con la inversión inicial. En otras palabras, es la tasa de descuento que hace que el Valor Actual Neto (VAN) sea igual a cero.

La TIR es una medida importante en la toma de decisiones de inversión, ya que permite comparar la rentabilidad de diferentes proyectos. Si la TIR es mayor que la tasa de descuento utilizada, el proyecto se considera rentable. Por el contrario, si la TIR es menor que la tasa de descuento, el proyecto se considera no rentable.

En resumen, la TIR es una herramienta útil para evaluar la rentabilidad de una inversión y tomar decisiones informadas sobre qué proyectos seguir adelante.

$$\begin{aligned}VAN &= -I_0 + \sum_{t=1}^n \frac{F_t}{(1 + TIR)^t} \\ &= -I_0 + \frac{F_1}{(1 + TIR)^1} + \frac{F_1}{(1 + TIR)^2} + \dots + \frac{F_n}{(1 + TIR)^n} = 0\end{aligned}$$

2.2.4.3. Periodo de retorno de la inversión

El Periodo de Retorno de la Inversión (PRI) o Payback Time es el tiempo que tarda una empresa en recuperar su inversión inicial en un proyecto. Se calcula en base a las entradas de efectivo y es un método estático utilizado para medir proyectos. La fórmula para calcular PRI es:

$$PRI = a + \left[\frac{(b - c)}{d} \right]$$

donde:

a, es el año inmediato anterior en el que se recupera la inversión

b, es la inversión inicial

c, es la suma de los flujos de caja anteriores hasta la fecha de recuperación de la inversión

d, es el flujo de caja neto del año en que se satisface la inversión.

El PRI es adecuado para situaciones de incertidumbre o limitaciones financieras, ya que un período de recuperación más corto indica un riesgo menor.



Es una métrica importante para la toma de decisiones, ya que permite a las empresas evaluar el tiempo que tardarán en recuperar su inversión inicial.

2.2.5. Diseño de taludes en minería a cielo abierto

El objetivo último de una explotación minera, ya sea por métodos subterráneos o de minería a cielo abierto, es extraer mineral con beneficios. La rentabilidad de una explotación depende de una planificación técnica y económica adecuada en función de todos los parámetros pertinentes. Uno de estos parámetros es el ángulo de inclinación del tajo. La determinación de la inclinación del tajo es de vital importancia en la planificación de la explotación, ya que afecta al tamaño y la forma del tajo final. Una pendiente más pronunciada disminuye la tasa de extracción y, por tanto, aumenta la rentabilidad de la mina. Por otro lado, la pared del tajo debe permanecer estable y el uso de una pendiente más pronunciada aumenta la probabilidad de fallo. Por lo tanto, un plano de mina óptimo debe tener la pendiente final más pronunciada que se mantenga estable durante toda la vida útil de la mina.

El aumento o la disminución del ángulo de inclinación de una mina a cielo abierto de mediana o gran envergadura alterará significativamente el coeficiente de extracción, es decir, la cantidad de residuos que deben retirarse para extraer una cantidad unitaria de mineral. En una mina a cielo abierto con las siguientes dimensiones y gravedad específica, habría que retirar un total de 23 millones de toneladas adicionales de residuos si se redujera la pendiente media en 5°: 1.900 m de longitud de cresta a cresta, 75 m de anchura del fondo de punta a punta, 150 m de profundidad vertical y 2,65 de gravedad específica del material (Seegmiller, 1979).



Por otra parte, un diseño inadecuado de los taludes puede poner en peligro toda la actividad minera. La interrupción de las operaciones mineras causada por la inestabilidad de los taludes puede ser grave. Algunas consecuencias son la pérdida de mineral, los costes adicionales debidos a la limpieza de las fallas, los retrasos en la producción y los daños en los equipos y la probabilidad de lesiones. Se han registrado muchos fallos en taludes como consecuencia de un diseño inadecuado de los mismos (Seegmiller, 1979). Por ejemplo, un fallo a gran escala que se produjo en la mina Afton en 1986 (Reid, y Stewart, 1986).

En general, las pendientes de los tajos se diseñan con la mayor inclinación posible para reducir el coeficiente de extracción. Singh (1986) demostró la importancia de los ángulos de inclinación de los tajos en el diseño y la economía de los tajos abiertos y descubrió que una inclinación más pronunciada no sólo produce un mayor valor actual neto, sino también un periodo de amortización más corto. La determinación de los taludes de los tajos es un paso crítico durante el proceso de planificación de los tajos. La mayor pendiente necesaria para minimizar el índice de extracción debe sopesarse con las pendientes más planas necesarias para garantizar la estabilidad. El diseñador de taludes debe, por tanto, conciliar estos dos requisitos contradictorios. También existe un problema circular en la determinación de las pendientes de los tajos, claramente expuesto por Call y Savely (1990):

"Por un lado, el diseñador de taludes necesita la posición, la orientación y la altura de las paredes del tajo para diseñar los taludes. Por otro, el planificador de la mina necesita los ángulos de los taludes para diseñar los límites del tajo".

Debido a la complejidad y a las incógnitas asociadas al macizo rocoso, cualquier decisión en ingeniería de rocas requiere una cantidad considerable de



experiencia. Es muy importante reconocer los factores que afectan a la estabilidad de los taludes durante las fases de diseño. Estos factores incluyen las condiciones geológicas, las características estructurales, las condiciones de las aguas subterráneas, la resistencia al corte de las discontinuidades, la geometría de los taludes y la técnica de excavación utilizada en la creación del talud. La planificación de la estabilidad de taludes implica las siguientes actividades:

1. Recolección de datos relevantes.
2. Identificación de las características geológicas críticas.
3. Determinación del parámetro de resistencia al corte de las discontinuidades.
4. Determinación de las condiciones del agua subterránea y de la presión de poros que actúa sobre las discontinuidades críticas.
5. Identificación de los modos de fallo potenciales.
6. Análisis de los modos potenciales de fallo utilizando un método apropiado.

Para diseñar ángulos de talud en minería a cielo abierto, el primer paso es detectar un modo de fallo potencial y, a continuación, utilizar uno de los métodos de estabilidad de taludes para evaluar la estabilidad. Para ello hay que juzgar si el talud es estable o no y tomar las decisiones pertinentes. Existen varios métodos que pueden utilizarse para evaluar la estabilidad o inestabilidad de un talud determinado. Son los siguientes: proyección estereográfica, métodos de equilibrio límite, métodos probabilísticos y métodos numéricos.

Los taludes de los tajos en el algoritmo de diseño óptimo de tajos se consideran restricciones y se definen en términos de bloques que deben eliminarse para proporcionar acceso a un bloque concreto en el modelo de bloques. No



existen pruebas significativas de la incorporación de información geotécnica en el diseño de taludes de tajo en los algoritmos de diseño óptimo de tajos. La razón principal es que, o bien utilizan un modelo, o un conjunto de bloques, para definir los taludes mineros, o bien utilizan taludes de tajo predefinidos que se supone que ya han sido diseñados.



CAPÍTULO III

MATERIALES Y METODOS

Para el presente trabajo de investigación se usó scripts en Python para obtener el modelo económico desde el modelo de corto plazo de geología de dimensiones 10 x 10 x 15m, el cual fue rebloqueado a 20 x 20 x 15m. Además, se usó el Software Optimining de propiedad de Yovani Achata Flores con el cual se optimizó el pit final del Tajo Pampacancha. Por último, se recopiló información necesaria del Reporte Técnico NI 43-101 de fecha 29 de marzo del 2021.

3.1. MATERIAL DE ESTUDIO

Para alcanzar los objetivos mencionados en el presente estudio se tomaron como input o datos de entrada el modelo de bloques (ley de mineral Cu, Au, Ag y Mo), recuperación metalúrgica, parámetros económicos (costo de mina, planta, G&A, venta, tratamiento y refinación), precio de los metales y parámetros geotécnicos (sectores geotécnicos). El software requerido para realizar la implementación del algoritmo de LG con ángulo variable es:

- Objetivo 1: Se usa el lenguaje de programación Python 3.4 para hacer la implementación original de LG.
- Objetivo 2: Se usa geometría, funciones trigonométricas y programación para implementar ángulo variable al algoritmo LG original del objetivo 1.
- Objetivo 3: Se usa el modelo de bloques de corto plazo del Tajo Pampacancha, Mina Constancia, Hudbay Perú S.A.C para optimizar el pit final de dicho tajo.

3.1.1. Modelo de Bloques

El modelo de bloques utilizado en el estudio de caso es un yacimiento de cobre skarn que contiene elementos de valor económico de Cu (metal principal), Ag y Au (metales secundarios) contenidos en los diferentes sectores geotécnicos

de la mineralización. El modelo de recursos consta de un total de 2,116,800 bloques, con dimensiones de 10m en las direcciones Este y Norte, y 15m en la dirección de Elevación. Las longitudes totales son de 2,950m y 2,750m en las direcciones Este y Norte, respectivamente, como se muestra en la Tabla 2.

Tabla 2

Modelo de bloques de Tajo Pampacancha 10 x 10 x 15m

Dirección	Mínimo	Máximo	Dimensión Bloque	# Bloques
Este (m)	203,680	205,360	10	168
Norte (m)	8,396,140	8,398,240	10	210
Elevación (m)	3,600	4,500	15	60

Además, para minimizar el número de bloques se realizó un rebloqueo considerando dimensiones de 20m en las direcciones Este y Norte, como se muestra en la Tabla 3.

Tabla 3

Modelo de bloques rebloqueado de Tajo Pampacancha 20 x 20 x 15m

Dirección	Mínimo	Máximo	Dimensión Bloque	# Bloques
Este (m)	203,680	205,240	20	78
Norte (m)	8,396,300	8,397,700	20	70
Elevación (m)	3,975	4,500	15	35

La ley máxima de Cu de 3.23%, con valores de Ag y Au de 19.033 ppm y 1.580 ppm, respectivamente. Es importante destacar que las recuperaciones varían según el tipo de roca, alteración, litología y mineralización. El modelo está clasificado en recursos medidos, indicados e inferidos, según se detalla en el Anexo 1, además, para el análisis y valorización se consideran los valores mínimos y máximos de cada variable, que se muestran en la Tabla 4.

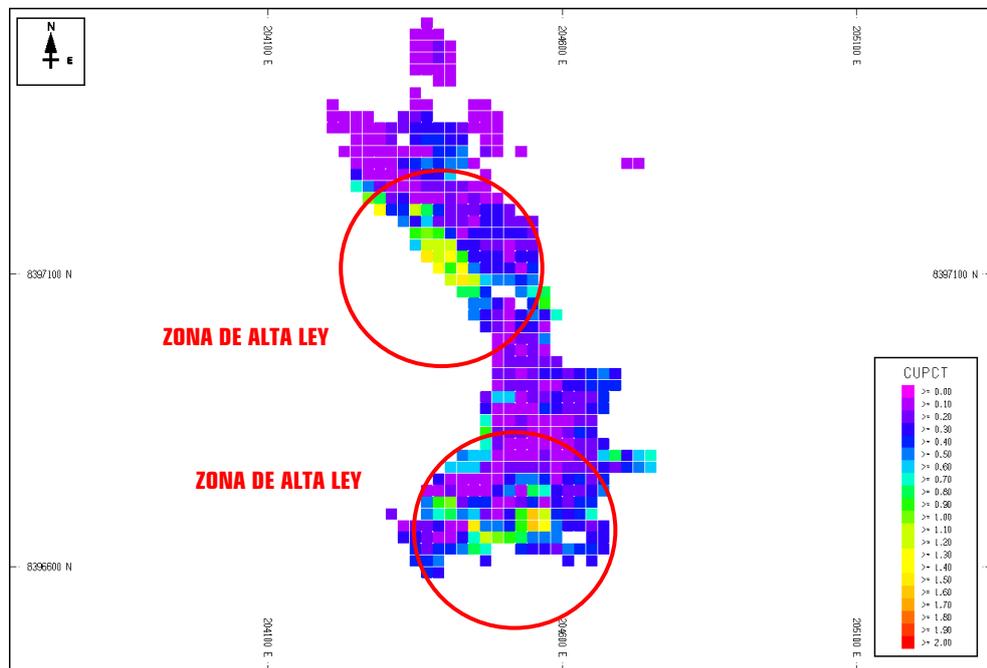
Tabla 4*Variables del modelo de bloques de Tajo Pampacancha*

Variable	Descripción	N° Decimales	Actual Min.	Actual Max.	N° Valores Distintos	Método Rebloqueo
TOPO%	Porcentaje de Topografía (%)	3	0.01	100	>100	Promedio
CUPCT	Ley de Cobre (%)	5	0	3.23	>100	Promedio
AUPPM	Ley de Oro (PPM)	3	0	7.92	>100	Promedio
AGPPM	Ley de Plata (PPM)	3	0	942.3	>100	Promedio
MOPCT	Ley de Molibdeno (%)	5	0	1.24	>100	Promedio
PBPCT	Ley de Plomo (%)	5	0	1.43	>100	Promedio
FEPCT	Ley de Hierro (%)	5	0	45.16	>100	Promedio
ZNPCT	Ley de Zinc (%)	5	0	8.32	>100	Promedio
SG	Densidad (ton/m3)	2	2.2	4.64	>100	Promedio
LITO	Litología	1	2	9	4	Mayoría
ALTER	Alteración	1	1	10	5	Mayoría
MATPS	Tipo de Material	1	1	26	13	Mayoría
OTYPE	Tipo de Material Ore	1	1	99	8	Mayoría
%OX	Porcentaje de Óxidos (%)	2	0	163.81	>100	Promedio
RECCU	Recuperación del Cu (%)	2	0	90.50	>100	Promedio
CLASS	Categoría de Recurso	1	1	3	3	Mayoría
NSR2	Net Smelting Return	3	0	218.59	>100	Promedio
CUSS	Porcentaje de Sulfuro (%)	2	0	145.60	>100	Promedio

Se presenta la visualización tridimensional de la geometría del modelo de bloques correspondiente a los recursos medidos e indicados. Figura 37, Figura 38 y Figura 39 representan una zona de alta ley en la zona noroeste y sur del yacimiento, un afloramiento en ese punto y una aproximación a la superficie a una profundidad media de 100 m, respectivamente. Las dimensiones del yacimiento son aproximadamente 1,000 m de longitud, 600 m de anchura y 250 m de profundidad.

Figura 37

Vista en Planta del Modelo de Bloques en 3D



Nota. El modelo de bloques en 3D muestra la ley de %Cu de recursos medidos e indicados

Figura 38

Vista lateral Izquierdo del modelo de bloques en 3D

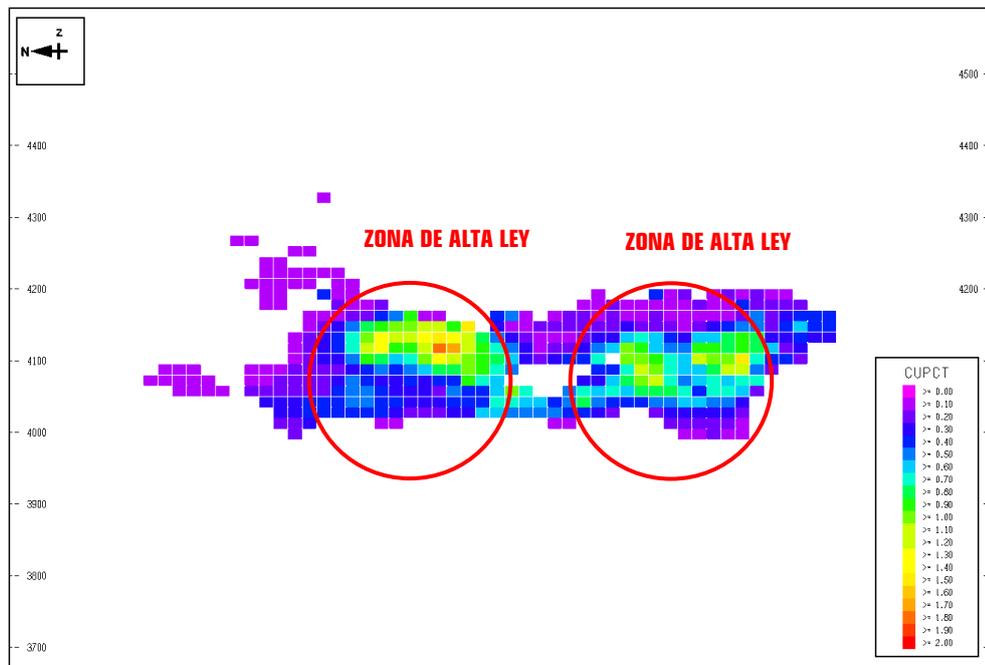
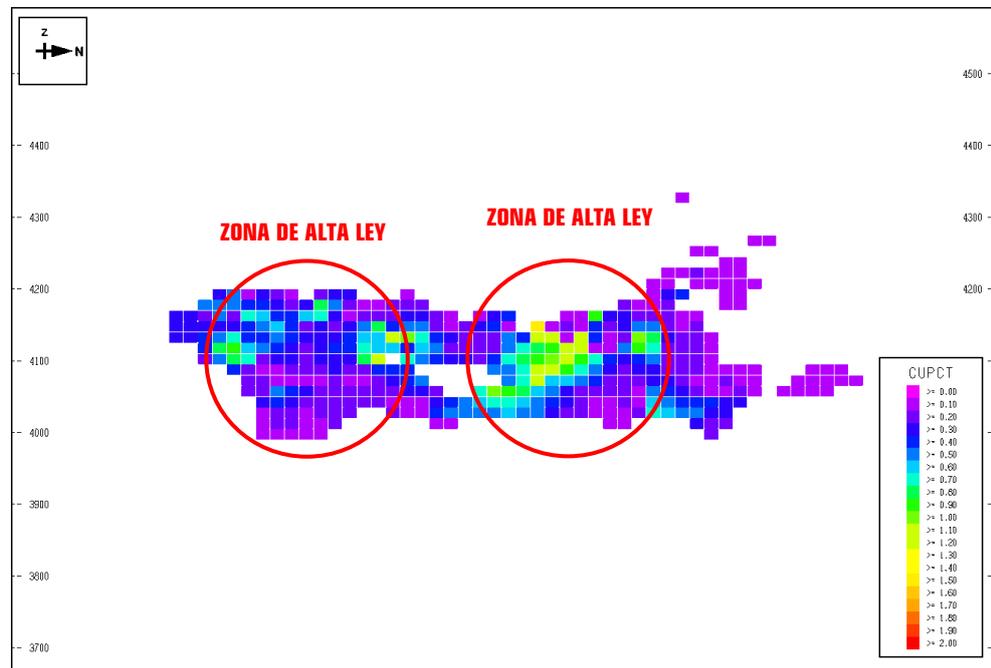


Figura 39

Vista lateral Derecho del modelo de bloques en 3D



3.1.2. Valorización del Modelo

La valorización del modelo de bloques se realizó con la ayuda de un script en Python, Anexo 1, utilizando como datos de entrada el valor económico de los diferentes tipos de mineral y desmonte para la optimización del tajo. El beneficio descontado de cada bloque se estimó a partir de parámetros económicos, recuperaciones metalúrgicas de los diferentes tipos de mineral, número de bancos por año, y tasa de descuento del 10%.

El costo de mina base es 1.85 \$/t en el nivel de referencia 4 305m, con un incremento por banco hacia arriba y debajo de 0.004 y 0.010 \$/t, respectivamente. Esto debido al incremento de la distancia de acarreo. El costo de procesamiento es de 4.54 \$/t y 1.60 \$/t el costo por G&A. Se considero el precio de metal fijo y conservador.

3.1.2.1. Recuperación metalúrgica

La recuperación metalúrgica del metal y concentrados fue calculada en función de las siguientes formulas en función del tipo de mineral: Hipógeno, Supérgeno, Mixto, Skarn y High Zinc, como se muestra en la Tabla 5.

Tabla 5

Modelo de recuperación metalúrgica

CU Recuperación & Ley de Concentrado	
<u>CU Recuperación</u>	
Todo Otypes	$88.73\% - 0.02\%*P80 + 13.19*Cu\% - 30.87*(Zn\%+Pb\%) - 36.00*(CuSS\%/Cu\%)$
<u>Cu Ley de Concentrado</u>	
Hypogene	$22.84\% + 9.40*Cu\% - 17.48*Zn\% - 0.30*Fe\%$
Supergene & Mixed	$24.35\% + 5.05*Cu\% - 8.45*Zn\% - 0.42*Fe\%$
Skarn & HiZinc	$23.06\% + 4.60*Cu\% - 5.64*Zn\% - 0.17*Fe\%$
Recuperación de otros metales	
<u>Mo Recuperación</u>	
Hypogene	$[1.65*CuRec\% + 43.32*(Mo\%/Zn\%) - 96.93\%]*0.66$
Supergene	$[1.65*CuRec\% + 43.32*(Mo\%/Zn\%) - 96.93\%]*0.66$
Skarn	$[1.65*CuRec\% + 43.32*(Mo\%/Zn\%) - 96.93\%]*0.40$
Skarn Pampa	$[1.65*CuRec\% + 43.32*(Mo\%/Zn\%) - 96.93\%]*0.20$
Mixed & HiZinc	0
<u>Ag Recuperación</u>	
Todo Otypes	$1.30*CuRec\% - 5.30*(Zn\%+Fe\%)/Ag(g/t) - 37.52\%$
<u>Au Recuperación</u>	
Todo Otypes	$0.54*CuRec\% + 99.94*Au(g/t) + 69.55*Pb\% - 46.21*Zn\% - 3.01$
<u>Zn Ley de Concentrado</u>	
Todo Otypes	$21.92*(Zn\%/Cu\%) - 0.88$
<u>Pb Ley de Concentrado</u>	
Todo Otypes	$10.6*(Pb\%/Cu\%) + 0.187$
<u>Ag Ley de Concentrado</u>	
Todo Otypes	$Ag\ ozs*31.1034/Cu\ Conc\ tonnes$
<u>Au Ley de Concentrado</u>	
Todo Otypes	$Au\ ozs*31.1034/Cu\ Conc\ tonnes$
<u>Zn Recuperación</u>	
Todo Otypes	$Cu\%*CuRec\%*Zn\%\ Conc / Cu\%Conc.*Zn\%$
<u>Pb Recuperación</u>	
Todo Otypes	$Cu\%*CuRec\%*Pb\%Conc / Cu\%Conc.*Pb\%$

Fuente: Reporte Técnico NI 43-101

3.1.2.2. Net Smelting Return

El NSR es definido como el beneficio contractual esperado. Es calculado en función a los contenidos metálicos, recuperaciones metalúrgicas y los costos en asociados a su comercialización (No se considera el costo de mina ni de procesamiento). Para el caso de Mina Constancia se definió un modelo de valorización de bloque en función al NSR considerando costo de mina, proceso, venta, tratamiento y refinación.

$$\text{Revenue Cu} = \text{Precio Metal}$$

$$* (\text{Metal Contenido Pagable por Tonelada Seca})\text{Cu}$$

$$\text{Revenue Mo} = \text{Precio Metal}$$

$$* (\text{Metal Contenido Pagable por Tonelada Seca})\text{Mo}$$

$$\text{NSR} = \text{Total Revenue}$$

$$- (\text{Mining} + \text{Process} + \text{Selling} + \text{Treatment} + \text{Refining})\text{Cost}$$

Para realizar la valorización del modelo de bloques se estimaron los costos para el proyecto basado en resultados operacionales del tajo Constancia y en otros, se consideró el benchmarking con operaciones y proyectos de similares características. Los cuales se detallan a continuación en Tabla 6, Tabla 7, Tabla 8, Tabla 9, Tabla 10 y Tabla 11.

3.1.2.2.1. Costo Mina

Tabla 6

Costo de Minado

Costo de Minado	Unidad	Valor
<u>Costo de Minado Base</u>		
Ore	\$/t	1.85
Waste	\$/t	1.55
<u>Costo Variable por Banco</u>		
Elevación Base	-	4,305
Arriba	\$/t	0.004
Abajo	\$/t	0.010

Fuente: Reporte Técnico NI 43-101



3.1.2.2.1. Costo Proceso

Tabla 7

Costo de Procesos y Gastos Administrativos

Costo Proceso y G&A	Unidad	Valor
<u>Costo Proceso</u>		
Ore	\$/t Ore	4.54
<u>Costo G&A</u>		
Ore	\$/t Ore	1.60
Waste	\$/t	-

Fuente: Reporte Técnico NI 43-101

3.1.2.2.1. Costo Venta

Tabla 8

Costo de venta de concentrado de cobre y molibdeno

Costo Venta Concentrado	Unidad	Valor
Cu Road Transport	\$/wmt	51.23
Cu Port Charges	\$/wmt	26.00
Cu Shipping Cost	\$/wmt	25.00
Cu Insurance	\$/wmt	5.00
Mo Road Transport	\$/dmt	204.16
Mo Port Charges	\$/wmt	26.00
Mo Shipping Cost	\$/wmt	0.00
Mo Insurance	\$/wmt	0.00

Fuente: Reporte Técnico NI 43-101

3.1.2.2.1. Costo Tratamiento

Tabla 9

Costo de tratamiento

Costo Tratamiento Concentrado	Unidad	Valor
Cu Smelting	\$/dmt	76.73
Mo Roasting	\$/dmt	1653.00

Fuente: Reporte Técnico NI 43-101

3.1.2.2.1. Costo Refinación

Tabla 10

Costo de refinación por metal

Costo Refinación Concentrado	Unidad	Valor
Cu	\$/lb	0.08
Ag	\$/oz	0.45
Au	\$/oz	4.50
Mo	\$/lb	1.20

Fuente: Reporte Técnico NI 43-101

3.1.2.2.1. Precio de metal

Tabla 11

Precio base de los metales

Precio Metales	Unidad	Valor
Cu	\$/lb	2.8
Ag	\$/oz	18.00
Au	\$/oz	1500.00
Mo	\$/lb	11.00

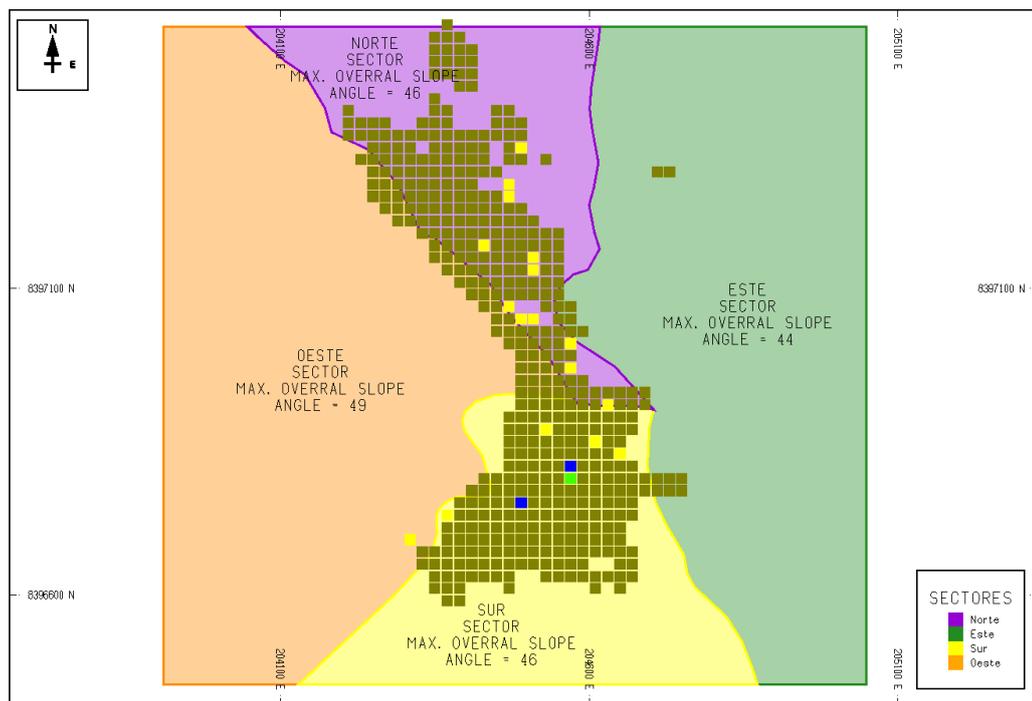
Fuente: Reporte Técnico NI 43-101

3.1.2.3. Parámetros geotécnicos

El modelo de recursos del estudio de caso se divide en 4 dominios geotécnicos, los cuales se encuentran definidos y codificados en el modelo de recursos, tal como se ilustra en la Figura 21. Cada dominio presenta un ángulo específico para determinar la inclinación de la pared del tajo final, según se indica en la Tabla 6. Estos ángulos se utilizan en ambas metodologías de optimización y son necesarios para establecer las precedencias de extracción.

Figura 40

Sectores geotécnicos del Tajo Pampacancha



3.2. METODOLOGÍA

Para demostrar la hipótesis de la presente investigación, en su defecto, para lograr los objetivos de la investigación se hará la revisión bibliográfica de los algoritmos para la optimización del pit final y diferentes métodos para abordar la implementación de ángulos variables. Luego, se procederá con la implementación, en código Python, del algoritmo de Lerchs y Grossmann incorporando ángulos variables con la metodología de Plantilla Cono; haciendo uso de la teoría de grafos y técnicas de programación; recursividad, programación dinámica, etc. La siguiente etapa describirá el método para estimar el conjunto de ángulos provenientes de información geotécnica, como parámetros de entrada para determinar el pit final. A continuación, se describirá la estructura del software, presentación numérica y grafica de los resultados, requerimiento de data y manejo del software. La siguiente etapa se presentará un caso de estudio usando data real de un proyecto a cielo abierto; para la cual se optimizará los límites máximos de minado



a nivel de estudio conceptual; esto con el fin de validar y evaluar el software. Por último, se darán las conclusiones del trabajo de investigación y rumbo para investigaciones a futuro.

3.3. TIPO DE INVESTIGACIÓN

El tipo de investigación elegido para el presente estudio es investigación aplicada, ya que se usarán las formulaciones teóricas ya elaboradas: teoría de grafos para la implementación del algoritmo de Lerchs y Grossmann original y haciendo uso de geometría euclidiana y funciones trigonométricas para la implementación de ángulo variable por el método de Plantillas Cono (Cone Template) para resolver una de las etapas de la planificación minera; problema del pit final en minas a cielo abierto.

3.4. DISEÑO DE INVESTIGACIÓN

El diseño que se usará en esta investigación es del tipo: Experimental, transversal-descriptivo con un enfoque cuantitativo, ya que los datos muestreados como: Costo de mina, costo de planta, costo de refinería y costo de venta, análisis de precios del metal, topografía de la mina, modelo de bloques, etc. Se recolectaron en años anteriores, 2022 – actualidad. y, en base a estas se desarrollarán los elementos de la planificación minera (modelo económico y diseño de pit final) para cualquier mina a cielo abierto.

3.5. VARIABLES

3.5.1. Variables Independientes

- Algoritmo de Lerchs y Grossmann original
- Método de plantilla cono para ángulo variable
- Parámetros de optimización (geotécnicos, económicos y modelo de bloques)

3.5.2. Variables Dependientes

Tabla 12

Variables Dependientes

Variable	Descripción	Unidad	Indicador
Optimización del pit final	Ganancia o pérdida	USD	Volumen de producción Valor descontado del pit final
Implementación de algoritmo de LG con ángulo variable	Desarrollo e implementación de LG original y método de Plantilla como para ángulo variable	Años	Funcionamiento de software con implementación de LG con ángulo variable

3.6. POBLACIÓN Y MUESTRA

3.6.1. Población

La población está conformada por el modelo de bloques de corto plazo actualizado a abril del 2023, con sus respectivos parámetros: topografía, densidad, mineralización, tipo de roca, clasificación, valor económico, etc., que tiene el proyecto en cuestión.

3.6.2. Muestra

La muestra se divide en dos grandes grupos: parámetros económicos y parámetros técnicos.

Parámetros económicos:

- Precio del cobre, oro, plata y molibdeno
- Tasa de descuento
- Costo de producción y comercialización
- Parámetros técnicos: Costo incremental, bancos por año, etc.
- Ritmo de producción



- Ley de corte: BreakEven e interna.

Parámetros técnicos:

- Ángulos de talud por sector (roseta geotécnica)

CAPÍTULO IV

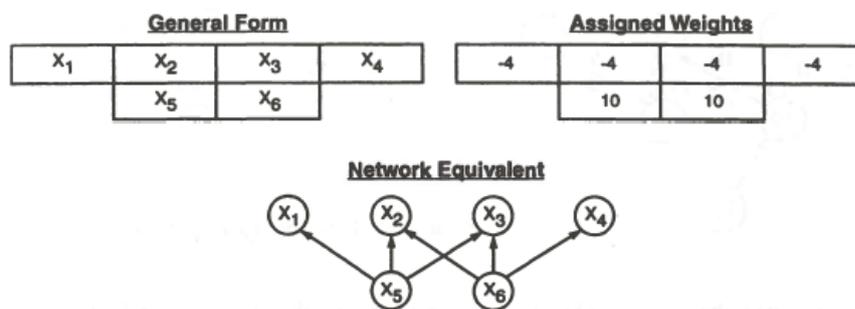
RESULTADOS Y DISCUSIÓN

4.1. IMPLEMENTACIÓN DEL ALGORITMO ORIGINAL DE LG EN PYTHON

A continuación, se muestra la implementación del algoritmo LG en Python, el cual se describe línea a línea con un ejemplo representativo, Figura 41, el cual no pudo ser resuelto por el algoritmo de Cono Flotante y Korobov.

Figura 41

Ejemplo base para implementación de algoritmo de LG

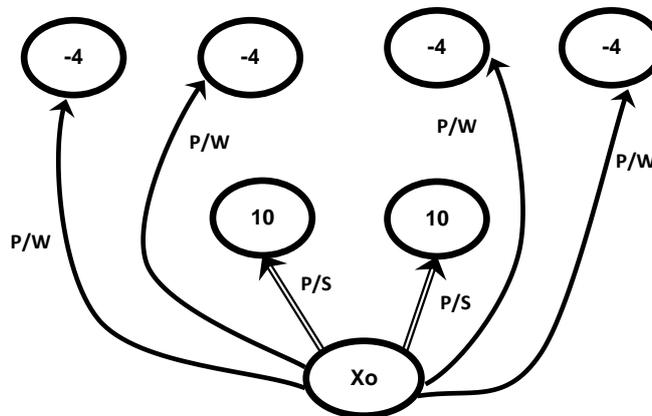


Fuente. Hustrulid, W., and Kuchta, M. 2006, “Open Pit Mine Planning and Design”, 2nd Edition., p. 447.

Paso 1: Se crea el grafo con 7 nodos (incluido el nodo raíz) y 6 arcos, los cuales son conectados del nodo raíz a los 6 nodos del grafo. Luego se etiquetan los arcos con Plus (p) si sale del nodo raíz o Minus(m) si entra hacia el nodo raíz. Además, se colocan los pesos correspondientes, como se muestra en la Figura 42.

Figura 42

Ejemplo base LG: Grafo normalizado

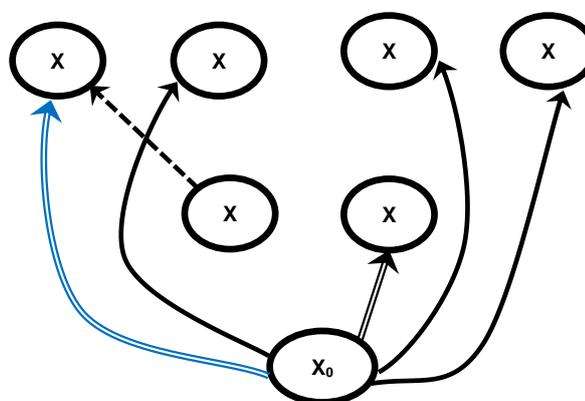


Paso 2: Se crean dos conjuntos $Y^\circ = [(x_0, x_5), (x_0, x_6)]$ y $X-Y^\circ = [(x_0, x_1), (x_0, x_2), (x_0, x_3), (x_0, x_4)]$. El primer conjunto tiene valores positivos y el segundo valor negativos, respectivamente. El conjunto Y° es la cerradura máxima o maximum closure el cual es 20.

Paso 3: Luego, se busca todas las posibles conexiones entre los dos conjuntos Y° y $X-Y^\circ$ respetando las siguientes restricciones de ángulo de talud $R = [(x_5, x_1), (x_5, x_2), (x_5, x_3), (x_6, x_2), (x_6, x_3), (x_6, x_4)]$. Seleccionamos una de las restricciones (x_5, x_1) y borramos el arco (x_0, x_5) y creamos el arco (x_5, x_1) , Figura 43.

Figura 43

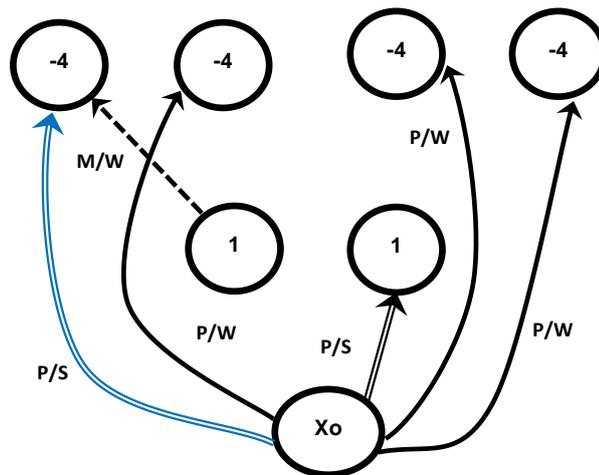
Ejemplo base: Selección y creación de arco (x_5, x_1)



Paso 4: Luego, normalizamos el árbol colocando las etiquetas a los arcos, como se muestra en la figura. Si el arco es “Positive” y “Strong”, entonces se agrega al conjunto Y° , actualizando $Y^\circ = [(x_0, x_1), (x_5, x_1), (x_0, x_6)]$, $X-Y^\circ = [(x_0, x_2), (x_0, x_3), (x_0, x_4)]$ y $R = [(x_5, x_2), (x_5, x_3), (x_6, x_2), (x_6, x_3), (x_6, x_4)]$

Figura 44

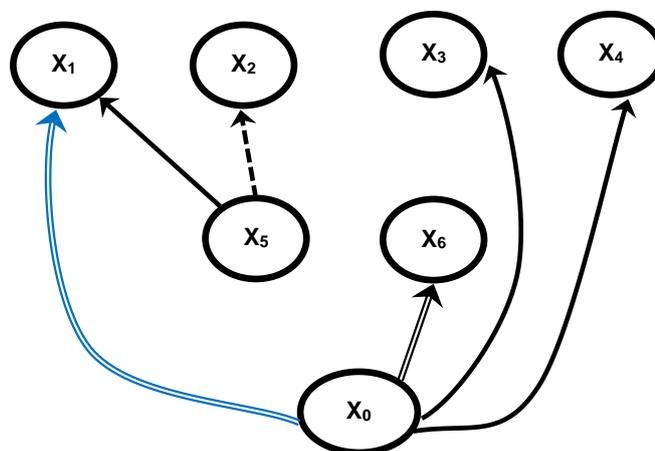
Ejemplo base LG: Etiquetado de los arcos



Paso 5: Ahora se retorna al Paso 3 para buscar conexiones adicionales entre Y° y $X-Y^\circ$. Hay 5 posibles conexiones, $R = [(x_5, x_2), (x_5, x_3), (x_6, x_2), (x_6, x_3), (x_6, x_4)]$. Se escoge el arco (x_5, x_2) , el cual será agregado al árbol y el arco (x_0, x_2) se borrará, Figura 45.

Figura 45

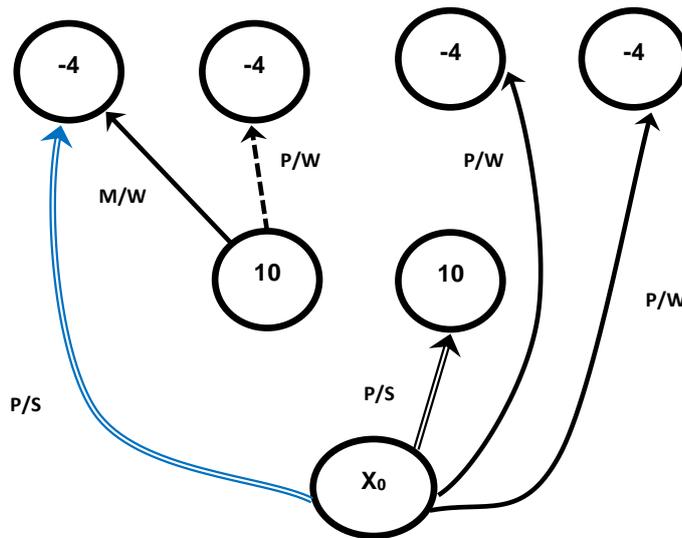
Ejemplo base LG: Adición de arco (x_5, x_2) y eliminación de (x_0, x_2)



Paso 6: El nuevo árbol es normalizado. El resultado se muestra en la figura 1. Actualizando $Y^{\circ} = [(x_0, x_1), (x_5, x_1), (x_5, x_2), (x_0, x_6)]$, $X - Y^{\circ} = [(x_0, x_3), (x_0, x_4)]$ y $R = [(x_5, x_3), (x_6, x_2), (x_6, x_3), (x_6, x_4)]$. El conjunto Y° es la cerradura máxima o el cual es 12.

Figura 46

Ejemplo base LG: Etiquetado de los arcos

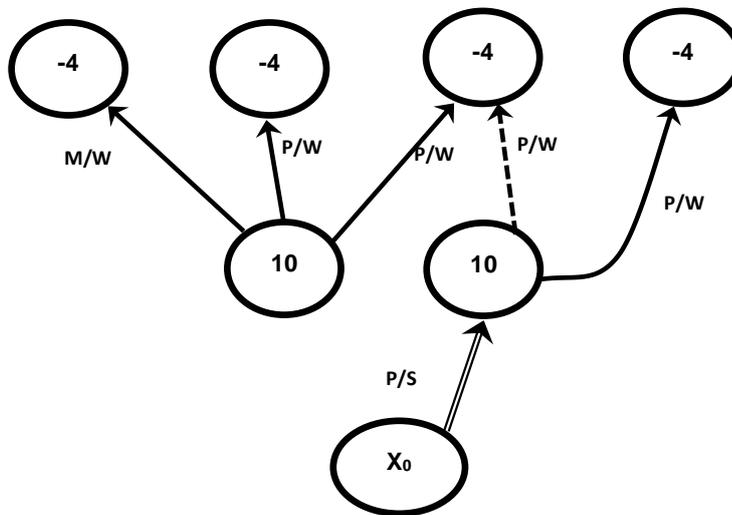


Paso 7: Volviendo al paso 5, existen 4 posibles conexiones restantes $R = [(x_5, x_3), (x_6, x_2), (x_6, x_3), (x_6, x_4)]$. Se agregará el arco (x_5, x_3) y se borrará el arco (x_0, x_3) . El árbol normalizado es mostrado en la figura 1. Como se puede ver, el arco (x_0, x_1) ahora se convirtió en “P/W”.

Actualizando $Y^{\circ} = [(x_0, x_6)]$, $X - Y^{\circ} = [(x_0, x_1), (x_5, x_1), (x_5, x_2), (x_5, x_3), (x_0, x_4)]$ y $R = [(x_6, x_2), (x_6, x_3), (x_6, x_4)]$

Figura 48

Ejemplo base LG: Adición de arco (x_6, x_3) y eliminación de (x_0, x_1)



El proceso de normalizar el árbol consiste en conectar el árbol al nodo raíz siguiendo los valores de la tabla 1, Plus(P): Si sale del nodo raíz, Minus(M): Si entra hacia el nodo raíz, luego el valor Positivo o Negativo en función de los pesos de los nodos, ver Tabla 1. Por ejemplo, el arco (x_0, x_1) es Plus porque sale del nodo raíz y tiene un valor Negativo por que el nodo tiene un valor de -4 , Por lo tanto, este arco es Plus/Weak. Cabe mencionar que no ingresa al conjunto de valores positivos Y.

Ahora se resolverá el ejemplo base, Figura 41, con Python para lo cual se importa la librería “grafo” y “datetime”. La primera fue desarrollada por Yovani Achata Flores y es una clase la cual se inicializa las variables y funciones necesarias para la implementación de LG.

```
from Grafo import Grafo
import datetime
```



Se inicializa la clase Grafo con 7 nodos el cual incluye el nodo raíz.

```
myGrafo = Grafo(7)
```



Se agregan las dependencias o arcos en el formato nodo inicio, nodo final, y peso del nodo (proceso de normalización), como se muestra a continuación:



```
myGrafo.AddArc(0, 1, -4)
myGrafo.AddArc(0, 2, -4)
myGrafo.AddArc(0, 3, -4)
myGrafo.AddArc(0, 4, -4)
myGrafo.AddArc(0, 5, 10)
myGrafo.AddArc(0, 6, 10)
```

python

Se calculan las restricciones de los bloques con valor positivo a los bloques con valor negativo.

```
myGrafo.CalculeConstraints()
myGrafo.AddConstraint(5, 1)
myGrafo.AddConstraint(5, 2)
myGrafo.AddConstraint(5, 3)
myGrafo.AddConstraint(6, 2)
myGrafo.AddConstraint(6, 3)
myGrafo.AddConstraint(6, 4)
```

python

Luego se recorre los nodos del grafo para determinar la cerradura máxima.

```
lconos = []
cadroot = 0
upl = []
print ('inicio determinar final closure %s ' %
str(datetime.datetime.now()))

while True:
```

python

Primero, se revisa todos los arcos “strong”; para que un arco sea “Strong” tiene que ser positivo.

```
for elem in myGrafo.d2dpos:
    stats = myGrafo.CheckNodeStatus(elem)
    if stats[0] == 'S':
        lconos.append(elem)
```

python

En la lista “lconos” se almacenan los nodos “Strong” 5 y 6.

```
for elems in lconos:
    chkns = ['S', cadroot]
```

python

Se recorre cada todas las dependencias desde los nodos positivos a los nodos negativos, PASO 1.

```
for selem in myGrafo.d2dpos[elems]:  
    cade = str(selem[0])  
    chkns = myGrafo.CheckNodeStatus(elems)  
    chknw = myGrafo.CheckNodeStatus(cade)
```



Hasta aquí es PASO 1, es decir si no hay más arcos “Strong” y “Weak” entonces ir al PASO 4.

```
if chknw[0] == 'W' and chkns[0] == 'S':  
    if chknw[1] not in upl and chkns[1]  
not in upl:  
        upl.append(chkns[1])  
        nconflic =  
myGrafo.checkweaknode(str(elems), str(cade))
```



PASO 4, no hay más posibles conexiones y todos los arcos “Strong” están conectados al nodo raíz.

```
if len(myGrafo.d2dpos) == len(lconos):  
    break  
print ('fin determinar final closure %s ' %  
str(datetime.datetime.now()))  
print("Maximum Closure:  
{0}".format(myGrafo.UltimatePitLimit()))
```



Finalmente, se obtiene la cerradura máxima el cual es el siguiente arreglo de nodos, Máximum Closure: ['1', '2', '3', '4', '5', '6'] con un valor de +4.

4.2. IMPLEMENTACIÓN DE ANGULO VARIABLE

Casi todos los algoritmos usan un modelo de bloques del depósito, por ejemplo, un arreglo tridimensional de bloques de tamaño común que cubren en su totalidad el mineral y suficientemente los lados del desmonte para permitir acceso a los bloques de mineral más profundos. Uno de estos es el algoritmo de LG basado en la teoría de grafos, es el único método que asegura siempre obtener el pit óptimo. Sin embargo, el algoritmo original asume ángulos fijos el cual están en función de la dimensión de los bloques.

Aquí se presentará un método para incorporar ángulos variables en el algoritmo de LG. Se asume que la zona de mineral y desmonte está dividida en regiones o sectores



de dominio en el cual la característica de la roca son las mismas y las regiones son norte, sur, este y oeste. Los ángulos pueden variar a través del depósito para seguir las características de la roca y son independientes de las dimensiones de los bloques.

Muchos factores rigen el tamaño forma del pit. El ángulo de pit es uno de los factores clave que gobierna la cantidad de desmonte a ser minado para tener acceso al mineral y este no está restringido a un ángulo constante. Pequeñas variaciones en el ángulo pueden cambiar la cantidad de desmonte y tener un efecto significativo en el grado de selectividad en las operaciones mineras. Es importante cambiar los ángulos de talud ya sea por razones geotécnicas o seguir diferentes estructuras y tipos de roca en el depósito, aparte de la necesidad de mantener la cantidad de desmonte lo más pequeña posible.

Muchos intentos se han hecho para sobrepasar la dificultad de incorporar ángulos variables en el algoritmo de LG, pero ninguno provee una solución adecuada para casos en los cuales los ángulos variables son controlados por estructuras complejas y geología. Alford y Whittle reportaron la incorporación de ángulos variables en el algoritmo, pero sin detalles.

4.2.1. Restricciones de acceso y minado

Para un depósito representado por su valor económico, los ángulos de pit son dados en función de bloques que deben ser minados para dar acceso a cada bloque dentro del modelo de bloques. En el algoritmo de LG los arcos generados son los que imponen estas restricciones. Ellos indican cuales bloques deben ser removidos para que un bloque en particular pueda ser minado, por ejemplo, en la Figura 49 en la cual cada bloque tiene tres inmediatos sucesores. Los bloques sucesores inmediatos de cualquier bloque específico debe ser removido antes que

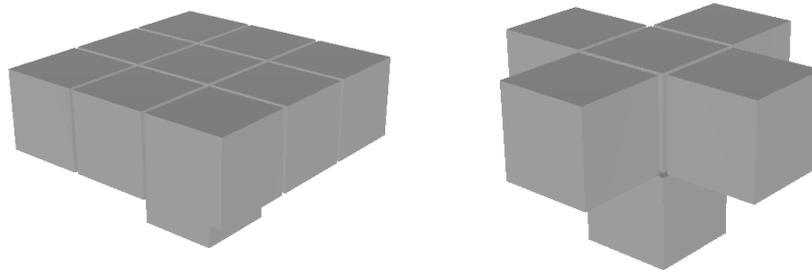
el bloque pueda ser minado. Existen dos patrones para especificar restricciones de minado para los diferentes modelos de bloques, estos se clasifican en:

4.2.2. Métodos no basados en conos:

Este método involucra el uso de un patrón o un conjunto de bloques para definir los ángulos de pit, una configuración 1 : 5, 1 : 9, o una combinación de estos es conocido como la configuración 1 : 5 : 9, Figura 49.

Figura 49

Configuración de método no basado en conos



Nota. Configuración no basada en conos a: 1:5, 5 bloques deben ser removidos para minar el bloque inferior; b: 1:9 nueve bloques deben ser removidos para minar el bloque inferior.

En este método si se proyecta la configuración por varios niveles se obtiene un ángulo de pared indeseable, por ejemplo, con la configuración 1 : 5 en un modelo de bloques cubico el ángulo de pit promedio estaría en el rango de 45 - 55° y con la configuración 1 : 9 en un modelo de bloques cubico el ángulo de pit promedio estaría en el rango de 35 - 45°. Además, una aproximación a 45° se obtiene usando la configuración combinada 1 : 5 para el primer nivel y 1 : 9 para el segundo nivel.

El uso de este método es la dificultad de establecer límites de pit óptimos con ángulos de pit variable. En este método los ángulos de pit son definidos por las dimensiones de los bloques, por ejemplo, si se usa una configuración 1 : 5 : 9

en un modelo rectangular con dimensiones 10 x 10 x 5m se obtiene un ángulo de pit de 25°. Por lo tanto, si se usa este método, diferentes ángulos de pit necesitaran diferentes dimensiones de bloques. Las leyes de los bloques con estimadas con diferentes configuraciones tendrían errores de estimación, por ende, crearía dificultad en la cuantificación del nivel de confianza de los niveles asociados con el valor del pit final (el pit optimo es usado para definir las reservas minables conciertos niveles de confianza). Además, diferentes partes del yacimiento necesitarían diferentes dimensiones de bloques lo cual no es óptimo; ya que solo se acostumbra trabajar con dimensiones constantes en todo el yacimiento.

4.2.3. Métodos basados en conos

En este método un cono es usado en diferentes formas para definir el ángulo de pit. DOWD AND ONUR usaron la idea de conos plantillas para obtener el pit optimo con ángulo variable. Este método involucra la construcción de un cono desde un bloque en cierto nivel hacia la superficie uniendo los anillos de los bloques. Si el centroide del bloque está dentro del cono, se asume que debe ser minado antes que el bloque base.

4.2.4. Algoritmo De Lerchs Y Grossmann Con Angulo Variables

Para implementar ángulos variables en el algoritmo de LG se asume que la zona de mineral y desmonte haya sido dividida en regiones o dominios en función de información geotécnica. También, se supone que en cada región o dominio las características de la roca son las mismas y puede ser caracterizada por un conjunto de ángulos y aproximada por un polígono.

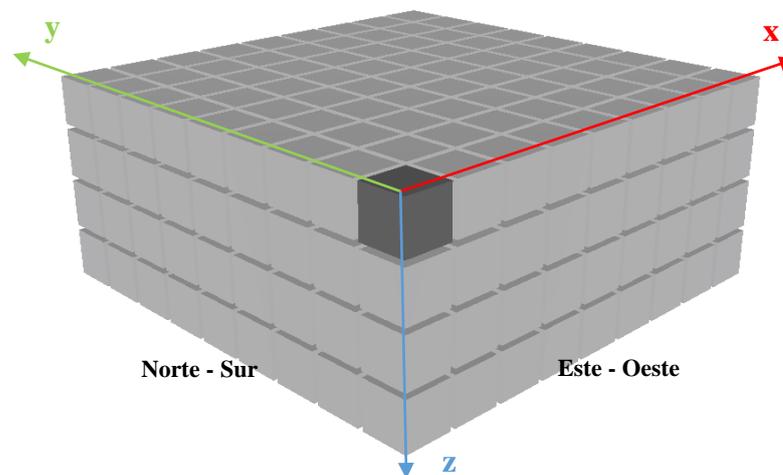
Dos tipos de sistemas de coordenadas, como se muestra en la figura 1, son usados. El primero es el sistema X, Y, Z en el cual el eje X va de Oeste a Este, el eje Y va desde Norte a Sur, y el eje Z es el vertical. El origen del sistema está

localizado en dirección Sur-Oeste del nivel más alto del modelo, el bloque sombreado en la

Figura 50. El segundo sistema es un sistema de índices i, j, k ; las coordenadas i, j, k se incrementan en la dirección de X, Y, Z . Además, se usan los siguientes parámetros para definir el modelo de bloques: x_{dim} , dimensión del bloque en la dirección de X (Oeste-Este); y_{dim} , dimensión del bloque en la dirección de Y (Sur-Norte); z_{dim} , dimensión del bloque en la dirección de Z (Vertical); num_x , número de bloques en la dirección X (Oeste-Este); num_y , número de bloques en la dirección de Y (Sur-Norte); y num_z , número de bloques en la dirección de Z o número de niveles.

Figura 50

Modelo de bloques con sistema de coordenadas

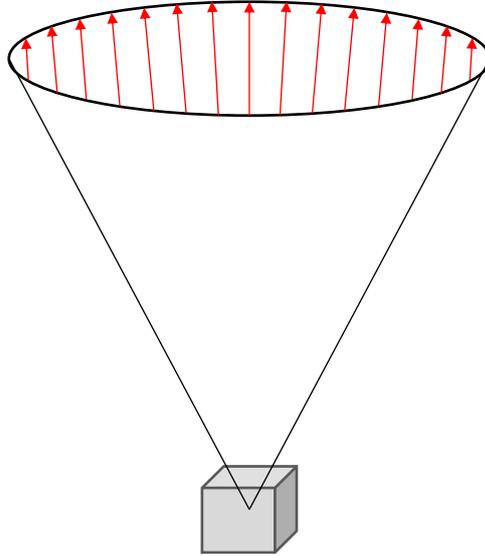


4.2.5. Angulo variable

Los ángulos pueden ser aproximados construyendo un cono que represente un volumen de extracción. Esto puede ser hecho creando anillos o envolturas desde el centroide del bloque base y extendiéndolo hacia la superficie Figura 51.

Figura 51

Cono de extracción con envolventes



Si los ángulos en las 4 direcciones principales no son los mismos, la parte superior del cono en cada nivel consistirá de 4 cuadrantes de elipses diferentes. Si los ángulos son los mismos, entonces el área superior en cada nivel serán círculos, Figura 52, Figura 53 y Figura 54 muestran el cono de extracción y los bloques en el primer nivel y en las 2 secciones correspondientes.

Figura 52

Área superior del cono en el nivel superior

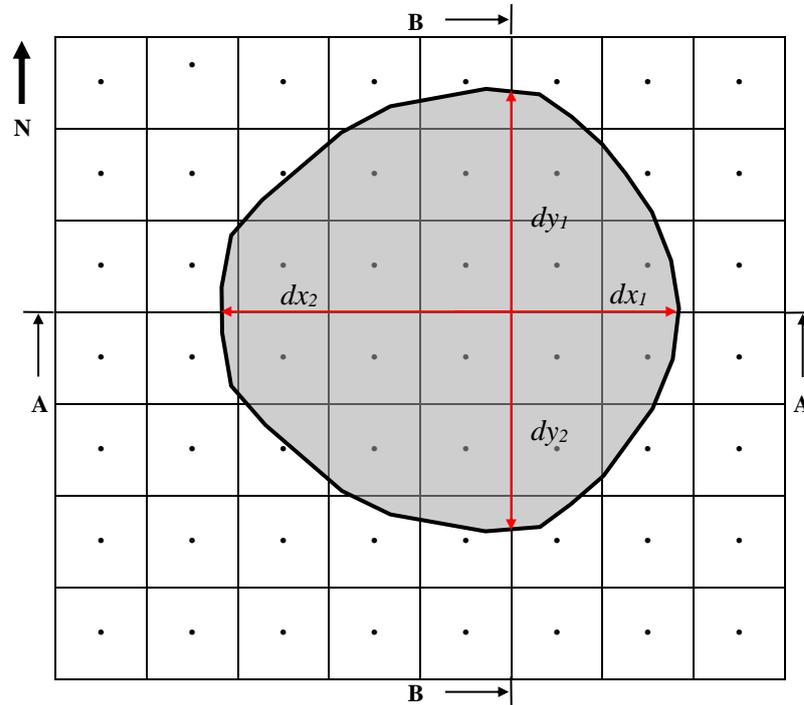


Figura 53

Sección A-A Norte-Sur

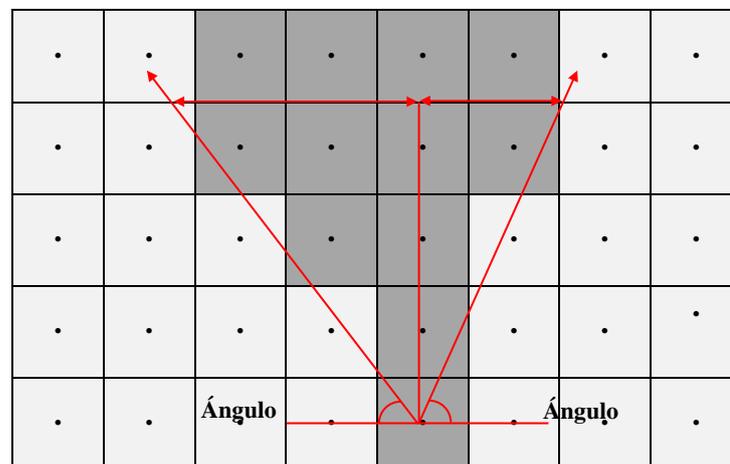
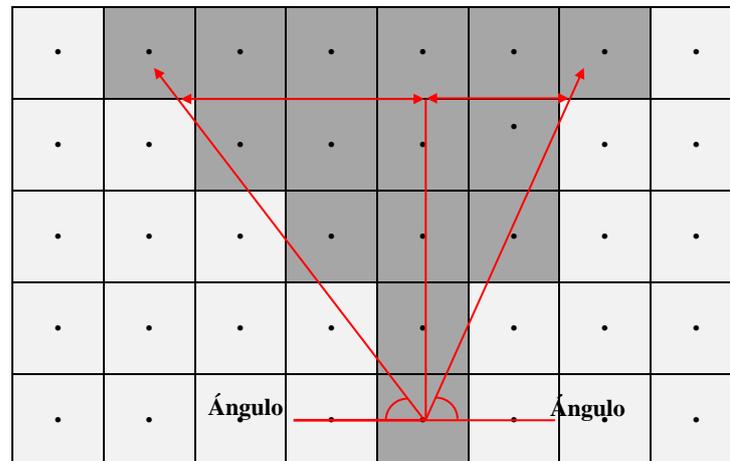


Figura 54

Sección A-A Este-Oeste



En cada nivel los valores de los dos ejes semi mayores y los dos ejes semi menores dependen en los 4 ángulos principales y distancia vertical desde el bloque base. Estos parámetros se pueden hallar mediante funciones trigonométricas. El número de bloques en las direcciones principales en cualquier nivel encima del bloque base pueden ser calculadas dividiendo estos parámetros por la dimensión correspondiente. Considere un bloque $X_{i,j,k}$ en el nivel k : los parámetros y el número de bloques en las direcciones principales, así como se muestra en la Figura 52, Figura 53 y Figura 54 puede ser calculada con las siguientes ecuaciones.

$$dx_1 = \frac{(k - t) * z \text{ dim}}{\tan(\text{Angulo Oeste})}$$

$$dy_1 = \frac{(k - t) * z \text{ dim}}{\tan(\text{Angulo Sur})}$$

$$dx_2 = \frac{(k - t) * z \text{ dim}}{\tan(\text{Angulo Este})}$$

$$dy_2 = \frac{(k - t) * z \text{ dim}}{\tan(\text{Angulo Norte})}$$



$$m_1 = \frac{dx_1}{x \dim}$$

$$n_1 = \frac{dy_1}{y \dim}$$

$$m_2 = \frac{dx_2}{x \dim}$$

$$n_2 = \frac{dy_2}{y \dim}$$

Donde t es el nivel encima del bloque base y v varia entre 1 a $k - 1$; m_1 es el número de bloques desde el bloque base hacia el este; n_1 es el número de bloques desde el bloque base hacia el norte; m_2 es el número de bloques desde el bloque base hacia el Oeste; n_2 es el número de bloques desde el bloque base hacia el Sur.

Cuando el número de bloques dentro de área superior del cono en cualquier nivel son calculados en las cuatro direcciones principales todos los bloques $X_{m,n,k-t}$ donde $m = i - m_2, i - m_1$ y $n = j - n_2, j - n_1$ deben ser examinados para determinar si ellos están dentro del cono o volumen de extracción. Esto puede realizarse con la ecuación de la elipse:

$$a = xdim * (i - m)$$

$$b = ydim * (j - n)$$

Si “ m ” es igual o mayor que “ i ” y “ n ” es igual o mayor que “ j ”, entonces:

$$Valor = \frac{a^2}{(dx_1)^2} + \frac{b^2}{(dy_1)^2}$$

Si “ m ” es igual o mayor que “ i ” y “ n ” es igual o menor que “ j ”, entonces:

$$Valor = \frac{a^2}{(dx_1)^2} + \frac{b^2}{(dy_2)^2}$$

Si “ m ” es igual o menor que “ i ” y “ n ” es igual o menor que “ j ”, entonces:

$$Valor = \frac{a^2}{(dx_2)^2} + \frac{b^2}{(dy_2)^2}$$

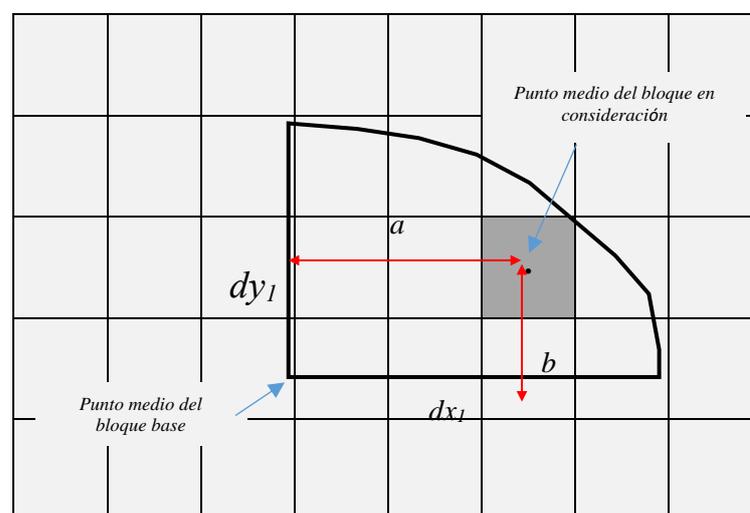
Si “m” es igual o menor que “i” y “n” es igual o mayor que “j”, entonces:

$$Valor = \frac{a^2}{(dx_2)^2} + \frac{b^2}{(dy_1)^2}$$

Donde “a” y “b” son las distancias horizontales desde el centroide del bloque bajo consideración a el bloque base medido las direcciones Oeste-Este y Sur-Norte, respectivamente, así como se muestra en la figura 7. Si el “Valor” es menor o igual que 1, se asume que el bloque está dentro del cono de extracción y debe ser removido antes que el bloque base. De lo contrario, se asume que el bloque esta fuera con cono. Bloques que caen dentro del cono de extracción son enviado al algoritmo de Lerchs y Grossmann. La implementación en Python se escribió de tal forma que los conos de extracción solo son establecidos para los bloques de mineral o valor descontado positivo. Esto previene incrementos innecesarios en tiempo de cálculo y bloques de desmonte ser considerados varias veces.

Figura 55

Vista en Planta de un bloque en el nivel superior





4.3. SOFTWARE

Todos los métodos descritos en los capítulos anteriores, el algoritmo para el diseño óptimo de tajo abierto con ángulos variables, el procedimiento para crear un modelo de bloque de valor descontado del yacimiento y el método para diseñar cuatro ángulos de pendiente principales, se han codificado como una librería de Python en donde se ha programado las variables, funciones y clases necesarias para la implementación del algoritmo original de LG. De manera similar se hizo con la implementación de ángulo variable. El software se escribió completamente en código Python utilizando el Software “Visual Studio Code” como entorno de desarrollo; tanto para escribir las funciones, clases y hacer el debugging correspondiente. El software ha sido probado con los ejemplos mencionados en el capítulo II y un caso real del tajo Pampacancha. Este capítulo describe la estructura y el funcionamiento del software. También hay una breve descripción de los programas para la visualización numérica y gráfica de los datos de entrada y los resultados de la optimización.

4.3.1. Requerimientos del sistema

El software es un programa para Windows el cual requiere como mínimo la siguiente configuración para su implementación:

- Una computadora de escritorio o portátil con un procesador Core i7.
- Memoria RAM 16 GB
- Un disco duro con suficiente espacio
- Sistema operativo Windows 10 o 11.

4.3.2. Estructura del software

El software está escrito íntegramente en Python utilizando algunas librerías en Python como tkinter, matplotlib, math, os, time, entre otros. Se trata

de una aplicación para Windows de 64 bits y consta de los siguientes programas, cada uno de los cuales se describe brevemente a continuación:

4.3.3. Clases del software:

4.3.3.1. Clase Importa_Data:

Esta clase importa el modelo de bloques desde un archivo separado por comas en formato o extensión “.csv”. Cada fila representa las coordenadas de un bloque X, Y, Z. Además, se tienen las variables “Sector” y “Valor Descontando (DPRF)”, el primero representa el ángulo variable para el bloque en mención y el segundo el valor con el cual el algoritmo de Lerchs and Grossman trabajara.

Primero, importar variables del modelo de bloques

```
For linea in modeloBloques:  
    cordX = linea[0]  
    cordY = linea[1]  
    cordZ = linea[2]  
    topog = linea[3]  
    sector = linea[4]  
    valDesc = linea[5]
```

End For

Segundo, importa configuración o parámetros del modelo de bloques:

```
For linea in parametros:  
    xMin = linea[0]  
    yMin = linea[1]  
    ZMin = linea[2]  
    xMax = linea[3]  
    yMax = linea[4]  
    zMax = linea[5]  
    xDim = linea[6]  
    yDim = linea[7]  
    zDim = linea[8]
```

End For

4.3.3.2. Clase AlgoritmoLG:

```
# Se inicializa el grafo conectando los nodos a el nodo raíz (Árbol normalizado)
For each node in nodes:
    arcosTabla [] = node (x0, xn)
End For
# Se recorre el grafo de arriba hacia abajo para determinar la cerradura máxima por banco

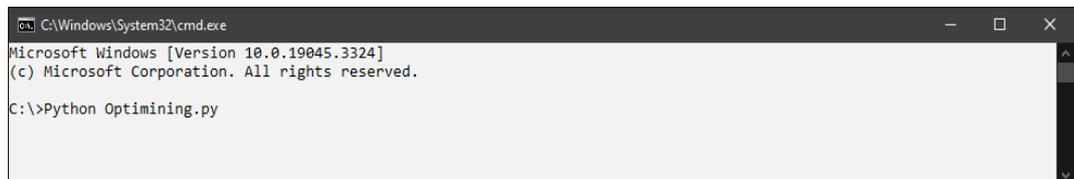
bancoActual = nivelSuperior
cerraduraFinal = false
while not cerraduraFinal:
    cerraduraBanco = false
    while not cerraduraBanco:
        strongNodos = arcosTabla [ x0, xn]
        For each strongNodos:
            weakNodes = conoTemplate [ x0, ..., xn]
        End For
        if weakNodes = []:
            cerraduraBanco = true
        else:
            funciónTrasnformarArbol () #Paso 2 LG
            funciónNormalizarArbol () #Paso 3 LG
            nodosMinados ()
        if bancoActual = bancoInferior:
            cerraduraFinal = true
        else:
            bancoActual = bancoActual - 1
```

4.3.3.3. Clase Graphical Unit Interface (GUI)

Este software se inicia con la ejecución del comando “Python Optimining.py” desde la ruta en la que el archivo “*.py” se encuentra o como se muestra en la siguiente imagen.

Figura 56

Comando para ejecución de software en CMD



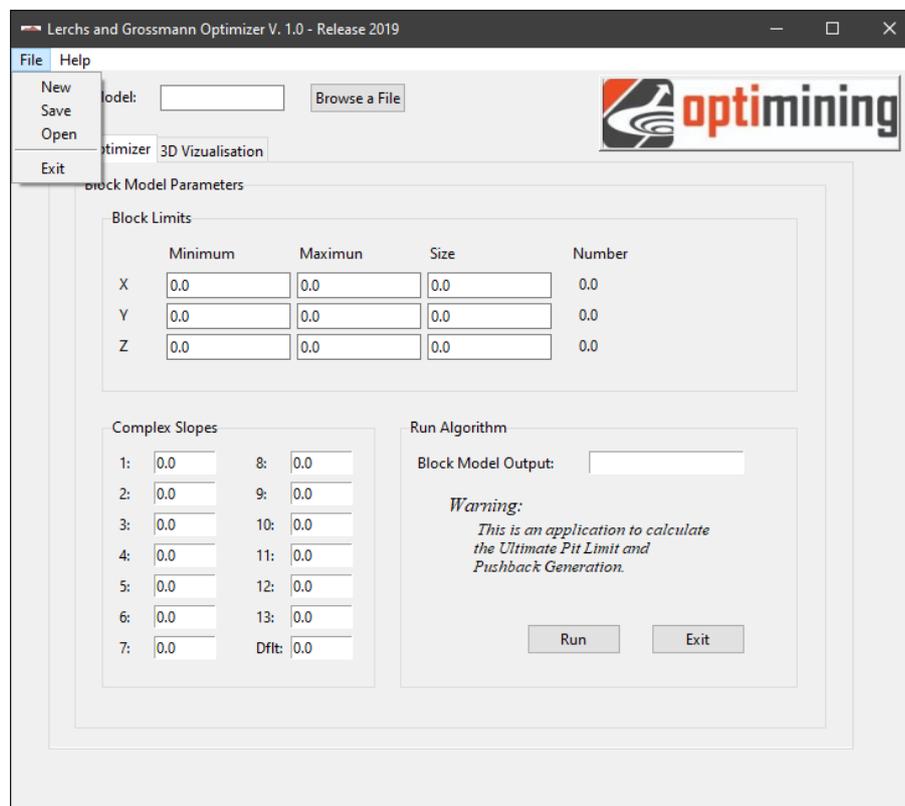
El software se compone de los siguientes componentes:

- En la ventana general se tiene el título, menú, iconos para minimizar, maximizar y cerrar el programa.

- La barra de herramientas contine varias funciones los cuales realizan una función en específico, por ejemplo, se tiene la función “New” el cual borra todos los parámetros ingresados en la ventana de inicio o “UPL Optimizer”; la función “Save” guarda todos los parámetros ingresados en la ventana “UPL Optimizer” en un archivo de salida con extensión “*.csv”; y por último la función “Open” el cual carga los parámetros desde el archivo generado previamente.

Figura 57

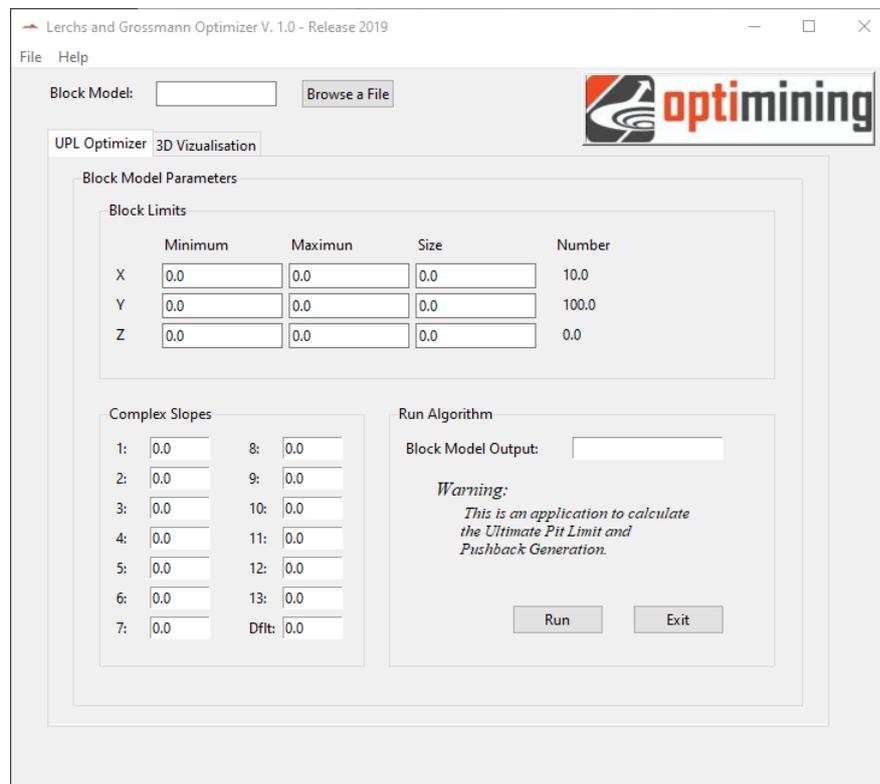
GUI: Barra de herramientas



- **Ventana Principal:** Esta ventana contiene al botón “Browse a File” para cargar el modelo de bloques en formato “*.csv”. También, dentro de esta ventana principal se tiene 2 sub ventanas: “UPL Optimizer” y “3D Vizualisation”.

Figura 58

GUI: Ventana principal



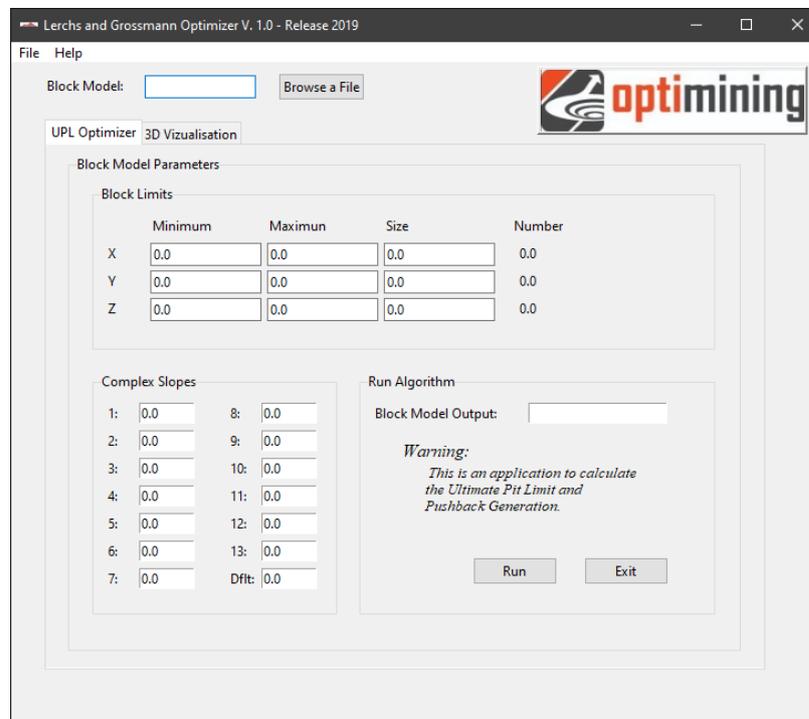
- Ventana UPL Optimizer:
- Parámetros del Modelo de Bloques: La primera sub ventana es en donde se ingresan todos los parámetros del modelo de bloques como limites del modelo, ángulos complejos y el nombre del archivo de salida.
- Es necesario que se ingrese los limites mínimos y máximos en los tres ejes coordenados X, Y, Z. Además, es necesario especificar la dimensión de bloque. El número de bloques será calculado automática e interactivamente por el software. Estos calculo también pueden ser hechos en cualquier hoja de cálculo.
- Ángulos Complejos: En el cuadro de ángulos complejos es necesario colocar el ángulo por defecto “Default” en el caso que no se cuenten con sectores

definidos. Por otro lado, si se cuenta con sectores definidos entonces llenar el ángulo correspondiente asignado al valor que se tiene en el modelo de bloques.

- **Ejecutar Algoritmo:** En el cuadro de correr o ejecutar algoritmo se necesita especificar el nombre del archivo de salida en formato “*.csv”. Este archivo es el resultado final de la optimización por el método de Lerchs y Grossmann para obtener el Pit Final.

Figura 59

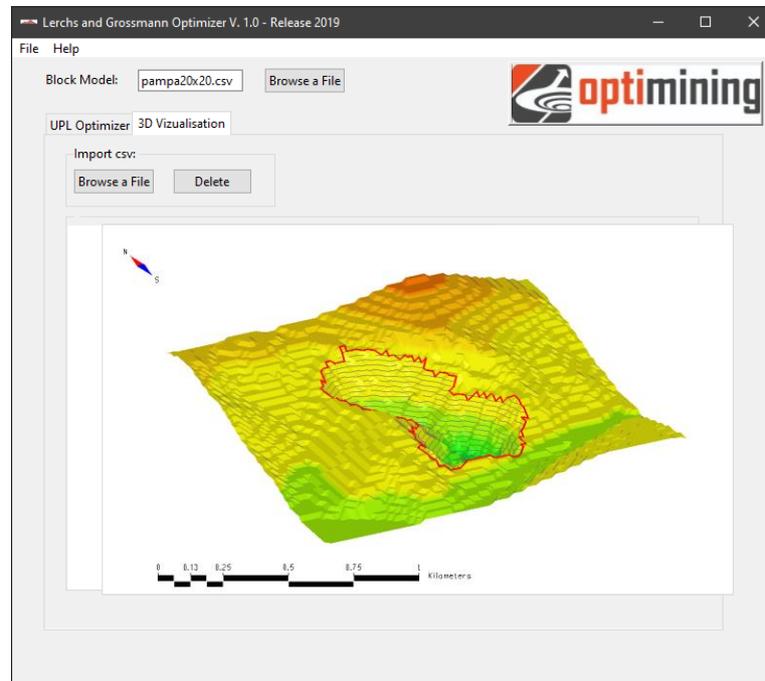
GUI: Ventana UPL Optimizer



- **Ventana 3D Visualisation:** La segunda ventana es donde se puede visualizar los resultados de la optimización. El cuadro “Import csv” permite importar el resultado de la ventana “UPL Optimizer” así como también eliminar del visualizador el resultado, como se muestra en la siguiente imagen.

Figura 60

GUI: Ventana 3D Visualisation



4.3.4. Funciones del algoritmo original de Lerchs y Grossmann

4.3.4.1. Función HallarCamino:

Esta función busca el camino en el grafo pasándole un nodo inicio y final. Si encuentra algún camino entonces lo agrega a la lista camino y por último devuelve dicha lista.

Función para encontrar camino en el grafo

```
Def HallarCamino(nodoInicio, nodoFinal, camino = []):  
    camino = camino + [nodoInicio]  
    If nodoInicio == nodoFinal:  
        Return camino  
    Else:  
        For nodo in grafo[nodoInicio]:  
            If nodo not in camino:  
                caminoNuevo = HallarCamino(nodo, nodoFinal,  
                    camino)  
                Return caminoNuevo  
    Return none  
End For
```

4.3.4.2. Función RevisarEstadoNodo:

Esta función revisa el estado del arco ya sea “Strong” o “Weak”

```
# Función para revisar el estado del nodo
Def RevisarEstadoNodo(nodo):
    caminoB = HallarCamino(nodo, nodoRaiz)
    If pesoRaiz > 0:
        estado = ['S', xm]
    Else:
        estado = ['W', xm]
    Return estado
```

4.3.4.3. Función AgregarPrecedencia:

Esta función actualiza el grafo con la precedencia. Además, recibe como input dos variables nodoK que pertenece a los valores positivos o conjunto Y y nodoL que pertenece a los valores negativos o conjunto X-Y’

```
Def AgregarPrecedencia(nodoK, nodoL):
    If nodoL in Grafo:
        antecedente = Grafo[nodoL]
    Else:
        antecedente = []
    Grafo[nodoK] = antecedente
```

4.3.4.4. Función Normalizar:

Esta función Normaliza el Grafo. El proceso de normalizar consiste en: Si existe un arco “Strong” no conectado al nodo raíz, este debe ser conectado al nodo raíz.

```
Def Normalizar(ListaAnormalizar):
    If plusOrMinus == 'P' y valor > 0:
        Delete arcos[Strong, x ]
    End If
```

4.3.4.5. Función EliminarArco:

Se elimina el arco conectado al nodo raíz en el diccionario “grafo” y “arcos”

```
Def EliminarArco ():
```

```
Delete Grafo[nodo][nodoRaiz]
```

4.3.5. Requerimientos de Entrada (Input):

Los datos de entrada necesarios para diseñar el tajo abierto óptimo son el modelo de bloques económico y las pendientes del tajo. El único dato de entrada predeterminado es el modelo de bloques del yacimiento; el resto de la información puede introducirse durante la ejecución del programa. El modelo de ley de bloques del yacimiento debe ser representado en un archivo de texto. Este archivo debe contener un registro por cada bloque y cada registro debe contener:

- Coordenada Este (X): Centroides del bloque en la dirección X
- Coordenada Norte (Y): Centroides del bloque en la dirección Y
- Coordenada Vertical (Z): Centroides del bloque en la dirección Z
- Valor Descontado del Bloque
- Categoría de Sector Geotécnico
- Topografía: Porcentaje del bloque que está por debajo de la topografía.

Los registros del archivo tienen que estar en el orden descrito anteriormente y la información puede registrarse en formato “*.csv”. Los bloques con topografía > 0 describen la topografía de la superficie. Los valores que faltan generalmente denotan bloques para los que hay muy pocos datos como para proporcionar una estimación.



La información necesaria para definir el ángulo de talud depende del número de regiones. Si se especifica una región, se debe completar el input “Default”. Se pueden especificar hasta un máximo de 13 regiones diferentes y no hay limitación en el número de coordenadas que definen cada región.

4.3.6. Resultados (Output):

Existen varias formas de visualizar los resultados de la optimización. La forma usada para la presentación de resultados en el “Software Optimining” es la representación de las coordenadas en un sistema de coordenadas UTM. Aquí se puede visualizar la superficie topográfica en intersección con el límite del pit final optimo.

4.4. OPTIMIZACIÓN DEL PIT FINAL CON LA LIBRERÍA DE LG

4.4.1. Aspectos generales de la mina Constanca – Hudbay

4.4.1.1. Ubicación y acceso

La mina Constanca se ubica en el distrito de Chilloroya, provincia de Chumbivilcas, departamento de Cusco. Se encuentra 634 km al sureste de la ciudad de Lima y 112 km al Sur de la ciudad de Cusco (distancia en línea recta), a una altitud que varía entre los 4 000 y 4 400 m.s.n.m. La ubicación geográfica está dada por las siguientes coordenadas UTM (WGS 84 Zona 19S)

- N: 8 393 167 a 8 400 860
- E: 197 036 a 203 806

Figura 61

Ubicación Mina Constancia



4.4.1.2. Descripción del proyecto

Hudbay llegó a Perú con la adquisición del proyecto de cobre Constancia en 2011. Se completó la construcción de Constancia en 2014 y la mina pasó de la primera producción a la producción total en solo cinco meses. Las operaciones de Constancia también incluyen el tajo satélite cercano Pampacancha, un depósito de cobre y oro de alta ley donde las actividades mineras comenzaron en 2021. Se adquirió una serie de propiedades cerca de Constancia, agregando un gran bloque contiguo de prospectos de exploración dentro de la distancia de camiones de la planta de procesamiento de Constancia.

La unidad minera Constancia tiene dos tajos de explotación a la fecha, los cuales son Tajo Constancia y Tajo Pampacancha. Los cuales son yacimiento del tipo pórfido Skarn. La mena principal es la calcopirita y su producto principal es el Cobre.

En marzo del 2021 se emitió el informe titulado “Constancia-NI-43-101-Technical Report” en base a un total de 28 perforaciones de circulación inversa y 112 HQ perforaciones diamantinas. En este informe se indicó los cálculos de

recursos y reserva que se realizaron en base a los sondeos, muestreos en interior mina (labores ya realizadas) e informaciones geológicas y geoquímicas disponibles a esa fecha. Logrando cubicar 32'400,000 Toneladas de material probado, con una ley promedio de 0.59% Cu, también reporto material probable total de 7'500,000 toneladas con ley promedio de 0.62% Cu, como se muestra en la Tabla 13.

Tabla 13

Reservas probadas y probables de Tajo Pampacancha

Reservas Pampacancha	Toneladas	Ley Cu (%)	Ley Mo (g/t)	Ley Au (g/t)	Ley Ag (g/t)
Probado	32,400,000	0.59	178	0.368	4.48
Probable	7,500,000	0.62	173	0.325	5.75
Total, Probado + Probable	39,900,000	0.60	177	0.360	4.72

Fuente: Reporte Técnico NI 43-101

4.4.2. Mineralización del Tajo Pampacancha

El depósito de Pampacancha es un sistema Skarn relacionado con pórfido Cu-Mo-Au. El pórfido de diorita citado como fuente de mineralización de skarn intruye diorita de sótano no mineralizada del oligoceno. Esto, a su vez, está cortado por intrusiones de monzonita intramineral que proporcionan aumentos locales menores en Cu-Au y también reemplazan localmente la mineralización de Cu-Au de skarn que está más desarrollada en los márgenes superior e inferior del cuerpo de piedra caliza. El skarn de magnetita-calcopirita-pirita varía hasta el skarn marginal de granate y piroxeno menos mineralizado, localmente sobreimpreso por un skarn retrógrado portador de epidota.

La mineralización epitermal como el estilo Au + Cu de cuarzo-sulfuro de baja sulfuración explica las anomalías comunes de Au enriquecido con material



Supérgeno junto con otras características como la alteración hidrotermal y las vetas típicas de las ubicaciones cercanas a los pórfidos.

4.4.3. Modelo de bloques 3D

La unidad minera Constanca específicamente el tajo Pampacancha tiene un modelo de corto plazo de dimensiones siguientes: 10m (este-oeste) x 10 m (norte-sur) x 15m (vertical). El número de bloques en dirección este-oeste, norte-sur y vertical es de 168 ,210 y 60 respectivamente. Algunos bloques no se han estimado por falta de datos y sus coordenadas se omiten en el modelo de grado de bloques. Además, se han omitido algunos bloques en los residuos circundantes, pero esto no afecta a la determinación del pit óptimo. Otros datos de entrada son:

4.4.4. Valorización económica con Script en Python

En el siguiente código abierto en Python se realiza la valorización del modelo de bloques para obtener el valor del mineral y el desmonte. Para ello, se han utilizado algunas herramientas informáticas comúnmente empleadas en el análisis de datos.

4.4.5. Optimización del Pit Final

Al optimizar el pit final del tajo Pampacancha se obtuvo un valor neto de USD 545,903 con una cantidad de desmonte de 63.9 millones de toneladas y 12.3 millones de toneladas de mineral con una ley promedio CU% 0.58, AUPPM0.28 y AGPPM 3.34, así como se muestra en la Figura 64, Figura 65 y Figura 66.

Figura 62

Vista 3D-Planta: Pit Final del tajo Pampacancha

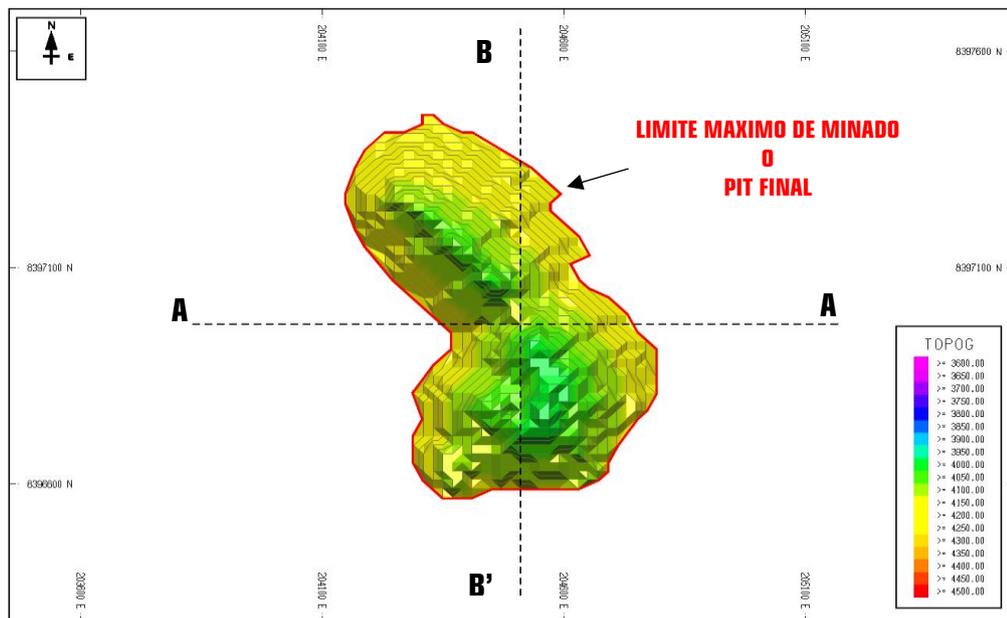


Figura 63

Vista 2D-Seccion A-A': Pit Final del tajo Pampacancha

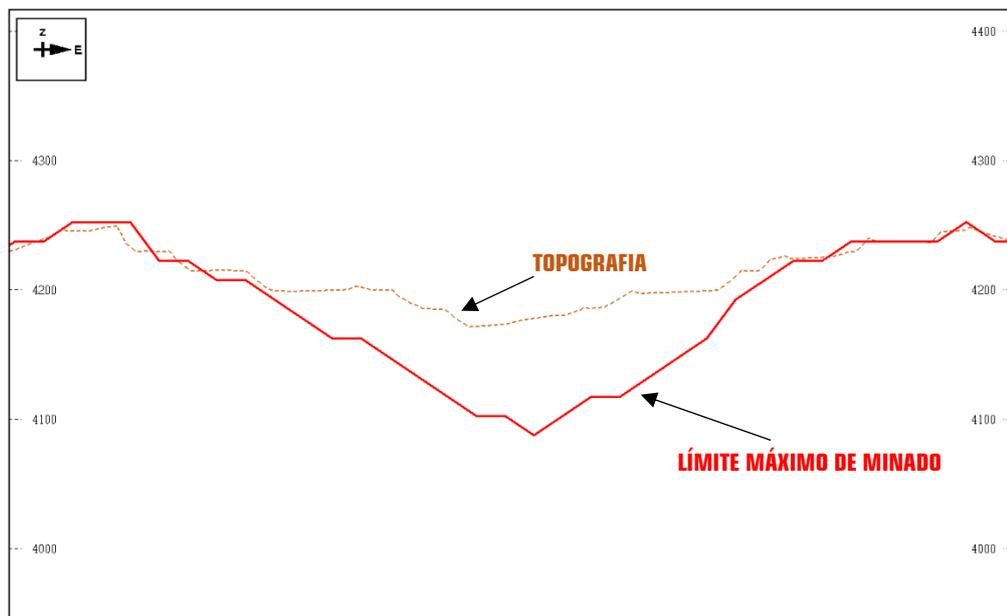


Figura 64

Vista Planta: Pit Final-Modelo de bloques.

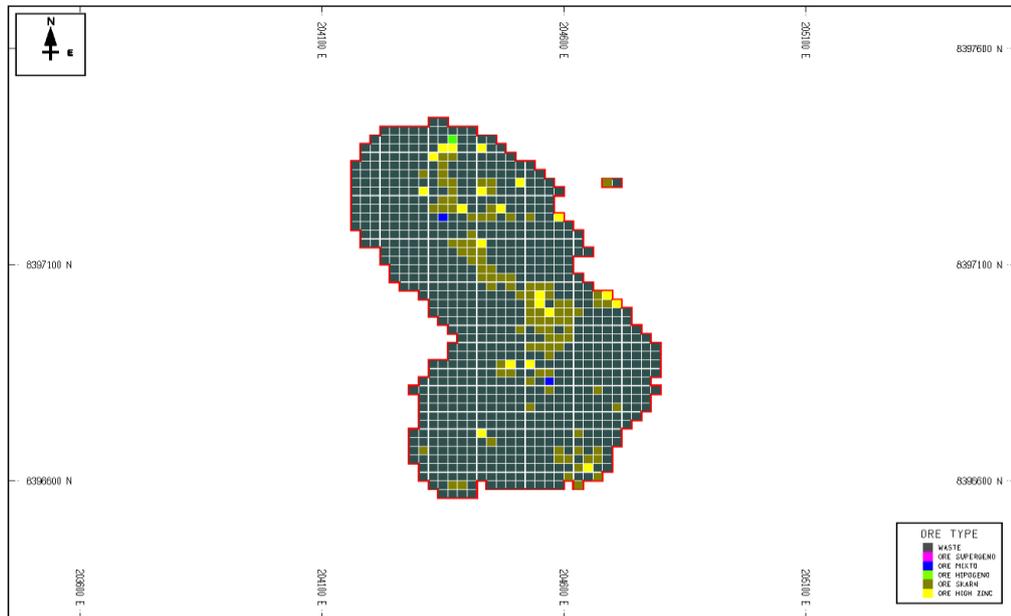


Figura 65

Vista Sección A-A': Pit Final-Modelo de bloques

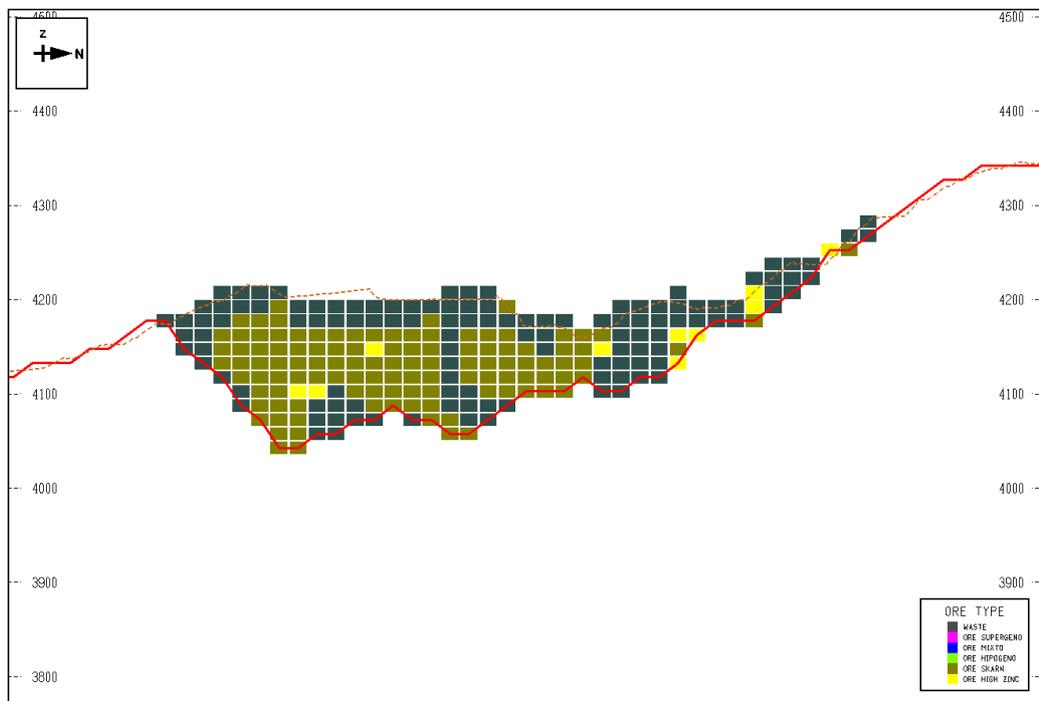
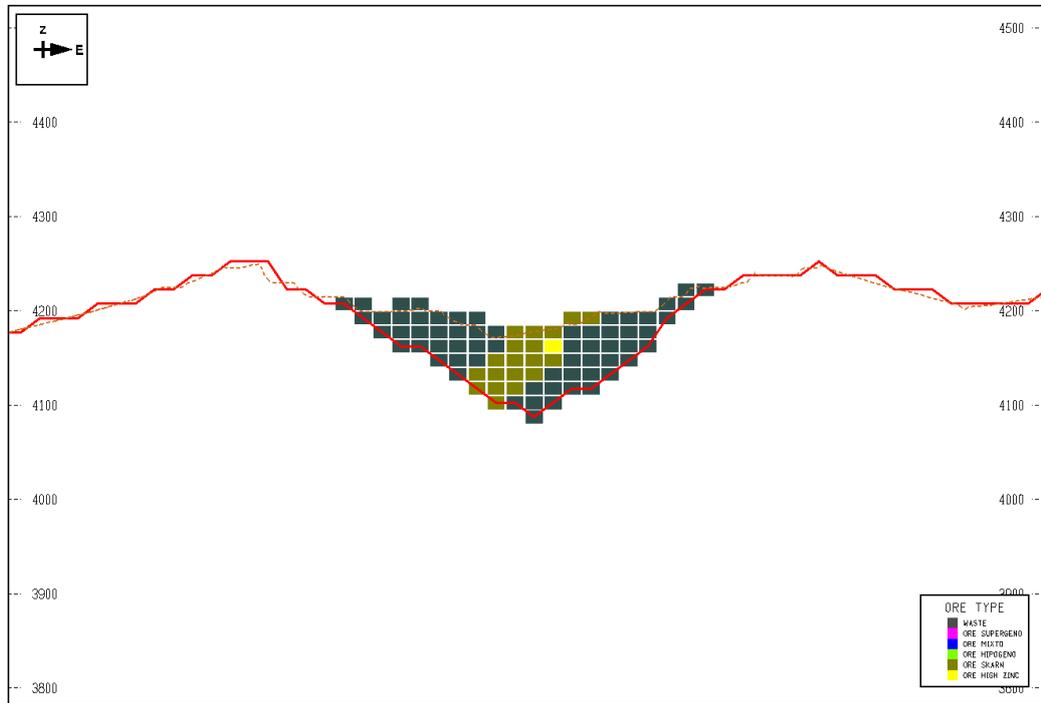


Figura 66

Vista Sección B-B': Pit Final-Modelo de bloques





V. CONCLUSIONES

En este estudio, se ha logrado desarrollar de manera exitosa un aplicativo en Python en el cual se implementó el algoritmo original de Lerchs y Grossmann y la metodología de cono plantilla para ángulos variables. Esto ha permitido abordar con eficacia el diseño óptimo y la maximización del valor neto del pit final, específicamente en yacimientos de cobre tipo skarn. La utilización de esta herramienta open-source ha demostrado su potencial para resolver el problema del pit final en la industria minera sin depender de costosas licencias de software y tener flexibilidad y accesibilidad, lo que facilita su implementación y uso por parte de profesionales de la minería.

La implementación exitosa del algoritmo de Lerchs y Grossmann original en el lenguaje de programación Python se ha demostrado como una solución no efectiva para abordar la optimización del pit final en minas a cielo abierto, ya que este algoritmo originalmente no considera ángulos variables. Lo que evidencia su uso solamente para fines académicos.

La exitosa implementación de la metodología de cono plantilla al algoritmo de Lerchs y Grossmann original en el lenguaje de programación Python representa un avance significativo en la optimización del pit final. Esta implementación ha demostrado su capacidad para abordar la complejidad del problema del pit final al considerar de manera efectiva los cambios en la inclinación del tajo, las variaciones geológicas y las estructuras del yacimiento. La herramienta desarrollada no solo ha superado los desafíos técnicos asociados con la incorporación de ángulos variables, sino que también ha proporcionado una solución práctica y flexible para adaptarse a las condiciones cambiantes de los yacimientos minerales. Esto ha mejorado la eficiencia y la precisión en la delimitación del pit final, lo que a su vez se traduce en una explotación más rentable y sostenible de los recursos minerales.



La determinación del diseño del pit final con la maximización del valor neto mediante el uso del aplicativo desarrollado en Python se ha revelado como un enfoque eficiente y efectivo en la planificación de minas a cielo abierto. Esta metodología ha permitido optimizar de manera precisa y rentable el pit final del tajo Pampacancha en donde se obtuvo un valor neto de U\$D 545,903 con una cantidad de desmonte de 63.9 millones de toneladas y 12.3 millones de toneladas de mineral con una ley promedio CU% 0.58, AUPPM0.28 y AGPPM 3.34.

Casi todos los algoritmos informáticos se basan en un modelo de bloques del depósito, que requiere una gran memoria para almacenar las características de los bloques. Los recientes avances tanto en el hardware como en el software de los PC han sido tan rápidos que muchas aplicaciones, incluidas las de la industria minera, aún no han aprovechado el enorme potencial del que ahora disponen. El software OPTIMINING, escrito en Python y presentado en el capítulo 4, es una aplicación que funciona en sistemas operativos de 64 bits, como Windows 10 y 11. La aplicación del software al diseño óptimo de tajos abiertos es muy sencilla y pueden utilizarla personas no profesionales de la informática. Gracias a un estilo de programación diferente y sencillo de entender.

En resumen, esta investigación ha corroborado que la utilización de la librería de Lerchs y Grossmann en Python es una estrategia eficaz para determinar el diseño óptimo del pit final en minas a cielo abierto, lo que se traduce en la maximización del valor neto de los proyectos mineros. Este enfoque contribuye al avance y la mejora de las prácticas en la industria minera, promoviendo la rentabilidad y la sostenibilidad en la explotación de recursos minerales.



VI. RECOMENDACIONES

Los principales objetivos de este trabajo eran resolver el problema del pit final y de taludes variables, incorporar el diseño de los taludes en el diseño óptimo de la explotación a cielo abierto e implantar los métodos en un software general basado en PC. Ahora es posible utilizar el algoritmo de Lerchs-Grossmann con ángulos de talud variables en la planificación del límite óptimo del tajo para yacimientos grandes y complejos en los que los taludes del tajo pueden variar a lo largo del yacimiento en cada fase de la explotación. Sin embargo, aún pueden introducirse muchas mejoras y a continuación se resumen las recomendaciones para futuros trabajos.

- Los caminos de acarreo son otra característica importante del diseño de minas a cielo abierto que no se ha tenido en cuenta en este estudio. Se recomienda investigar un método para incorporar los caminos de acarreo y añadirlo al software. Por ejemplo, el método de Onur y Dowd (1993) podría adaptarse fácilmente e incorporarse al software aquí descrito.
- Desarrollar e incorporar una representación gráfica tridimensional para la visualización del tajo óptimo y la topografía.
- Desarrollar e incorporar una visualización gráfica de los límites del tajo óptimo en secciones transversales para cualquier dirección.
- Añadir e incorporar información geotécnica para el diseño de los ángulos de las caras de los bancos.
- Es recomendable la estandarización del uso de la versión del software entre el área de servicios técnicos (Minas) y la consultoría; puesto que versiones diferentes conducen a resultados con sesgo, tal es el caso de MineSight, exactamente, la herramienta MS Reserve.



VII. REFERENCIAS BIBLIOGRÁFICAS

- Alford, C., & Whittle, J. (1986). Application Of Lerchs-Grossman Pit Optimization To The Design Of Open Pit Mines. *The AuslMM-IE Aust Newman Combined Group*.
- Benito Lopez, R. (2013). Determination of Optimal Mining Limits with Database Applications. *Mining Engineering*, 65.
- Bhattacharyaa, J. (2010). *Principles of Mine Planning*. Kharagpur: Allied Publishers PVT. LTD.
- Cacceta, L., & Giannini, L. M. (1987). An Application of Discrete Mathematics in the Desgin of an Open Pit Mine. *Elsevier Science Publishers*, 19.
- Carlos Lopez, J., & Manuel Bustillo, R. (1997). *Manual de Evaluación y Diseño de Explotaciones Mineras*. Madrid: Entorno Gráfico, S.L.
- Castillo, L. (2009). Modelos de Optimización para la Planificacion Minera a Cierlo Abierto.
- Chura, H. (2019). *Optimizar el Planeamiento a Largo Plazo de la Mina a Tajo Abierto Tacaza-CIEMSA Empleando los Softwares Mineros Gemcom Whittle y Minesight*. Puno: Universidad Nacional del Altiplano, Departamento de Ingenieria de Minas.
- Couzens, T. (1979). Aspects of Production Planning: Operating Layout and Phase Plans. *New York: SME-AIME*.
- Dowd, P. &. (1993). Open-Pit Optimization Part 1: Optimal Open Pit Design. *ResearchGate*.
- Dowd, P., & Onur, A. (1993). Open-Pit Optimization Part 2: Production Scheduling and Inclusion of Roadways. *Research Gate*, 50(3).
- Garcia, G. A. (2017). Planeamiento Estratégico a Cielo Abierto con Temporalidad de Modelo Económico. *Instituto de Ingenieros de Minas del Peru*.



- Hustrulid, W., Kuchta, M., & Martin, R. (2013). *Open Pit Mine Planning and Design* (Vols. 1-Fundamentals). Great Britain: CRC Press/Balkema.
- Johnson, T. (1971). Three-Dimensional Dynamic Programming Method for Optimal Ultimate Open Pit Design. *U S Bur Mines, Rep Invest 7553*.
- Khalokakaie, R., Dowd, P., & Fowell, R. (2000). Lerchs-Grossmann Algorithm with Variable Slope Angles. *Research Gate*.
- Kim, Y. (1979). Ultimate Pit Limit Design Methodologies Using Computer Models - The State of the Art. *The American Institute of Mining, Metallurgical, and Petroleum Engineers, 40(3)*.
- Latorre Nanjarí, E. O. (2015). Incorporación del valor del dinero en el tiempo para la determinación de límites económicos de un open pit. *Boletín de Ciencias de la Tierra, 184*.
- Lerchs, H., & Grossman, I. (1965). Optimum Design of Open Pit Mines. *Joint CORS and ORSA Conference, Montreal: Canadian Institute of Mining and Metallurgy*.
- Loor, V. E. (2020). Aplicación de la Inteligencia Artificial para el Agendamiento y Optimización del Diseño de Fases en Minería a Cielo Abierto.
- Morales, N., Jélvez, E., Nancel-Penard, P., Marinho, A., & Guimarães, O. (2015). A Comparison of Conventional and Direct Block Scheduling Methods for Open Pit Mine Production Scheduling. *APCOM, 1040 - 1051*.
- Nelis, G. (2016). *Planificación adaptativa utilizando simulación geoestadística antitética*. Santiago de Chile: Repositorio Académico - Universidad de Chile.
- Opazo, F. A. (2011). *Definición de Pit final Capacitado Bajo Incertidumbre*. Universidad de Chile, Departamento de Ingeniería de Minas.
- Paihua, T. G. (2010, Abril). Diseño de Minas a Tajo Abierto. 163.



- Picard, J. (1976). Maximal Closure of a Graph and Applications to Combinatorial Problems. *International Journal of Mining and Geological Engineering*, 1268-1272.
- Underwood, R., & Tolwinski, B. (1996). A mathematical programming viewpoint for solving the ultimate pit problem. *European Journal of Operational Research*.
- Whittle, J. (1998). *Four -X User Manual*. Melbourne, Australia: Whittle Programming Pty Ltd.



ANEXOS

Anexo 1. Script para Valorización Del Modelo Geológico

```
1 from grail.data import model
2 from grail.data import pcf
3 import time
4 import sys
5 import os
6 import string
7 from grail import const
8 from grail import gsys
9 from grail import messages
10 try:
11     from grail.ms3d import mssys
12     sys.stdout = mssys.stdout # re-direct 'print' statements
13 except ImportError:
14     pass
15
16 """ MODEL FILE 15 SPECIAL CALCULATIONS
17
18 To update all blocks of the model. Enhanced to speed up the process
19 by defining a model/slab for each item of interest in the model instead of
20 one model with multiple items.
21
22 """
23 === Project Mapping and Input/Output variables =====
24 PCFname = "C:\\ms_hudbay\\MODE10.DAT"
25 FILE15name = "MODE15.DAT"
26 #Map model file 15 input/output variablesA
27 #Input variables
28 i1 = "SG"
29 i2 = "TOPO"
30 i3 = "MATPS"
31 i4 = "CUPCT"
32 i5 = "AUPPM"
33 i6 = "RECCU"
34 i7 = "AGPPM"
35 i8 = "MOPCT"
36 i9 = "PBPCT"
37 i10 = "FEPCT"
38 i11 = "ZNPCT"
39 i12 = "CUSS"
40
41 #Output variables
42 o1 = "DPFT"
43 o2 = "PRFT"
44 o3 = "REVN"
45 o4 = "PCST"
46 o5 = "MCST"
47 o6 = "DEST"
48 o7 = "PPTN"
49 o8 = "RECU"
50 o9 = "REAU"
51 o10 = "REAG"
```



```
52 o11 = "REMO"
53
54 ##### CHANGE THESE TO FIT PROJECT #####
55
56 def modelUpdate():
57     # for speed, make globals FILE15name & PCFname local because they are used
58     # in loop below.
59     pcfname = PCFname
60     file15name = FILE15name
61     ti1 = i1
62     ti2 = i2
63     ti3 = i3
64     ti4 = i4
65     ti5 = i5
66     ti6 = i6
67     ti7 = i7
68     ti8 = i8
69     ti9 = i9
70     ti10 = i10
71     ti11 = i11
72     ti12 = i12
73     to1 = o1
74     to2 = o2
75     to3 = o3
76     to4 = o4
77     to5 = o5
78     to6 = o6
79     to7 = o7
80     to8 = o8
81     to9 = o9
82     to10 = o10
83     to11 = o11
84
85     p = pcf.Pcf(pcfname)
86     nz = p.nz() # Returns the number of blocks (int) in the z-direction.
87     ny = p.ny() # Returns the number of blocks (int) in the y-direction.
88     nx = p.nx() # Returns the number of blocks (int) in the x-direction.
89     bvol = p.dx()*p.dy()*p.dz() # block volume in cuft
90
91     innerRange = range(1, nx+1)
92     start = time.clock()
93     print 'Updating model. Please wait...'
94
95     for lvl in range(1, nz+1): # loop through all levels: (1,nz+1)
96         print "Calculating block values for level ", lvl
97
98         mi1 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [ti1])
99         mi2 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [ti2])
100        mi3 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [ti3])
101        mi4 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [ti4])
102        mi5 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [ti5])
103        mi6 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [ti6])
104        mi7 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [ti7])
105        mi8 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [ti8])
106        mi9 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [ti9])
107        mi10 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [ti10])
108        mi11 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [ti11])
```



```
109     mi12 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [ti12])
110     mo1 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [to1])
111     mo2 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [to2])
112     mo3 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [to3])
113     mo4 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [to4])
114     mo5 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [to5])
115     mo6 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [to6])
116     mo7 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [to7])
117     mo8 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [to8])
118     mo9 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [to9])
119     mo10 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [to10])
120     mo11 = model.Model(pcfname, file15name, lvl, lvl, 1, ny, 1, nx, [to11])
121
122     modifyModel(mi1, mi2, mi3, mi4, mi5, mi6, mi7, mi8, mi9, mi10, mi11,
123               mi12, mo1, mo2, mo3, mo4, mo5, mo6, mo7, mo8, mo9, mo10, mo11, bv1,
124               lvl, ny, innerRange)
125
126     print "Model has been updated. cpu time used by c code in
127     grail.data.model: %f" % (time.clock()-start)
128
129 def modifyModel(mi1, mi2, mi3, mi4, mi5, mi6, mi7, mi8, mi9, mi10, mi11, mi12,
130               mo1, mo2, mo3, mo4, mo5, mo6, mo7, mo8, mo9, mo10, mo11, bv1, lvl, ny,
131               innerRange):
132
133     slabi1 = mi1.slab()
134     slabi2 = mi2.slab()
135     slabi3 = mi3.slab()
136     slabi4 = mi4.slab()
137     slabi5 = mi5.slab()
138     slabi6 = mi6.slab()
139     slabi7 = mi7.slab()
140     slabi8 = mi8.slab()
141     slabi9 = mi9.slab()
142     slabi10 = mi10.slab()
143     slabi11 = mi11.slab()
144     slabi12 = mi12.slab()
145     slabo1 = mo1.slab()
146     slabo2 = mo2.slab()
147     slabo3 = mo3.slab()
148     slabo4 = mo4.slab()
149     slabo5 = mo5.slab()
150     slabo6 = mo6.slab()
151     slabo7 = mo7.slab()
152     slabo8 = mo8.slab()
153     slabo9 = mo9.slab()
154     slabo10 = mo10.slab()
155     slabo11 = mo11.slab()
156
157     for row in range(1, ny+1):
158         for col in innerRange:
159             # get the values for input variables at (lvl,row,col)
160             # normally all input variables should be defined
161             dens = slabi1.modget(0, lvl, row, col)
162             topo = slabi2.modget(0, lvl, row, col)
163             rclas = slabi3.modget(0, lvl, row, col)
164             tcu = slabi4.modget(0, lvl, row, col)
165             tau = slabi5.modget(0, lvl, row, col)
```



```
166         recau = slabi6.modget(0, lvl, row, col)
167         tag = slabi7.modget(0, lvl, row, col)
168         tmo = slabi8.modget(0, lvl, row, col)
169         tpb = slabi9.modget(0, lvl, row, col)
170         tfe = slabi10.modget(0, lvl, row, col)
171         tzn = slabi11.modget(0, lvl, row, col)
172         tcuss = slabi12.modget(0, lvl, row, col)
173
174         if dens < 0.00: dens = 2.57
175         if topo < 0.00: topo = 0.00
176         if tcu < 0.00: tcu = 0.00
177         if tau < 0.00: tau = 0.00
178
179
180         #calculate the block value
181         # Discount factor
182         bdisc = float(0.83)
183         dfactor = 1.0/(1.0+bdisc/100.0)**(lvl)
184         # block tonnes
185         bton = bvol*dens*topo/100.0
186         # Variable mining costs
187         # by level (incremental)
188         base = int(13) #bench 4145 crusher elevation
189         nb = abs(lvl - base)
190         if lvl > base:
191             # apply incr. cost below
192             incmcost = nb*(0.010)
193         else:
194             # apply incr. cost above
195             incmcost = nb*(0.004)
196         #perform calculations
197         *** costs if block is treated as waste ***#
198         mcostw=(1.550+incmcost)*bton #MCST
199         bvwaste = -mcostw # PRFT if waste
200         dbvwaste = bvwaste*dfactor # DPFT if waste
201         if bton > 0.00:
202             pptnw=bvwaste/bton # PPTN if waste
203         else:
204             pptnw=0.00
205         *** profits and costs if block is treated as ore ***#
206         # MATPS: Considerar como mineral el 25 y 26;
207
208         #Metal recoreveries
209         reccu = 0.00
210         recau = 0.00
211         recag = 0.00
212         recmo = 0.00
213         #Concentrate metal recoveries
214         reccuc = 0.00
215         recauc = 0.00
216         recagc = 0.00
217         recmoc = 0.00
218
219         # Select only blocks that are within the category of ORE
220         # SUPERGENE ORE = 5
221         # MIX ORE = 6
222         # HYPOGENE ORE = 7
```



```
223 # SKARN ORE = 8, 25 y 26
224 # HIGH ZINC ORE = 9
225 if (rclas>=5 and rclas<=9 and bton>0.0) or (rclas>=25 and
226 rclas<=26 and bton>0.0):
227     if rclas==5:
228         #Mill Process - Recoveries by Otype
229         reccu = (min(90.5, 88.73 - 0.02*155 + 13.19*tcu -
230 30.87*(tpb+tztn) - 36*(tcuss/tcu)+2.5))/100.0
231         if (min(50, 0.54*reccu*100.0 + 99.94*tau + 69.55*tpb -
232 46.21*tztn - 3.01))/100.0 > 0.0:
233             recau = (min(50, 0.54*reccu*100.0 + 99.94*tau +
234 69.55*tpb - 46.21*tztn - 3.01))/100.0
235         else:
236             recau = 0.0
237         recag = (min(72, 1.3*reccu*100.0 - 5.3*(tztn+tfe)/tag -
238 37.52))/100.0
239         recmo = (min(50, (1.65*reccu*100.0 + 43.32*(tmo/tzn) -
240 96.93)*0.66 ))/100.0
241
242         #Mill Process - Concentrate Grades by Otype
243         reccuc = (24.35 + 5.05*tcu - 8.45*tztn - 0.42*tfe)/100.0
244         brtoncu = bton*tcu*reccu/reccuc/100.00
245         recauc = ((bton*tau*recau)/31.1034)*31.1034/brtoncu
246         recagc = ((bton>tag*recag)/31.1034)*31.1034/brtoncu
247         recmoc = 0.50
248
249     if rclas==6:
250         #Mill Process - Recoveries by Otype
251         reccu = (min(90.5, 88.73 - 0.02*155 + 13.19*tcu -
252 30.87*(tpb+tztn) - 36*(tcuss/tcu)+2.5))/100.0
253         recau = (min(50, 0.54*reccu*100.0 + 99.94*tau +
254 69.55*tpb - 46.21*tztn - 3.01))/100.0
255         if (min(72, 1.3*reccu*100.0 - 5.3*(tztn+tfe)/tag -
256 37.52))/100.0 > 0.0:
257             recag = (min(72, 1.3*reccu*100.0 - 5.3*(tztn+tfe)/tag -
258 37.52))/100.0
259         else:
260             recag = 0.00
261         recmo = 0.00
262         #Mill Process - Concentrate Grades by Otype
263         reccuc = (24.35 + 5.05*tcu - 8.45*tztn - 0.42*tfe)/100.0
264         brtoncu = bton*tcu*reccu/reccuc/100.00
265         recauc = ((bton*tau*recau)/31.1034)*31.1034/brtoncu
266         recagc = ((bton>tag*recag)/31.1034)*31.1034/brtoncu
267         recmoc = 0.00
268
269     if rclas==7:
270         #Mill Process - Recoveries by Otype
271         reccu = (min(90.5, 88.73 - 0.02*155 + 13.19*tcu -
272 30.87*(tpb+tztn) - 36*(tcuss/tcu)+2.5))/100.0
273         recau = (min(50, 0.54*reccu*100.0 + 99.94*tau +
274 69.55*tpb - 46.21*tztn - 3.01))/100.0
275         if (min(72, 1.3*reccu*100.0 - 5.3*(tztn+tfe)/tag -
276 37.52))/100.0 > 0.0:
277             recag = (min(72, 1.3*reccu*100.0 - 5.3*(tztn+tfe)/tag -
278 37.52))/100.0
279         else:
```



```
280         recag = 0.0
281     recmo = (min(50, (1.65*reccu*100.0 + 43.32*(tmo/tzn) -
282         96.93)*0.66 ))/100.0
283     #Mill Process - Concentrate Grades by Otype
284     reccuc = (22.84 + 9.4*tcu - 17.48*tzn - 0.3*tfe)/100.0
285     brtoncu = bton*tcu*reccu/reccuc/100.00
286     recauc = ((bton*tau*recau)/31.1034)*31.1034/brtoncu
287     recagc = ((bton>tag*recag)/31.1034)*31.1034/brtoncu
288     recmoc = 0.50
289
290     if rclas==8 or rclas==25 or rclas==26:
291         #Mill Process - Recoveries by Otype
292         reccu = (min(90.5, 88.73 - 0.02*155 + 13.19*tcu -
293         30.87*(tpb+tzn) - 36*(tcuss/tcu)+2.5))/100.0
294         recau = (min(50, 0.54*reccu*100.0 + 99.94*tau +
295         69.55*tpb - 46.21*tzn - 3.01))/100.0
296         # recag = (min(72, 1.3*reccu*100.0 - 5.3*(tzn+tfe)/tag -
297         37.52))/100.0
298         if (min(72, 1.3*reccu*100.0 - 5.3*(tzn+tfe)/tag -
299         37.52))/100.0 > 0.0:
300             recag = (min(72, 1.3*reccu*100.0 - 5.3*(tzn+tfe)/tag -
301             37.52))/100.0
302         else:
303             recag = 0.0
304         recmo = (min(50, (1.65*reccu*100.0 + 43.32*(tmo/tzn) -
305         96.93)*0.4 ))/100.0
306         #Mill Process - Concentrate Grades by Otype
307         reccuc = (23.06 + 4.6*tcu - 5.64*tzn - 0.17*tfe)/100.0
308         brtoncu = bton*tcu*reccu/reccuc/100.00
309         recauc = ((bton*tau*recau)/31.1034)*31.1034/brtoncu
310         recagc = ((bton>tag*recag)/31.1034)*31.1034/brtoncu
311         recmoc = 0.50
312
313     if rclas==9:
314         #Mill Process - Recoveries by Otype
315         reccu = (min(90.5, 88.73 - 0.02*155 + 13.19*tcu -
316         30.87*(tpb+tzn) - 36*(tcuss/tcu)+2.5))/100.0
317         recau = (min(50, 0.54*reccu*100.0 + 99.94*tau +
318         69.55*tpb - 46.21*tzn - 3.01))/100.0
319         if (min(72, 1.3*reccu*100.0 - 5.3*(tzn+tfe)/tag -
320         37.52))/100.0 > 0.0:
321             recag = (min(72, 1.3*reccu*100.0 - 5.3*(tzn+tfe)/tag -
322             37.52))/100.0
323         else:
324             recag = 0.0
325         recmo = 0.00
326         #Mill Process - Concentrate Grades by Otype
327         reccuc = (23.06 + 4.6*tcu - 5.64*tzn - 0.17*tfe)/100.0
328         brtoncu = bton*tcu*reccu/reccuc/100.00
329         # print(row,col,lvl, bton)
330         recauc = ((bton*tau*recau)/31.1034)*31.1034/brtoncu
331         recagc = ((bton>tag*recag)/31.1034)*31.1034/brtoncu
332         recmoc = 0.00
333
334     brtonau = bton*tau*recau/recauc
335     brtonag=0.0
336     if recag>0.0 or recagc >0.0:
```



```
337         brtonag = bton*tag*recag/recage
338     brtonmo = 0.0
339     if recmo>0.0 or recmoc >0.0:
340         brtonmo = bton*tmo*recmo/recmoc/100.00
341
342     # Contained Metal per Dry Tonne of Concentrate
343     CMDTCcu = (reccuc-1/100.00)*2204.623
344     CMDTCau = recauc/31.1035*0.91
345     CMDTCag = recagc/31.1035*0.90
346     CMDTCmo = recmoc*2204.623*0.98
347
348     #Revenue per metal
349     brevcu = CMDTCcu*2.6*brtoncu
350     brevau = CMDTCau*1500.0*brtonau
351     brevag = CMDTCag*18.0*brtonag
352     brevmo = CMDTCmo*11.0*brtonmo
353
354     #Total Revenue
355     brevenue=brevcu+brevau+brevag+brevmo
356     # mining costs
357     bmcost=(1.850+incmcost)*bton #MCST if ore
358     # processing costs + incremental haulage + g&a
359     bpcost=(4.54)*bton #PCST if ore
360     # G&A costs
361     bgacost=(1.60)*bton #PCST if ore
362     # Selling costs
363     bscost=(115.8084)*brtoncu + (234.06)*brtonmo
364     # Treatment costs
365     btcost=(76.73)*brtoncu + (1653.00)*brtonmo
366     # Refining costs
367     brcost=(0.0767*2204.623*reccuc)*brtoncu +
368     (0.45/31.1035*recagc)*brtonag +
369     (4.5/31.1035*recauc)*brtonau + (1.2*2204.623*recmoc)*brtonmo
370
371     # Profits
372     bprfacu=brevenue-bmcost-bpcost-bgacost-bscost-btcost-brcost #UPFT
373     bvore=bprfacu #PRFT if ore
374     dbvore=bvore*dfactor # DPFT if ore
375     if bton > 0.00:
376         pptno=bvore/bton # PPTN if ore
377     else:
378         pptno=0.00
379     ### block routing ###
380     if (bvore>bvwaste) and (bvore > 0.00):
381         bdestin=3 #ore
382     elif bvore>bvwaste:
383         bdestin=2 #marginal
384     else:
385         bdestin=1 #waste
386     else:
387     # it is a waste block
388         bdestin = 1 # waste
389
390     # store the output parameters
391     if bdestin==1:
392         ## assign output model variables as waste
393         slabol.modset(0, lvl, row, col, dbvwaste) #discounted profit
```

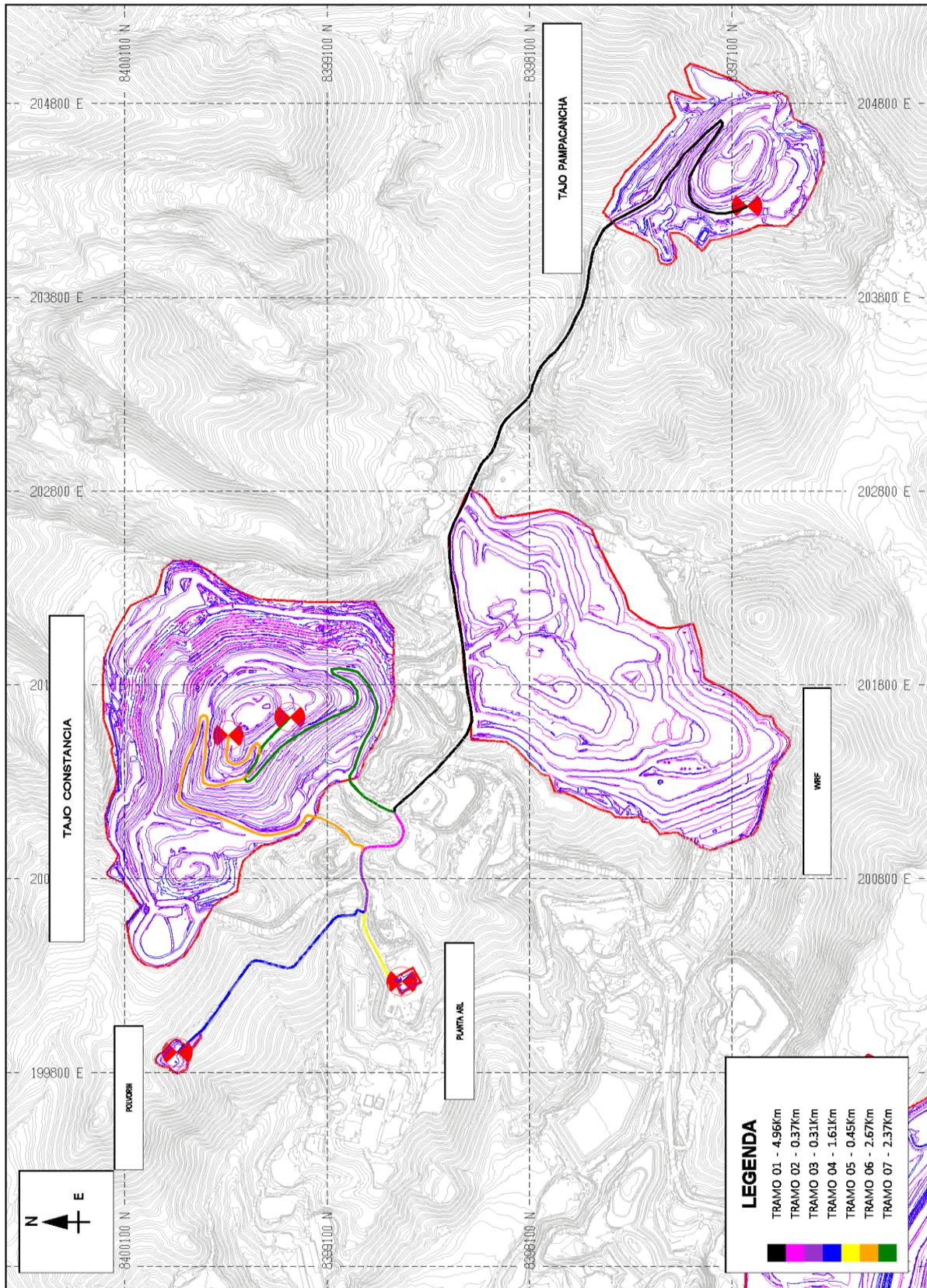


```
394         slabo2.modset(0, lvl, row, col, bwaste) #profit
395         slabo3.modset(0, lvl, row, col, 0.0) #revenue
396         slabo4.modset(0, lvl, row, col, 0.0) #processing cost
397         slabo5.modset(0, lvl, row, col, mcostw) #mining cost
398         slabo6.modset(0, lvl, row, col, bdestin) #destination
399         slabo7.modset(0, lvl, row, col, pptnw) #profit per tonne
400         slabo8.modset(0, lvl, row, col, reccu)
401         slabo9.modset(0, lvl, row, col, recau)
402         slabo10.modset(0, lvl, row, col, recag)
403         slabo11.modset(0, lvl, row, col, recmo)
404     elif bdestin>=2 and bdestin<=3:
405         ## assign output model variables as ore/marginal
406         slabo1.modset(0, lvl, row, col, dbvore) #discounted profit
407         slabo2.modset(0, lvl, row, col, bvore) #profit
408         slabo3.modset(0, lvl, row, col, brevenue) #revenue
409         slabo4.modset(0, lvl, row, col, bpcost) #processing cost
410         slabo5.modset(0, lvl, row, col, bmcost) #mining cost
411         slabo6.modset(0, lvl, row, col, bdestin) #destination
412         slabo7.modset(0, lvl, row, col, pptno) #profit per tonne
413
414         slabo8.modset(0, lvl, row, col, reccu)
415         slabo9.modset(0, lvl, row, col, recau)
416         slabo10.modset(0, lvl, row, col, recag)
417         slabo11.modset(0, lvl, row, col, recmo)
418     else:
419         ## assign output model variables as unminable (beyond prop limits)
420         slabo1.modset(0, lvl, row, col, -899999) #discounted profit
421         slabo2.modset(0, lvl, row, col, -899999) #profit
422         slabo3.modset(0, lvl, row, col, 0.0) #revenue
423         slabo4.modset(0, lvl, row, col, 0.0) #processing cost
424         slabo5.modset(0, lvl, row, col, 0.0) #mining cost
425         slabo6.modset(0, lvl, row, col, bdestin) #destination
426         slabo7.modset(0, lvl, row, col, 0.0) #profit per tonne
427
428         slabo8.modset(0, lvl, row, col, reccu)
429         slabo9.modset(0, lvl, row, col, recau)
430         slabo10.modset(0, lvl, row, col, recag)
431         slabo11.modset(0, lvl, row, col, recmo)
432
433     mo1.storeslab()
434     mo2.storeslab()
435     mo3.storeslab()
436     mo4.storeslab()
437     mo5.storeslab()
438     mo6.storeslab()
439     mo7.storeslab()
440
441     mo8.storeslab()
442     mo9.storeslab()
443     mo10.storeslab()
444     mo11.storeslab()
445
446 def gmain(msg, data):
447     """We are running from ms3d."""
448     if msg == messages.gPRE_RUN:
449         pass # do pre-processing.
450     elif msg == messages.gRUN:
```



```
451     modelUpdate()
452     elif msg == messages.gPOST_RUN:
453         pass          # do post processing.
454     else:
455         return gsys.grailmain(msg, data)
456
457 if __name__=="__main__":
458     # We are running from the command line.
459     modelUpdate()
```

Anexo 2. Plano De Ubicación, Tajo Constancia Y Pampacancha





DECLARACIÓN JURADA DE AUTENTICIDAD DE TESIS

Por el presente documento, Yo Yovani Achata Flores
, identificado con DNI 70000289 en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado

Ingeniería de Minas

, informo que he elaborado el/la Tesis o Trabajo de Investigación para la obtención de Grado

Título Profesional denominado:

"Implementación en Python del Algoritmo de Lerchs y Grossmann Para la Optimización del Pit Final en Minas a Cielo Abierto en Yacimientos de Cobre Tipo SKarn"
" Es un tema original.

Declaro que el presente trabajo de tesis es elaborado por mi persona y **no existe plagio/copia** de ninguna naturaleza, en especial de otro documento de investigación (tesis, revista, texto, congreso, o similar) presentado por persona natural o jurídica alguna ante instituciones académicas, profesionales, de investigación o similares, en el país o en el extranjero.

Dejo constancia que las citas de otros autores han sido debidamente identificadas en el trabajo de investigación, por lo que no asumiré como tuyas las opiniones vertidas por terceros, ya sea de fuentes encontradas en medios escritos, digitales o Internet.

Asimismo, ratifico que soy plenamente consciente de todo el contenido de la tesis y asumo la responsabilidad de cualquier error u omisión en el documento, así como de las connotaciones éticas y legales involucradas.

En caso de incumplimiento de esta declaración, me someto a las disposiciones legales vigentes y a las sanciones correspondientes de igual forma me someto a las sanciones establecidas en las Directivas y otras normas internas, así como las que me alcancen del Código Civil y Normas Legales conexas por el incumplimiento del presente compromiso

Puno 08 de setiembre del 2023

FIRMA (obligatoria)



Huella



AUTORIZACIÓN PARA EL DEPÓSITO DE TESIS O TRABAJO DE INVESTIGACIÓN EN EL REPOSITORIO INSTITUCIONAL

Por el presente documento, Yo Jovani Achata Flores
identificado con DNI 70000289 en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado

Ingeniería de Minas

, informo que he elaborado el/la Tesis o Trabajo de Investigación para la obtención de Grado

Título Profesional denominado:

"Implementación en Python del Algoritmo de Lerchs y Grossmann Para la Optimización del Pit final en Minas a Cielo Abierto en Yacimientos de Cobre Tipo SKarn"

" Por medio del presente documento, afirmo y garantizo ser el legítimo, único y exclusivo titular de todos los derechos de propiedad intelectual sobre los documentos arriba mencionados, las obras, los contenidos, los productos y/o las creaciones en general (en adelante, los "Contenidos") que serán incluidos en el repositorio institucional de la Universidad Nacional del Altiplano de Puno.

También, doy seguridad de que los contenidos entregados se encuentran libres de toda contraseña, restricción o medida tecnológica de protección, con la finalidad de permitir que se puedan leer, descargar, reproducir, distribuir, imprimir, buscar y enlazar los textos completos, sin limitación alguna.

Autorizo a la Universidad Nacional del Altiplano de Puno a publicar los Contenidos en el Repositorio Institucional y, en consecuencia, en el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, sobre la base de lo establecido en la Ley N° 30035, sus normas reglamentarias, modificatorias, sustitutorias y conexas, y de acuerdo con las políticas de acceso abierto que la Universidad aplique en relación con sus Repositorios Institucionales. Autorizo expresamente toda consulta y uso de los Contenidos, por parte de cualquier persona, por el tiempo de duración de los derechos patrimoniales de autor y derechos conexos, a título gratuito y a nivel mundial.

En consecuencia, la Universidad tendrá la posibilidad de divulgar y difundir los Contenidos, de manera total o parcial, sin limitación alguna y sin derecho a pago de contraprestación, remuneración ni regalía alguna a favor mío; en los medios, canales y plataformas que la Universidad y/o el Estado de la República del Perú determinen, a nivel mundial, sin restricción geográfica alguna y de manera indefinida, pudiendo crear y/o extraer los metadatos sobre los Contenidos, e incluir los Contenidos en los índices y buscadores que estimen necesarios para promover su difusión.

Autorizo que los Contenidos sean puestos a disposición del público a través de la siguiente licencia:

Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

En señal de conformidad, suscribo el presente documento.

Puno 08 de setiembre del 2023

FIRMA (obligatoria)



Huella