

- * About WordPress
 - o About WordPress
 - o WordPress.org
 - o Documentation
 - o Support
 - o Feedback
- * MiningMath
 - o Dashboard
 - o Themes
 - o Widgets
 - o Menus
- * Customize
- * Theme Settings
- * 4949 updates available
- * 00 Comments in moderation
- * Page Views: 31 - Online: 1
 - o Today's Visitors: 1
 - o Today's Visits: 4541
 - o Yesterday's Visitors: 1
 - o Yesterday's Visits: 7107
 - o Unlock Mini Chart!
- * New
 - o Post
 - o Media
 - o Page
 - o Forum
 - o Topic
 - o Reply
 - o Landing Page
 - o Template
 - o Order

- o Map
- o Menu
- o User Feedback
- o Doc
- o Downloads
- o Poll
- o User
- o WPForms
- * Edit with Elementor
- o Mining Optimization AlgorithmPost
- o Getting StartedPost
- o Quick CheckPost
- o Put It to Run!Post
- o How to Run a ScenarioPost
- o Results of the OptimizationPost
- o Current Best PracticesPost
- o Home PagePost
- o Model TabPost
- o Scenario TabPost
- o ViewerPost
- o Formatting the Block ModelPost
- o Importing the Block ModelPost
- o Data ValidationPost
- o Super Best CasePost
- o Optimized PushbacksPost
- o Short-term PlanningPost
- o Exporting DataPost
- o Play with Predefined ScenariosPost
- o AcademyPost
- o TeachingPost
- o Press ReleasesPost

- o [Public DatasetsPost](#)
- o [The CalculatorPost](#)
- o [Slope AnglesPost](#)
- o [DestinationsPost](#)
- o [RecoveryPost](#)
- o [StockpilesPost](#)
- o [Stocked blocksPost](#)
- o [Artificial stockpilesPost](#)
- o [Reclaim PolicyPost](#)
- o [Variable Mining CostsPost](#)
- o [Production ConstraintsPost](#)
- o [Geometric ConstraintsPost](#)
- o [Vertical RatePost](#)
- o [Widths and lengthsPost](#)
- o [Surface ConstraintsPost](#)
- o [Force MiningPost](#)
- o [Restrict MiningPost](#)
- o [Combining Force and RestrictPost](#)
- o [Creating SurfacesPost](#)
- o [Validating SurfacesPost](#)
- o [Surfaces as a GuidePost](#)
- o [AveragePost](#)
- o [SumPost](#)
- o [Material TypesPost](#)
- o [Mining FrontsPost](#)
- o [OverviewPost](#)
- o [Save AsPost](#)
- o [RunPost](#)
- o [Output filesPost](#)
- o [Tips and TricksPost](#)
- o [Coordinates to index conversionPost](#)

- o DilutionPost
- o Keep your Original DataPost
- o Operational SolutionsPost
- o Optimization HintsPost
- o Optimization RuntimePost
- o Optimizing your WorkflowPost
- o Percent ModelsPost
- o Stochastic ModelsPost
- o Cut-Off GradesPost
- o IntegrationPost
- o NPV CalculationPost
- o Multi-minePost
- o Sustainable analysisPost
- o Uncertainties at the BeginningPost
- o Warning MessagesPost
- o Common IssuesPost
- o Decision TreesPost
- o Economic ValuesPost
- o Evaluate Project PotentialPost
- o Optimized SchedulesPost
- o Constraints ValidationPost
- o Fixing DestinationsPost
- o Optimizing ScenariosPost
- o FAQPost
- o Get StartedPost
- o Practice First, Optimizing Scenarios, Decision Trees, Theory Behind, Interface overviewPost
- o Economic ValuesPost
- o Formatted examplePost
- o Importing DataPost
- o Progress BarPost
- o Constraints ValidationPost

o Integrated Workflow, Best Case, Exploratory Analysis, Schedule Optimization, Short-term PlanningPost

o CSV Formatting IssuesPost

o Must-Read ArticlesPost

o Time LimitPost

o MiningMath UniquenessPost

o Guaranteed SolutionsPost

o Theory ValidationPost

o In-Depth MiningMathPost

o Imperial SystemPost

o TranslationsPost

o Floating-point numbersPost

o ReblockingPost

o NPV Upside PotentialPost

o Best-Worst Range AnalysisPost

o Selectivity AnalysisPost

o Design EnhancementPost

o NPV EnhancementPost

o Bottleneck AnalysisPost

o Multivariate Sensitivity AnalysisPost

o Ask GTPPost

o Site SettingsSite

o Theme BuilderSite

*

SEO

Focus keyphrase not set

o Focus keyphrase: not set

o SEO score:

Focus keyphrase not set

o Readability:

Not available

o Front-end SEO inspectorPremium

o Analyze this page

* Check links to this URL

* Google Rich Results Test

* Facebook Debugger

* Google Page Speed Test

* Mobile-Friendly Test

o SEO Tools

* Semrush

* Wincher

* Google trends

o How to

* Learn more SEO

* Improve your blog post

* Write better content

o Help

* Yoast.com help section

* Yoast Premium support

* WordPress.org support forums

* Learn more SEO

o SEO Settings

* General

* Settings

* Tools

* Premium

o Get Yoast SEO Premium "

* Site Kit

The Site Kit by Google plugin requires JavaScript to be enabled in your browser.

* Notes

* ExactMetrics

* WP Mail SMTP !

* WPForms

3

o Notifications

o All Forms

o Add New

o Community

o Help Docs

o Upgrade to Pro

*

Top of Form

Search

Bottom of Form

* Howdy, Lucas Rizzo

o Lucas RizzoLucas

o Edit Profile

o Log Out

Log Out

MiningMath

Loading...

All parameters simultaneously handled,delivering multiple scenarios

* Knowledge Base

* Forum

o Forum

o MM-Points

* Our Partners

o Content reviewers

- o Translators

- o Top members

- * Research

- o Published papers

- o Academic Licenses

- * About Us

Lucas Rizzo

Our Plans

Knowledge Base

Knowledge Base

- * Getting Started

- o Quick Check

- o Put It to Run!

- o Optimizing Scenarios

- * Tutorials

- o Must-Read Articles

- o In-Depth MiningMath

- * Translations

- * Ask GPT

- * Essential Topics

- o How to Run a Scenario

- o Results of the Optimization

- o Play with Predefined Scenarios

- o The Calculator

- o NPV Calculation

- o Evaluate Project Potential

- o Decision Trees

- o FAQ

- * Interface Overview

- o Home Page

- o Model Tab

- o Scenario Tab
- o Viewer
- o Overview
- o Save As
- o Run
- * Handling Data
 - o Formatting the Block Model
 - o Importing the Block Model
 - o Economic Values
 - o Data Validation
 - o Constraints Validation
 - o Exporting Data
 - o Public Datasets
 - o Output files
- * Workflow
 - o Super Best Case
 - o Optimized Pushbacks
 - o NPV Upside Potential
 - o Best-Worst Range Analysis
 - o Selectivity Analysis
 - o Design Enhancement
 - o NPV Enhancement
 - o Bottleneck Analysis
 - o Multivariate Sensitivity Analysis
 - o Optimized Schedules
 - o Short-term Planning
 - o Multi-mine
 - o Sustainable analysis
 - o Uncertainties at the Beginning
- * Theory
 - o Current Best Practices

- o MiningMath Uniqueness
- o Guaranteed Solutions
- o Theory Validation
- o Mining Optimization Algorithm
- * General Content
 - o Destinations
 - o Recovery
 - o Stockpiles
 - o Stocked blocks
 - o Artificial stockpiles
 - o Reclaim Policy
 - o Variable Mining Costs
 - o Production Constraints
 - o Average
 - o Sum
 - o Material Types
 - o Mining Fronts
 - o Stochastic Models
 - o Time Limit
- * Academy
 - o Teaching
 - o Press Releases
- * Common Issues
 - o Floating-point numbers
 - o Get Started
 - o Practice First, Optimizing Scenarios, Decision Trees, Theory Behind, Interface overview
 - o Economic Values
 - o Formatted example
 - o Importing Data
 - o Progress Bar
 - o Constraints Validation

- o Integrated Workflow, Best Case, Exploratory Analysis, Schedule Optimization, Short-term Planning

- o CSV Formatting Issues

- o Warning Messages

- * Geometry

- o Slope Angles

- o Geometric Constraints

- o Widths and lengths

- o Vertical Rate

- o Force Mining

- o Restrict Mining

- o Combining Force and Restrict

- o Creating Surfaces

- o Validating Surfaces

- o Surfaces as a Guide

- o Surface Constraints

- * Tips and Tricks

- o Reblocking

- o Cut-Off Grades

- o Integration

- o Coordinates to index conversion

- o Dilution

- o Keep your Original Data

- o Operational Solutions

- o Optimization Hints

- o Optimization Runtime

- o Optimizing your Workflow

- o Percent Models

- o Imperial System

- o Fixing Destinations

Getting Started

Articles

* Quick Check

* Put It to Run!

* Optimizing Scenarios

Quick Check

System requirements

The only mandatory requirement for using MiningMath is a 64-bits system, due to its use of Direct Block Scheduling technology. Other minimum requirements are listed further:

1. Windows 10
2. 64-bits system (mandatory)
3. 110 MB of space (installation) + additional space for your projects' files.
4. Processor: processors above 2.4 GHz are recommended to improve your experience.
5. Memory: at least 4 GB of RAM is required. 8 GB of RAM or higher is recommended to improve your experience.
6. OpenGL 3.2 or above. Discover yours by downloading and running the procedure available at [here](#)

Recommended Hardware

Memory should be a higher priority when choosing the machine in which MiningMath will be run on. In addition, a list of priority upgrades to improve performance with large scale datasets is given below.

1. Higher Ram
2. Higher Ram frequency
3. Higher processing clock

Multithreading

MiningMath utilizes the linear programming solver provided by CLP to implement the Simplex algorithm. Each execution of CLP utilizes one core/nucleus/processor because Simplex is not easily parallelizable (meaning it can't be run faster by using multiple processors). As far as our studies have shown, there hasn't been a market solution for this, and scientific research hasn't yielded very satisfactory results yet.

Put It to Run!

Installing, Activating and Running

Installing and activating MiningMath is quick and straightforward. All you need to do is follow the setup wizard and have an internet connection to activate your license.

Video 1: MiningMath installation process.

Figure 6: MiningMath installation process from left to right.

Figure 1: MiningMath installation process from left to right.

Figure 2: MiningMath installation process from left to right.

Figure 3: MiningMath installation process from left to right.

Figure 4: MiningMath installation process from left to right.

Figure 5: MiningMath installation process from left to right.

Figure 6: MiningMath installation process from left to right.

Figure 1: MiningMath installation process from left to right.

[Previous image](#)

[Next image](#)

Activating Your License

To activate your license, you will need to:

1. Open MiningMath (it will open automatically after the installation, but you can open it manually afterwards).
2. On the left menu, click License.
3. Select the field "I have an activation code" and paste the License Code provided by MiningMath.
4. Click "Activate license".

NOTE:

MiningMath's licensing method demands an internet connection.

Optimizing Scenarios

Play with the predefined scenarios

MiningMath allows you to learn with each scenario by providing standard parameters which simulate some common constraints that a mining company may face. Standard scenarios are listed and described below so you can identify the main changes made within the "Overview" tab.

The ultimate goal of this practice is to prepare you to build Decision Trees, which allow you to organize scenarios in order to understand how variables influence one another and, consequently, how these variables determine the final NPV.

Figure 1: Scenarios on the Home Page

Market Conditions Decision Tree

1) BaseCase

The Base Case consists of the initial scenario, with a uniform production capacity and without sum, average or surface mining limits.

2) BaseCase-RampUp

While the base case considers a uniform production capacity, the BaseCase-RampUp scenario offers the possibility to vary the levels of production within the different timespans. We have an initial production capacity of 10Mton on the first 2 periods; 20 Mton on periods 3 and 4; and 30 Mton from period 5 until the end of the mine's lifetime, with a total movement constraint of 30, 60, and 80 Mton, considering the increase of production within the time-frames mentioned.

Figure 3: BaseCase-RampUp

3) PriceUp and PriceDown

Scenarios "PriceUp" and "PriceDown" differ in relation to the basic scenario in the economic value used for the calculation of the P1 process, where there is an increase and a decrease of 10% at the copper selling price, respectively. In the destination tab, "P1 Cu +10" and "P1 Cu -10" were the values used for the process.

Figure 5: PriceDown

Figure 4: PriceUp

Figure 5: PriceDown

Figure 4: PriceUp

[Previous image](#)

[Next image](#)

4) PriceUp-RampUp and PriceDown-RampUp

These scenarios consider a 10% copper selling price increase and decrease, and a ramp-up of the production capacity at the same time, as mentioned before.

Figure 7: PriceDown-RampUp

Figure 6: PriceUp-RampUp

Figure 7: PriceDown-RampUp

Figure 6: PriceUp-RampUp

[Previous image](#)

[Next image](#)

5) PriceUp-RampUp-Protection300 and PriceUp-RampUp-Protection400

This scenario considers a 10% copper selling price increase and a ramp-up of the products at the same time, as mentioned in the previous scenario. In addition, a restrict mining surface (this constraint is used to prohibit access to this area in a specific timeframe) was included up to the fourth period, since it may represent some legal constraints on a project.

Figure 8: PriceUp-RumpUp-Protections

Other Decision Trees

Below you can see a description of some scenarios of other Decision Trees.

1) MW150 (Geometries Decision Trees)

The MW150 scenario considers different geometries from the base case at the geometric constraints. In this scenario, 150 meters was used as mining width (the horizontal distance between the walls of two surfaces that belonged to consecutive periods), and a vertical rate advance of 180 meters.

Figure 9: MW150

2) AvgCu (Average Decision Tree)

In the AvgCu scenario, blending constraints were added in the average tab to consider allowed at the process plant 0.5% as a minimum and 0.7% as a maximum average copper grade. The optimization will have to fulfill the P1 process capacity and, as an additional challenge, it has to meet this new set of parameters related to the Cu average content within the ore.

Figure 10: AvgCu

3) Proc13000h and Proc13000h-33Mt

(Process throughput Decision Tree)

Scenario Proc13000h considers 13.000 hours of processing equipment used as the maximum limit. This constraint was inserted at the sum tab and it controls variables such as rock type feeding, energy consumption, and any parameter controlled by its sum. Scenario Proc13000h-33Mt considers an increase of 10% in the production, inserted at the production tab beyond the parameters mentioned previously.

Figure 12: Processing hours

4) Yearly-TriannualProduction

(Short-Long Term Integration Decision Tree)

This considers a yearly production for period range 1-4 and triannual planning for range 5-end. This way is possible to integrate both short and long-term planning in a single run, facilitating the analysis and strategic definitions.

Any kind of timeframe can be used according to your needs.

Figure 13: Short-Long Term Integration

Tutorials

Articles

* Must-Read Articles

* In-Depth MiningMath

Must-Read Articles

In order to take the maximum of MiningMath's Optimization we recommend this flow through our Knowledge Base. It will guide you step-by-step in order to integrate multiple business' areas and to improve your strategic analysis through risk assessments unconstrained by step-wise processes.

Set up and first run

1. Quick Check: Here you'll have all the necessary instructions to install, activate and run MiningMath.

2. How to run a scenario: Once everything is ready, it's time to run your first scenario with MiningMath so you can familiarize with our technology!

Find new results

1. Playing with pre-defined scenarios: Each change in a scenario opens a new world of possibilities, therefore, it's time to understand a little more about and see it in practice, playing with pre-defined scenarios.

2. Decision Trees: Decision Trees provide you a detailed broad view of your project, allowing you to plan your mining sequence by analyzing every possibility in light of constraints applied to each scenario, which options are more viable and profitable to the global project, as well as how these factors impact the final NPV.

Understand the technology in depth

1. Current best practices: Here we go through the modern technology usually employed by other mining packages. It is important to understand these in order to comprehend MiningMath differentials.

2. MiningMath uniqueness: Now that you've practiced the basics of MiningMath, and understand how other mining packages work, it's time to get deep into the theory behind the MiningMath technology.

3. Interface Overview: It's time understand our interface overview with detailed information about every screen and constraints available in MiningMath. Home page, Model tab, Scenario tab, and Viewer for a better understanding of the possibilities.

Using and validating your data

1. Formatting the Block Model: Learn how to format your block model data and use it in MiningMath.

2. Importing the block model: Go through the importation process and to proper configure your data.

3. Economic Values: MiningMath does not require pre-defined destinations ruled by an arbitrary cut-off grade. Instead, the software uses an Economic Value for each possible destination and for each block. After your data is formatted and imported into MiningMath, you may build your Economic Value for each possible destination.

4. Data Validation: Once your data is set, it's time to validate it by running MiningMath with a bigger production capacity than the expected reserves. Thus, you will get and analyze results faster.

5. Constraints Validation: Continuing the validation, start to add the first constraints related to your project so that you can understand its maximum potential.

Improve your results

1. Integrated Workflow: Each project has its own characteristics and MiningMath allows you to choose which workflow fits best in your demand and to decide which one should be used.

2. Super Best Case: In the search for the upside potential for the NPV of a given project, this setup explores the whole solution space without any other constraints but processing capacities, in a global multi-period optimization fully focused on maximizing the project's discounted cash flow.

3. Optimized Pushbacks: Identify timeframe intervals in your project, so that you can work with group periods before getting into a detailed insight. This strategy allows you to run the scenarios faster without losing flexibility or adding dilution on the optimization, which happens when you reblock.

4. Optimized Schedules: Consider your real production and explore scenarios to the most value in terms of NPV.

5. Short-term Planning: Now that you built the knowledge about your project based on the previous steps, it is time to start the integration between long and short-term planning in MiningMath. You may also optimize the short-term along with the long-term using different timeframes.

Export your results

1. Exporting Data: After running your scenarios, you can export all data. Results are automatically exported to CSV files to integrate with your preferred mining package.

In-Depth MiningMath

This tutorial provides a detailed guidance to the pages in the knowledge base for new MiningMath users. A shorter tutorial can be found here with a set of must read articles. In this tutorial, a larger number of pages is contextualized and recommended for those with no previous experience using MiningMath but who wish to gain a more advanced knowledge.

Software requirements

1. Quick check: Verify if your computer has all the minimum/recommended requirements for running the software.

2. Put it to run: Here you'll have all the necessary instructions to install, activate and run MiningMath.

Set up the block model

The next step after installation is to understand the home page interface and import your project data. The following pages go over these in detail.

1. Home page: MiningMath automatically starts on this page. It depicts your decision trees, recent projects and model information.

2. Import your block model: import your csv data, name your project, set fields and validation.

3. Modify the block model: this window aims to help you to modify your block model accordingly with what is required for your project and also allows you to "Export" the block model to the CSV format to be used with any other software.

4. Calculator: calculate and create new fields by manipulating your project inside MiningMath.

Handling unformatted data

If you don't have a block model ready to be imported you might want to create a new one. The following pages can guide you through this process.

1. Formating the block model
2. Economic values
3. Surfaces
4. Surface validation

Define the scenario and run

Once you have started your block model defined, there are several options to set up your project's parameters before running a scenario.

1. Scenario tab: set densities, economic parameters, slope angles, stockpiles, add/remove processes and dumps, production inputs, geometric inputs and so on.
2. Save as: save the scenario's configuration once it has been configured.
3. Run: the Run tab is the last step before running your project's optimization. Change the scenario name, set a time limit, and set up results files.

Results

After running your scenario it is important to analyze and understand the given results.

1. Output files and 3D viewer: by default, MiningMath generates an Excel report summarizing the main results of the optimization. It also creates outputs of mining sequence, topography, and pit surfaces in csv format so that you can easily import them into other mining packages. The 3D viewer enables a view of your model from different angles.
2. Export model: export your model as a csv file. This can be used in new scenarios or imported in other mining packages.

Extensive set-up

MiningMath offers a lot of customization. You might use pre-defined scenarios to learn with standard parameters. Otherwise, the following pages of the knowledge base detail several important parameters that might need to be fine tuned in your project.

1. NPV calculation
2. Slope angles
3. Stockpiles
4. Destinations
5. Recovery
6. Production constraints
7. Geometric constraints
8. Widths
9. Vertical rate
10. Surfaces

11. Force mining
12. Restrict mining
13. Force+restrict mining
14. Average constraints
15. Sum constraints
16. Time limit

Advanced content

Complex projects might need advanced configurations or advanced knowledge in certain topics. The following pages cover some subjects considered advanced in our knowledge base.

1. Variable mining cost
2. Multiple destinations
3. Fixing destination
4. Fake destinations
5. Reclaim policy
6. Material types
7. Mining fronts
8. Multi-mine
9. Stochastic models
10. Percent models
11. Artificial stockpiles

Theory

In order to understand the theory behind MiningMath's algorithm, a set of pages is provided to describe mathematical formulations, pseudo-code, and any rationale to justify the software design.

1. Current best practice
2. MiningMath Uniqueness
3. Guaranteed Solutions
4. Theory Validation
5. In-depth Algorithm

Workflows

ManingMath acknowledges and supports different workflows. This knowledge base provides a set of articles aimed at showing how MiningMath can be integrated into other workflows or have its results used by different mining packages.

1. Decision trees
2. Integration
3. Integrated workflow
4. Super best case
5. Optimized pushbacks
6. Optimized schedules
7. Short-term planning
8. Multi-mine
9. Sustainable analysis
10. Uncertainties at the beginning

Translations

MiningMath supports and encourages the translation of its knowledge base to multiple languages. If you would like to translate our knowledge base and have your profile advertised please contact us.

Portuguese

Base de Conhecimento (PT-BR)

Tradução de Silvia de Castro Martins.

Rrevisão de Douglas Mazzinghy.

Traduzido em 26/11/2020 com MiningMath até versão v2.1.11.

File Type

Link

pdf

Baixe aqui

docx

Baixe aqui

Ask GPT

You can use ChatGPT to help you with our knowledge base. First, you will need to have the Plugins options enabled on GPT-4.

After that, choose the AskYourPDF option:

Finally, you should enter the following prompt:

For the requests all along this chat, consider the following content: <https://miningmath.com/wp-content/uploads/2023/05/Knowledge-Base-MiningMath-Text.pdf>

Essential Topics

Articles

- * How to Run a Scenario
- * Results of the Optimization
- * Play with Predefined Scenarios
- * The Calculator
- * NPV Calculation
- * Evaluate Project Potential
- * Decision Trees
- * FAQ

How to Run a Scenario

Video 1: Downloading MiningMath.

Reach out to our knowledge base home page if you wish for a quick introduction to MiningMath technology prior to a hands-on approach.

On MiningMath's interface, you will find the Marvin block model and its scenarios (Figure 1). It is possible to preview the scenario and its parameters before opening it (Figure 2).

Choose and open Base Case, click the "Overview" tab (Figure 3) to check the parameters, and then click on "Run" to run the optimization (Figure 4).

After that, a short report with the results will be generated. To view it, check all the boxes on the "Load Options" window and click on "Load" (Figure 5).

Finally, whenever you feel ready to run your own scenarios, start by formatting your data here.

Figure 5: Loading the project

Figure 1: Marvin's scenarios

Figure 2: Open a scenario

Figure 3: Check the project's overview tab

Figure 4: Click on the run button

Figure 5: Loading the project

Figure 1: Marvin's scenarios

Previous image

Next image

Results of the Optimization

By default, MiningMath generates an Excel report summarizing the main results of the optimization. It also creates outputs of mining sequence, topography, and pit surfaces in .csv format so that you can easily import them into other mining packages.

Viewer

The 3D viewer enables a view of your model from different angles. The block colors are defined accordingly with each property displayed, varying from blue to red (smallest to largest), due to destinations, periods, or any other parameter. Therefore, it's possible to filter the blocks by the period in which they were mined or processed, for instance. In addition, it also allows you to compare multiple scenarios by loading different cases and using the left bar to change from one to another.

Figure 17: Surface 9

Figure 1: Mine Periods 1-9

Figure 2: Load executions

Figure 3: Mine Period 1

Figure 4: Mine Periods 1-2

Figure 5: Mine Periods 1-5

Figure 6: Mine Period 4

Figure 7: Mine Periods 8-9

Figure 8: Processing Periods 1-10

Figure 9: Processing Periods 3-9

Figure 10: Processing Periods 9-10

Figure 11: Destinations

Figure 12: Destination Process 1

Figure 13: Destination Waste

Figure 14: Surface 1

Figure 15: Surface 3

Figure 16: Surface 7

Figure 17: Surface 9

Figure 1: Mine Periods 1-9

[Previous image](#)

[Next image](#)

Output Files

After optimizing your block model and running your scenario(s), MiningMath generates standard output files with detailed reports. The main files have a universal format (.csv), which allows you to easily import them onto other mining packages to start your mine design and further steps of your projects.

To open the project folder, click on the scenario's name with the right button of your mouse and choose "Show in the Explorer". The optimization's main output files are:

- * Scenarioname.xlsx: Short report with the main results.
- * MinedBlocks.csv: Detailed report which presents all the blocks that have been mined.
- * Surface.csv: Grid of points generated through the pit each period.

Scenarioname.xlsx

Provides you with a short report with the main results of the optimization: several charts and sheets in which you can analyze the production on each period, the stockpiles by periods, the average grade of processes and dump, NPV per period, the cumulative NPV (Net Present Value), etc.

Figure 18: Graphic results

MinedBlocks.csv

This file offers a detailed report on all the mined blocks and their specificities: information on the mining sequence based on each block extracted, along with mined and processed periods, destinations, economic value, and all information used for the optimization. This file also allows you to identify blocks that were stocked and the algorithm decision-making process.

Figure 19: Mined blocks

Surface.csv

The surface scenario brings a grid of points generated through the pit of each period: each surface is named according to its mining periods and contains information about the topographic coordinates at that time. These files can be imported into the viewer separately, so that you can verify and validate your data before starting the optimization process. Note: The surfaces are exported/imported from/at MiningMath in Coordinates.

Figure 20: Surface's CSV

Video 1: Outputs and files hierarchy.

[Play with Predefined Scenarios](#)

MiningMath allows you to learn with each scenario by providing standard parameters which simulate some common constraints a mining company may face. Standard scenarios are listed and described below so you can identify the main changes made within the "Overview" tab.

The ultimate goal of this practice is to prepare you to build Decision Trees, which allow you to organize scenarios in order to understand how variables influence one another and, consequently, how these variables determine the final NPV.

Figure 1: Scenarios on the Home Page

Dataset

The examples in this page come preinstalled with every version of MiningMath. If you have deleted this project by any chance, please download the zip file below, extract the files and choose the "Open Project" option in MiningMath.

* Marvin_Strategy_Optimization.zip (demo project)

BaseCase

The Base Case consists of the initial scenario, with a uniform production capacity, and without sum, average, or surface mining limits.

Figure 2: BaseCase overview

BaseCase-RampUp

While the base case considers a uniform production capacity, the BaseCase-RampUp scenario offers the possibility to vary the levels of production within the various timespans. We have an initial production capacity of 10Mton on the first 2 periods; 20 Mton on periods 3 and 4; and 30 Mton from period 5 until the end of the mine's lifetime, with a total movement constraint of 30, 60, and 80 Mton, considering the increase of production within the time-frames mentioned.

Figure 3: BaseCase-RampUp

PriceUp and PriceDown

Scenarios "PriceUp" and "PriceDown" differ in relation to the basic scenario in the economic value used for the calculation of the P1 process, where there is an increase and a decrease of 10% at the copper selling price, respectively. In the destination tab, "P1 Cu +10" and "P1 Cu -10" were the values used for the process.

Figure 5: PriceDown

Figure 4: PriceUp

Figure 5: PriceDown

Figure 4: PriceUp

Previous image

Next image

PriceUp-RampUp and PriceDown-RampUp

These scenarios consider a 10% copper selling price increase and decrease, and a ramp-up of the production capacity at the same time, as mentioned before.

Figure 7: PriceDown-RampUp

Figure 6: PriceUp-RampUp

Figure 7: PriceDown-RampUp

Figure 6: PriceUp-RampUp

[Previous image](#)

[Next image](#)

PriceUp-RampUp-Protection300 and PriceUp-RampUp-Protection400

This scenario considers a 10% copper selling price increase and a ramp-up of the products at the same time, as mentioned in the previous scenario. In addition, a restrict mining surface (this constraint is used to prohibit access to this area in a specific timeframe) was included up to the fourth period, since it may represent some legal constraints on a project.

Figure 8: PriceUp-RumpUp-Protections

Below you can see a description of some scenarios of other Decision Trees.

MW150

The MW150 scenario considers different geometries from the base case at the geometric constraints. In this scenario, 150 meters was used as mining width (the horizontal distance between the walls of two surfaces that belonged to consecutive periods), and a vertical rate advance of 180 meters.

Figure 9: MW150

AvgCu

In the AvgCu scenario, blending constraints were added in the average tab to consider allowed at the process plant 0.5% as a minimum and 0.7% as a maximum average copper grade. The optimization will have to fulfill the P1 process capacity and, as an additional challenge, it has to meet this new set of parameters related to the Cu average content within the ore.

Figure 10: AvgCu

AvgCu-Stock5Mt

Here, the same blending constraints of the previous scenario (AvgCu) were added, in addition to a stockpile limit of 5Mton for process 1, on the destination tab. This feature allows you to control the stock limit of your whole process, which increases the optimization flexibility to feed the plant, while respecting the blending constraints that were already implemented.

Figure 11: AvgCU-Stock5Mt

Proc13000h and Proc13000h-33Mt

Scenario Proc13000h considers 13.000 hours of processing equipment used as the maximum limit. This constraint was inserted at the sum tab and it controls variables such as rock type feeding, energy

consumption, and any parameter controlled by its sum. Scenario Proc13000h-33Mt considers an increase of 10% in the production, inserted at the production tab beyond the parameters mentioned previously.

Figure 12: Processing hours

The Calculator

This feature allows the user to manipulate their project inside MiningMath, enabling adjustments and new field creation. Figure 1 shows a general view of the calculator. On the left side we have the block parameters and on the right the calculator itself, where the calculation can be done.

Figure 1: Calculator.

To manipulate the calculator just insert a name for the new field, select the type of field (to know more about field types, access this link), and build your expression. In case of a more complex expression, just mark the field "Logical Test" to enable conditional features. To manipulate the calculator just insert a name for the new field, select the type of field (to know more about field types, access this link), and build your expression. In case of a more complex expression, just mark the field "Logical Test" to enable conditional features.

* + : Addition

* - : Subtraction

* * : Multiplication

* / : Division

* % : Modulus

* ** : Exponential

* // : Floor division

Practical approach

To facilitate the understanding, let's work on some examples. You can see below a generic math expression (left), and its equivalent written on MiningMath's calculator (right)

$$(x2) \times (y2 - 1) \leftrightarrow x^{**2} * ((y/2) - 1)$$

Adding a field without logical expression

Using an example of the Marvin's Economic Value calculation, we are going to add a Block Tonnes field, as the figure 2:

Figure 2: Adding a new field

Adding a field with a logical expression

One more time using Marvin's block model, let's suppose we want a maximum slope angle of 45 degrees.

First, we name our Field, in this case, will be "SlopeMax45d", select the field type as "Slope" and check the Logical Test box. Then a double click on the Slope field select it and already put it on the Expression. The next step is to select the operator, as we want a maximum of 45 degrees, we choose

the operator ">" and insert the value 45 in the text box. If the value is true, that is, if this value is bigger than 45 it will now have the value of 45 assigned to it. If the value is false, i.e., lower than 45, then it will keep its value. Figure 3 shows this calculation:

Figure 3: Logical test expression

During the expression construction, green or red lines will underline it, highlighting the correct parts and the ones that need adjustments to become correct. When it is all set, just click on "Add field" and this new field will be available for use on the project on its correct field type assignments. In case the user needs to delete a field, just go to the parameters option, select and delete it.

NPV Calculation

MiningMath's internal math for NPV Calculation

The following video explains more about the NPV calculation made by MiningMath's algorithm. The understand of these steps might be useful for users working on projects with variable mining costs, which are not yet smoothly implemented on the UI.

Video 1: NPV calculation.

Discount rate per period

The discount rate (%/year) is provided by the user in MiningMath's interface, as depicted in the figure below.

Figure 1: Interface example to define discount rate (%/year)

Discounted cash flow for annual time frames

In a usual scenario period ranges are defined by annual time frames, as depicted in Fig-2.

Figure 2: Interface example with annual time frame

In this case, the annual discount rate multiplier (annual_multiplier) to return the discounted cash flow is performed as follows:

$$\text{annual_multiplier}(t) = 1(1 + \text{input discount rate})^t$$

The table below exemplifies one case for 10 periods.

Period
Process 1
Dump 1
NPV (Discounted) M\$
Annual multiplier
Undiscounted NPV M\$
1
P1
Waste
1.2
0.909
1.320
2
P1 +5%

Waste
137.9
0.826
166.859
3
P1 +5%
Waste
132.5
0.751
176.358
4
P1 +5%
Waste
105.4
0.683
154.316
5
P1 -5%
Waste
89
0.621
143.335
6
P1 -5%
Waste
92
0.564
162.984
7
P1 -5%
Waste
91.3
0.513
177.918
8
P1 -10%
Waste
52.3
0.467
112.110
9
P1 -10%
Waste
54.3
0.424
128.037
10
P1 -10%

Waste

12.1

0.386

31.384

Table 1: Example of annual multiplying factors and undiscounted cash-flows for a 10% discount rate per year. Process 1 exemplifies the use of different economic values per period.

In details, Table 1 lists:

- * the NPV (discounted) resulting from a 10 yearly period with a 10% discount rate per year.
- * the annual discount rate (annual_multiplier) for each period; and
- * the undiscounted NPV as the result of the discounted NPV divided by the annual_multiplier.

Discounted cash flow for custom time frames

MiningMath allows the creation of scenarios in which period ranges are defined with custom time frames (months, trienniums, decades, etc.), as depicted in Figure 3.

Figure 3: Interface example with custom time frames

In this case, the discount rate is still provided in years on the interface. However, the discount rate per period follows a different set of calculations.

In the example on Figure 3, the first range is composed of 12 periods of one month each. To calculate their discount rate, a timeframe factor for each period t is defined first. This is basically the value associated with each timeframe, as depicted in Figure 3, or a custom value given by the user. Another required information is the year in which t occurs. For instance, the 2nd month occurs in year 1, while the 13th month occurs in year 2.

$TF(t) = \begin{cases} 1 & \text{if } t \text{ is in years} \\ 1/12 & \text{if } t \text{ is in months} \\ 1/3 & \text{if } t \text{ is in trienniums} \end{cases}$

$year_of(t) = \text{year in which } t \text{ occurs}$

The discount rate of periods in which timeframes correspond to less than one year (low_multiplier) is, then, given by the formula:

$low_multiplier(t) = annual_multiplier(year_of(t)^x)$

where $x = \sum_{i=1}^n TF(i)$

Table 2 shows an example of 12 periods of one month each, with:

- * timeframe factor;
- * cumulative timeframe factors;
- * corresponding annual_multiplier for each period;
- * the low_multiplier of each period; and finally,
- * the undiscounted cash flow (discounted cashflow divided by the low_multiplier)

Period

Type

Process 1

Dump 1
NPV (Discounted) M\$
Time frame
Years rounded
Cumulative factor
annual multiplier
low multiplier
Undiscounted NPV M\$

1

Month

P1

Waste

10.8

0.0833

1

0

0.90909

0.99209

10.88612

2

Month

P1

Waste

13.6

0.0833

1

0.08333

0.90909

0.98424

13.81776

3

Month

P1

Waste

11.4

0.0833

1

0.16666

0.90909

0.97645

11.67490

4

Month

P1 +5%

Waste

12.9

0.833

1

0.25
0.90909
0.96873
13.31641
5
Month
P1 +5%
Waste
11.2
0.0833
1
0.33333
0.90909
0.96107
11.65373
6
Month
P1 +5%
Waste
13
0.0833
1
0.41666
0.90909
0.95346
13.63452
7
Month
P1 -5%
Waste
11.2
0.0833
1
0.5
0.90909
0.94592
11.84033
8
Month
P1 -5%
Waste
14.4
0.0833
1
0.58333
0.90909
0.93844
15.34467

9
 Month
 P1 -5%
 Waste
 11.8
 0.0833
 1
 0.66666
 0.90909
 0.93101
 12.67437
 10
 Month
 P1 -10%
 Waste
 14.7
 0.0833
 1
 0.75
 0.90909
 0.92365
 15.91517
 11
 Month
 P1 -10%
 Waste
 13.6
 0.0833
 1
 0.83333
 0.90909
 0.91634
 14.84165
 12
 Month
 P1 -10%
 Waste
 12
 0.0833
 1
 0.91666
 0.90909
 0.90909
 13.20000

Table 2: Example of discount rates calculated per month and undiscounted cash flows with a 10% discount rate per year. Process 1 exemplifies the use of different economic values per period.

Timeframes greater than a year

In the example on Figure 3, the last period ranges are composed of 3 trienniums. To calculate the discount rate of a triennium, the annual_multiplier of each year contained in it is computed and then averaged. Hence, the discount rate of periods in which timeframes are greater than one year (great_multiplifier) is given by the formula:

$$\text{great_multiplier}(t) = \frac{\sum_{y=t}^{t+\text{annual_multiplier}(y)} t^{-t+1}}{\text{annual_multiplier}(y)}$$

where t is the first year in period and t is the last year in period

Table 3 shows the example of 3 trienniums occurring after 14 periods (12 months and 2 years). The annual_multiplier of each year contained in each triennium is listed. The average of these factors return the great_multiplifier of each triennium, which, in turn, divides the discounted cash flow to calculate the undiscounted one.

Period
Type
Process 1
Dump 1
NPV (Discounted) M\$
Time frame
great multiplier
Undiscounted NPV M\$
Year rounded
annual multiplier
15
Triennium
P1
Waste
378.4
3
0.62280
607.57584
4
0.68301

5
0.62092

6
0.56447
16
Triennium
P1 -5%
Waste
120.8
3
0.46792
258.16321
7
0.51318

8
0.46650

9
0.42409
17
Triennium
P1 -10%
Waste
4.2
3
0.35155
11.94688
10
0.38554

11

0.35049

12

0.31863

Table 3: Example of undiscounted cash flows and discount rates calculated per triennium with annual discount rate of 10%. Process 1 exemplifies the use of different economic values per period.

Evaluate Project Potential

Certain constraints related to your project can be defined so that you can understand its maximum potential. The surface generated in this case could also be used as a restrict mining in the last period to reduce the complexity of your block model and the runtime of MiningMath, since it includes a set of constraints inputted.

Example

- * Set up a scenario with 1,000 Mt in the processing plants, which corresponds to a lot more mass than expected in the whole life of the mine.
- * Add the Minimum Bottom Width (100m). This constraint will allow you to have a suitable work-front for your equipment.
- * Restrict Mining surface, if you have this constraint in your project.
- * Grade constraint until 0.7%.
- * Timeframe: Years (1), since it would all be processed in 1 period.

Figure 3: Surface

Figure 1: Sequencing

Figure 2: Restrict surface

Figure 3: Surface

Figure 1: Sequencing

[Previous image](#)

[Next image](#)

Note: Sum constraints can restrict the total amount of handling material (ore + waste) of the mine. Therefore, do not use them in the validation.

Figure 6: Average constraints

Figure 4: Production constraints

Figure 5: Geometric constraints

Figure 6: Average constraints

Figure 4: Production constraints

[Previous image](#)

[Next image](#)

[Using results](#)

Now that Constraints Validation step is done, you are able to use this final surface as a guide for future optimizations. This approach reduces the runtime and the complexity of the algorithm because when you take it into account, the blocks below this final optimized surface won't be considered and the heuristics inside the interface would be facilitated. Notice that we did not make any change in the discount rate, thus, this first NPV does not represent the reality. If you need an accurate result in this step, make sure to adjust it.

It's important to remember that when we restrict the mining into this surface, the number of periods generated in future runs could be reduced because the average parameters of each one will have to meet the constraints of the overall package. Therefore, to achieve the same parameters in a lower timeframe, some blocks may be discarded due to the mining sequence and the optimization of destinations inside the whole mass.

Having this idea in mind, you should already have enough information to decide and structure the next step of the optimization. Based on the amount mined in the last item and in the processing capacity, define a good timeframe to identify the mining sequence. In this case, we had 231 Mt as the ore total mass to split in almost 23 years, since the processing capacity is 10Mt.

To improve efficiency in the optimization, before working on a yearly basis, we made the decision to consider the first 5 years. It is reasonable to generate a 10-year surface to consider the optimization inside this limit due to observations made before. Remember that each assumption here can be done accordingly with your project's demands and that MiningMath can work with any timeframe to meet your needs.

Decision Trees

Comparing Scenarios

Decision Trees provide you with a detailed broad view of your project, allowing you to plan your mining sequence by analyzing every possibility in light of constraints applied to each scenario, which options are more viable and profitable to the global project, as well as how these factors impact the final NPV. Consider, for instance, the plant production per year as a variable factor. Using Decision Trees (Figure 1), you will be able to analyze how each constraint, e.g. the ore price, affects that year's production and benefits or not the global project.

Figure 1: Essence of a Decision Tree, done in presentation software.

By running all the scenarios individually, just like what you did on Practice First, you will be able to identify how each change, within a set of constraints, impacts the NPV results and the mining sequence generated (Figure 2 and 3), which provides you a broader view of your project and enables you to decide which route you should take to generate value to your company.

Figure 3: PriceUp

Figure 2: BaseCase

Figure 3: PriceUp

Figure 2: BaseCase

[Previous image](#)

[Next image](#)

How to Analyze Multiple Scenarios

Increase in the value of copper

Analyzing first the scenario in which there is a change in the economic value of the P1 process ("scn-PriceUp"), values such as NPV would naturally be different. In this case, analyzing the NPV and the total movement (Figure 3), it's possible to understand that a different mining sequence was generated, which increased the mine's lifetime by one period. This market change has also increased cumulative NPV (Figure 4) values based on its direct relation with the copper selling price. The charts below were made with the help of MiningMath's results in simple spreadsheet software.

Figure 4: Total mass (Process+waste) handled on each scenario.

Figure 5: Cumulative NPV contrasts.

Adding an average grade limit

Now we can analyze the scenario in which a restriction in the average grade at P1 process was added, using a minimum and a maximum limit of copper ("scn41-AvgCu"). The blocks that would be processed would have to meet established targets, allowing a better selectivity of what should be processed or not. The ones which have higher or lower grades than required could be blended with others to generate an average grade that respects the constraints and improves the NPV.

Notice that there was a higher total production (Figure 5) in each period, caused by the increase of the stripping (ore/waste) ratio to meet the 30 Mtons of ore production at P1 Process and the average grade targets settled at the "scn41-AvgCu" scenario. A better stock pilling use is expected, in order to use all the blending capabilities and decision-making intelligence of the algorithm to decide which blocks could be mixed to fulfill the plant capacity. In addition, the cumulative NPV (Figure 6) shows that by inserting average grade constraints we consequently reduce the algorithm flexibility and lose some money to keep the operational stability frequently required at a processing plant.

In general, the main goal of MiningMath, considering the set of constraints provided, is to maximize the cumulative NPV in the shortest mine lifetime possible, which would reduce the project depreciation by interest rates. The charts below were made based on MiningMath results with the support of spreadsheet software.

Figure 6: Total mass handled on each scenario.

Figure 7: Cumulative NPV contrasts.

Building Decision Trees

You have been introduced to some of MiningMath's functionalities. Now let's take a closer look at how decision trees are built.

Mine project evaluation largely relies on technology from the 1960's, in which a step-wise process is usually necessary along with time-consuming activities, like pit-design, in order to create only one single scenario. Evaluating projects through this approach could take from weeks to months of multidisciplinary work just to produce a couple of scenarios. This process is often guided by some arbitrary decisions that may constrain the mathematical solution space, confining solutions to engineering expertise and judgment.

A global optimization scheduling can speed up the process of generating multiple scenarios for project overview prior to detailed work. MiningMath integrates the business' areas and allows managers to improve their decision-making process by structuring their strategic analysis through multiple decision trees with a broader and optimized view of their projects, comprising constraints from different areas of the company.

The following video shows a few possibilities recognized only when seeing the available paths to create value. The video is oriented to technical daily usage but also covers interesting subjects for the managerial perspective. For the last case, skip straight to minute 15:23.

Video 1: Video detailing the building of decision-trees.

Apply to your projects

Now that you have played with the sample data, it is time for a hands on approach and apply this optimized strategy to your own projects!

MiningMath already allows you to structure your Decision Trees layout at its home page, which facilitates and guides the decision-making and mining planning processes.

Take advantage of the possibility to add (+), rename, or delete Decision Trees (Figure 7), by clicking with the right button at their names and/or exchange scenarios (Figure 8) between trees to build different mining planning strategies. The icon is a shortcut, so you can easily open your scenario's full report.

Compare everything in a single look and identify how each change impacts your results to build your own analysis by using presentations based on MiningMath charts as shown in Figure 1.

Figure 9: Decision Trees' organization

Figure 8: Decision Trees' table

Figure 9: Decision Trees' organization

Figure 8: Decision Trees' table

[Previous image](#)

[Next image](#)

[FAQ](#)

What is MiningMath?

It is a software application that uses innovative technology for direct block scheduling. The MiningMath aims to maximize the Net Present Value (NPV) of a project deciding, based on an imported block model, which blocks will be mined, when and what the destination of each block is.

It is possible to define multiple processing plants, stockpiles and waste dumps, respecting their capacities. It is also possible to set physical limits or force mining in certain regions by importing surfaces.

As the software has a flexible algorithm, it will be possible to include other restriction types in the future, such as blending for example.

[Read more.](#)

OK, but why should I use MiningMath?

MiningMath allows running a complete schedule directly from the resource block model, with no need to define a final pit, nested pits, pushbacks, cut-off grade optimization and stockpiles as would be required as part of a traditional full scheduling exercise. MiningMath will find a mining schedule that aims to maximize the NPV of the project, combining all the steps mentioned and optimizing all the periods simultaneously. Therefore, an experienced professional can test multiple scenarios by modifying parameters and advance other stages of his work, while MiningMath performs the entire optimization.

[Read more.](#)

Is the generated solution operational, or only a mathematical result?

Each mining plan generated by the optimization respects important geometrical parameters, such as a minimum pit bottom width, a mining width and vertical rates of advance, which can be configured specifically for your project. Furthermore, MiningMath technology generates surfaces without geotechnical errors. The generated plans are close to the operational reality of the mine, which implies smaller variations in the parameters when ramps are designed.

In practice, how can I use MiningMath to scheduling optimization?

A block model, in CSV format, including indexes or coordinates and economic values of the blocks is first imported followed by entering the primary parameters of the model and production restrictions via interface. Upon completion of these steps, MiningMath is ready to perform the optimization.

The resulting surfaces will respect the user-defined parameters and a report will be generated with graphics containing the most important indicators.

To demonstrate the usage and power of the software, you can access our demo video [here](#).

Does MiningMath suit my project? What are the limitations of the current version?

The current version of MiningMath suits any open pit mining project that can be modeled with blocks of regular dimensions. If your project has multiple types of rock per block, there are ways to adapt the inputs to handle these cases. MiningMath is 100% based on 64-bit technology and has an efficient algorithm, capable of handling tens of millions of blocks without requiring supercomputers or cloud computing.

To date, MiningMath has focused on developing the best algorithms; future versions will introduce more facilities for the user, including the development of plug-ins to mining softwares on the market.

I have a sub-blocked model. How could MiningMath be used in this case?

If the model can be exported by dividing all blocks into sub-blocks, then we have a regular database formed only by sub-blocks and MiningMath can run it. We perform regular tests successfully using models with tens of millions of blocks. For future versions, we are planning significant efficiency improvements.

Does MiningMath have the Lerchs-Grossman (LG) algorithm implemented?

No. LG is a brilliant algorithm for its time, but none of the new software need to implement it anymore. The technological advances have proven that new methods overcome some barriers that the LG faces. These days, any software that has the same mathematical model of LG implemented will rather implement an algorithm based on maximum flow rate (Max Flow) which can be run tens or hundreds of times faster than the LG. However, both LG and Max Flow have no flexibility to include other important restrictions such as a minimum pit bottom width or blending.

MiningMath uses highly recommended technology currently in practice and inside research centers, being what is the most advanced, tested and available when talking about optimization. It was implemented using modern techniques based on mixed integer programming and heuristics. Its mathematical model is more realistic, for considering operational aspects and uses surfaces to return solutions that does not have any geotechnical errors. What in practice is mined are surfaces and not blocks. This type of technology has the flexibility to include other real restrictions, such as blending.

Read more.

Interface Overview

Articles

- * Home Page

- * Model Tab

- * Scenario Tab

- * Viewer

- * Overview

- * Save As

- * Run

Home Page

MiningMath automatically starts on the Home Page, as shown in Figure 1.

Overview of Recent Projects

Decision Trees feature enables you to create new tabs and ways to organize the mining planning strategies.

Open and View options for the current selected scenario under the Decision Trees tab.

Edit button shortcut, which will lead you to the Calculator functionality explored further ahead.

Model table discloses the main information regarding your block model and its parameters.

Figure 1: Interface example. Click to expand.

The following are three main information windows:

* The Recent Projects window allows you to select a project and have an overview of the saved scenarios that will appear right away at the Decision Tree tab. It allows the user to quickly navigate through recent projects and scenarios without opening them from their original folders. Clicking with the right button on the project name, the user will have 4 options:

1) New scenario: "Scenario config" window will open allowing you to decide in which decision tree to place the new scenario, its name, and a description for it. Afterward, is going to be directed to set it up in the scenario tab.

2) Show in explorer: This option takes to the directory containing the folder with the project.

3) Remove from list: Excludes the referred project from the Recent Projects list.

4) Delete project: Deletes the project and the scenarios from it.

* The Decision Trees feature enables you to create new tabs and ways to organize the mining planning strategies by exchanging scenarios between them, if necessary. It allows access to all paths involved in the project, providing a broad view and enhancing the decision-making process. It shows decision trees' scenarios with key information about them, such as name and description to easily identify its characteristics, NPV on M\$, runtime, and a direct link to the sheet containing all the results of the scenarios, which are available after the scenario's execution. It's possible to:

1) Add new trees by clicking on "+"

2) Rename a tree by double-clicking its name

3) Clicking with the right button, the user may add a new scenario, rename or delete this tree.

Getting into more details, some hidden options are available to the user when clicking with the right button on the scenario's name: open, view model, rename, show in explorer, delete, and the possibility to transfer it between decision trees. The scenario description can be easily edited with a double click.

To open any scenario, click on one Recent Project first, and then choose the one you want to work with at the Decision Trees tab by double-clicking with the right button of your mouse, and choosing the option "Open", or just selecting "Open" at the right bottom of this feature. The "View" button leads to the MiningMath's Viewer.

* The Model table discloses the main information regarding your block model and its parameters, so that you can easily review it at any time using the "Edit" button shortcut, which will lead you to the "Calculator" functionality.

Model Tab

This window aims to help you to modify your block model accordingly with what is required for your project and also allows you to "Export" the block model to the CSV format to be used with any other software.

This tab starts at the "Parameters" option, showing your previous set up at the importation, similar to Figure 2, and all the existing fields. It also allows you to remove any parameter at any time.

Figure 1: Parameters Model tab

The "Function" option discloses the "Validate your block parameters" table, so that you can choose a single block within your model to verify its values. It also enables MiningMath's internal Calculator to make adjustments and changes to your dataset with the addition of new fields.

Figure 2: Internal Calculator at Function in Model tab

Scenario Tab

General Parameters

MiningMath automatically switches to the Scenario Tab in the General option once a scenario is opened, as shown in Figure 1.

Figure 4: General tab and stockpiling

Figure 1: General tab and densities

Figure 2: General tab and economic parameters

Figure 3: General tab and slope angles

Figure 4: General tab and stockpiling

Figure 1: General tab and densities

Previous image

Next image

The General tab presents all the general inputs regarding densities (Figure 1), economic parameters (Figure 2), slope angles (Figure 3) and stockpiles (Figure 4), which are detailed next.

Densities

Densities, as shown in Figure 1, are used along with block size to calculate tonnages. The user has two options to define them:

- * "Field" shows the column(s) that has/have been assigned to the density during the importation. This option is intended to allow varying densities by block.

- * "Default value" is applicable to any block without density information, whether a density column is imported or not. It is also used when you choose the field as .

Economic Parameters

The Discount Rate is a field of Economic parameters, as seen in Figure 2. It is usually considered in an annual base, is responsible to define the impact of mining ore/waste over time, which influences the algorithm decision-making process.

While working with different time frames, the discount rate serves just a rough NPV approximation, and it doesn't affect so much the quality of the solution, given that the best materials would be allocated first. Thus, by multiplying or splitting it by the number of periods, you might get reasonable results.

Slope Angles

Slope angles are one of the most important parameters when considering constraints hierarchy. The user, as seen in Figure 3, has two options here:

- * Field shows the column(s) that has/have been assigned to the slope during the importation. This option is intended to allow different slope angles by block.

- * Default value is used to any block without slope information, even if a column was assigned. It is also used when you choose the field as <none> .

Stockpiling

The stockpiling feature can be used by activating the checkbox. When this option is enabled, shown in Figure 4, the user can define:

- * Fixed mining cost (cost/t) refers to the average mining cost used for the economic function. This value is used to decompose the economic value while considering stockpiles.

- * Rehandling cost (cost/t) represents the cost to reclaim blocks from the stockpile to the process.

To illustrate the other tabs inside the Scenario tab, the parameters used in the Marvin case will be employed as summarized below:

Parameter

Subparameter

Value

Densities

Field

Density

Default value

2.75

Slope angles

Field

Slope

Default value

45 degrees

Stockpiling

Fixed mining cost

0.9\$/t

Rehandling cost

0.2\$/t

Discount rate

-

10%

Table 1: Parameters used in the Marvin cases.

Destinations: Process, Dump and Stockpile

Figure 5: Destinations tab, Recoveries for each element/mineral and destination, and Stockpile limit in tonnages.

On the Destinations tab (Figure 5), you will define destinations to where the blocks can be sent. Each target must be mapped with their respective field containing the economic values. MiningMath requires at least one destination for the process and one destination for the dump. For each destination, you have an economic value and recoveries by elements.

Process & Dump

To add destinations, in the bottom corner of the window, click on:

- * Add Process

- * Add Dump

Each scenario must contain at least one process and one dump among the destinations imported. The destination of each block will be reported by assigning them with the numbers 1 or 2 (see the numbers beside the Name column), which depends on the order of addition.

Recovery

For each processing stream, the user must inform a process recovery, varying from 0 to 1, to any element/mineral whose column has been imported as a grade.

This value on the interface serves only for the purpose of generating reports, as it has been considered during the economic calculation. Use the following values for the processing stream:

- * Cu: 0.88

- * Au: 0.60

Stockpile

You can also define a tonnage limit for the stockpile if activated in the General tab (see Figure 4).

MiningMath considers the tonnage inputted as a cumulative upper limit that will be considered all over the life of mine.

In this example, a limit has not been defined, which implies an capacity to stock. Read more about stockpiles.

Economic Value

In the column of Economic value, you must assign each destination to its corresponding economic function. Therefore, use:

- * Destination 1 - Process 1 - Economic Value Process

- * Destination 2 - Dump 1 - Economic Value Waste

Figure 6 zooms in the destination fields, showing how they should look like for this example.

Figure 6: Economic values for each destination.

Production Inputs

Figure 7: Production tab and production limits for each destination based on an yearly timeframe.

After completing the previous fields, move to the Production tab (Figure 7). You can define limits (in tonnes) for each destination and the total amount of material moved per period and also add different timeframe ranges in the optimization. For this example, use the values as shown in Figure 7.

- * TimeFrame: Years (1)

- * Process 1: 30,000,000 t

- * Dump 1: 50,000,000 t

- * Total: 80,000,000 t

Read more about productions changing over time.

Geometric Inputs

Figure 8: Operational constraints.

On the Geometric tab (Figure 8), you can define parameters intending to find mathematical solutions that already consider basic requirements to be operationally feasible. Figure 8 highlights Operational Fields, which could differ for each period range and timeframe, such as minimum widths and vertical rate. Optional Fields are also allowed. These allow the definition of: 1) areas to be forced and/or restricted; and 2) periods to which surfaces are applied. In this example the values defined for all periods are:

- * Minimum width: Mining 100m, Bottom: 100 m

- * Vertical rate of advance: Maximum 150 m

In the Geometric tab you can also force mining and restrict mining using surfaces based on coordinates and defined as a 3D-grid of points in the CSV format. Surfaces are the most important constraints within MiningMath's hierarchy, allowing you to impose one's understanding and take control of prior results and operational aspects.

Using surfaces, you are able to play with geotechnical aspects, force certain regions to allocate waste material, restrict areas to protect the environment, and/or guide operational aspects by importing a designed pit.

Due to the complexity of this subject, surfaces will be treated in a specific section of our documentation.

Average

Figure 9: Blending constraints.

On the Average tab, you are able to define a minimum and/or maximum average grades for any element/mineral imported as grade (Figure 9 area 1).

Blending constraints can also be defined by period ranges (Figure 9 area 2) and/or destination (Figure 9 area 3).

It's worth mentioning that this minimum limit does not represent cut-off values. Since it is based on average parameters, the algorithm can use lower values to respect this parameter and increase the NPV with higher ones. If you wish to input a cut-off, a good way to do it is by filtering these blocks and assigning the the mass as a sum field, as mentioned here.

Sum

Figure 10: Other constraints.

On the Sum tab, you are able to consider any summed parameter as a minimum and/or maximum limits for any data imported as other (Figure 10 area 1).

Other constraints can also be defined by period ranges (Figure 10 area 2) and/or destination (Figure 10 area 3).

This feature is available only at the full versions of MiningMath. [Read more.](#)

Overview

Figure 13: Execution options.

Figure 11: Overview screen for final verification of inputted parameters.

Figure 12: Save as information.

Figure 13: Execution options.

Figure 11: Overview screen for final verification of inputted parameters.

Previous image

Next image

Click on Overview (Figure 11) for a single page summary of all parameters related to the direct block scheduling, as illustrated in the figure below.

The Save as option (Figure 12) can be used to redefine the name, description for an edited scenario, and its decision tree.

You can also decide which files MiningMath will produce as outputs by using the execution options (Figure 13) before clicking on "Run" your scenario.

Viewer

After running your scenario, the Mined Blocks file will disclose its results on the 3D viewer, enabling the view of your model from different angles. By selecting "Period Mined", you are able to view the mining sequence period by period. By selecting a surface, it is possible to identify the topography changes of each period and also modify its opacity, facilitating the visualization.

MiningMath also allows you to import surfaces already created if you place them in the same folder as the other ones, so that you can validate their geometry if necessary.

Click on "Load Scenario" to import multiple scenarios and compare them, in order to extract the best results accordingly with your project constraints.

Figure 2: Viewer

Figure 1: Viewer

Figure 2: Viewer

Figure 1: Viewer

Previous image

Next image

Overview

Last check before running

The Overview tab aims to let the user to give a last check on each parameter inputted. Every information available in this screen, as shown in Figure 1, is also present on the previous screens:

1. General
2. Destinations
3. Production
4. Geometric
5. Average
6. Sum

Note the user can even go straight to the overview tab to save time.

Figure 1: The Overview Tab.

Save As

Options to save a scenario

By clicking over Save As button, a window will pop-up, as shown in Figure 1. Then, the user will be able to set:

- * A scenario name
- * A scenario description

These fields intend to better identify a scenario among several ones by just taking a look at the Projects List.

Figure 1: Saving As.

Run

Running the optimization

The Run tab is the last step before running your project's optimization. The Figure 1 below shows its interface and in sequence we will discuss each part.

Figure 1 - Run tab

Editing scenario name

At this part it is possible to check the saving parameters, such as the Decision Tree where the scenario will be inserted, scenario's name and description.

Export to CSV

Allows choosing to export the surfaces to a CSV file and/or the block model, with other options such as: all blocks mined; only mined blocks, coordinates and indices.

Time limit

It is possible to indicate a time limit in hours before running a scenario in the "Run" tab as depicted in Fig. 1. The time limit is defined in hours due to the usual complexity of mining projects and by the fact that MiningMath will always try to deliver a reasonable solution. To know more how this is done read the article [here](#).

Reports visualization

The lower part of the screen allows you to view the results, in CSV, if the scenario has already been run, inside the software, facilitating strategic decisions.

Run button

Progress bar

Figure 2. Progress bar

Handling Data

Articles

- * Formatting the Block Model
- * Importing the Block Model
- * Economic Values
- * Data Validation
- * Constraints Validation
- * Exporting Data
- * Public Datasets

* Output files

Formatting the Block Model

Block Model Basic requirements

MiningMath requires the following formatting specifications:

- * Regularized block model: This means all blocks must be the same size.
- * Air blocks must be removed prior to importation. This is the way MiningMath recognizes the topography.
- * Coordinates of each block in the 3 dimensions.
- * Header Names should not have special characters or have them exceed 13. Use this recommendation for folders and files also.
- * The data format should be a CSV file (Comma Separated Value), which might be compatible with most mining packages.

Good practices

- * Configure Microsoft Windows number formatting to use dot as the decimal separator.
- * Use the metric system.
- * Set multiple fields that will consider different economic values, material types, contaminant limits, and any other variable you wish to analyze or control.

Understanding Field Types

Field Types are the fields MiningMath can understand. Each column imported should be assigned to the proper field type so that the software treats each variable accordingly with its meaning.

Figure 1: Field types

Mandatory Field Types and their meanings

- * Coordinates X, Y, and Z refer to your geo-referenced information. The same applies to Index X, Index Y, and Index Z, which requires a column for each one of the 3D axes.
- * Average refers to any variable that could be controlled by means of minimums and maximums considering its average: grades, haulage distance, and other variables.
- * Economic Value refers to the columns with the economic value, which represent the available destinations. It is possible to import multiple economic values at once, and they may be used simultaneously (ex.: multiple processing streams) or calculated in the internal calculator mentioned on the next page.

Optional Field Types and their meanings

- * Density refers to the block's density. This field is used to calculate the block's tonnage.
- * Slope refers to slopes varying block-by-block, which gives the flexibility to define slopes by lithotype and sectors.
- * Recovery refers to recoveries varying block-by-block.

* Sum refers to any variable that could be controlled by means of minimums and maximums considering its sum.

* Other refers to information that you wish to have in the exported outputs.

* Skip refers to any variable that should be ignored. This field type might help improving the runtime since these variables will not be considered and exported along with the optimization outputs.

Field names shortcut

Shortcuts can be used for automatic recognition in the importation process. These are listed in the table below.

Field name

Shortcuts

Coordinates

X | Y | Z

Average

@ | grade

Density

% | dens | sg

Economic value

\$ | dest | val

Recovery

* | recov

Slope

/ | slope

Sum

+

Skip

!

Mandatory requirements

Considering the specifications mentioned before, the formatted data set should have the following information for each block:

* Coordinates or Indexes.

* Grades (at least one element assigned as Average).

* Economic values (at least 1 process and 1 waste).

The following video gives an introduction on how to setup your block model.

Video 1: Block Model setup.

Attention to software conversions

The model's origin must be placed at the bottom portion, starting to count from the minimum coordinates at X, Y, and Z.

Figure 1 highlights a block model origin at the corner of the first block and the coordinates on its centroid.

Each software uses its own conventions for data format, naming and numbering systems, etc. These differences should be observed to prevent conflicts when transiting data from multiple software, each one for one specificity.

What you must know:

- * MiningMath uses coordinates (X,Y,Z) for which Z, which represents the elevation, starts upwards (Figure 3a).

- * Other mining software may use indexes with IZ starting downwards (Figure 3b). MineSight is an example that uses this notation.

Figure 2: Blocks Matrix.

There is no right or wrong convention, but there is a correct procedure for each software.

To invert coordinates use the following formula to convert:

$$\text{new}(Z) = \text{max}(Z) + 1 - \text{current}(Z)$$

Figure 3a: The lowest IZ value is at the bottom of the model.

Figure 3b: The lowest Z value is at the top of the model, which will not fit MiningMath requirements.

Air Blocks

MiningMath recognizes that all imported blocks of your model are underground. This means it is necessary to remove all the air blocks prior to importation. Unless your topography is totally flat, which is unlikely, the image below shows an example of your model should be displayed.

The non-removal of air blocks may lead to unsatisfactory results and long processing times, since it would be considering blocks that do not exist in reality.

Figure 4: Example of how block models should look like with a rectangular base.

More Details on Air Blocks

The following video shows how to do remove air blocks using filters on MS Excel. These tips are also applicable to any mining software of your choice.

Video 1: Removing air blocks using filters on MS Excel.

Importing the Block Model

Block Model File

To import the block model, select the option New Project on the left panel of MiningMath (Figure 1).

Figure 1: Creating a new project to import a new model.

Afterwards, the file name input field is shown in red, indicating a mandatory field. (Figure 2) Browse for and select the CSV formatted file. Press Next to advance.

Figure 2: Importing a CSV model.

Project Naming

In the next window, shown in Figure 3, the Model Name must be entered.

Optionally, the destination folder (Model Folder) can be changed as well as the Scenario Name, and a Scenario Description can be added.

Figure 3: Defining a name for the model and the first scenario.

Decimal separator

MiningMath does not recognize CSV's using semicolon (;) as a decimal separator. To change it on Windows, go to:

'Control Panel' > 'Clock, Language, and Region' > 'Region' > 'Change date, time, or number formats' > 'Additional Settings'

Next, change the List Separator to the comma sign (,). Press OK and OK. Finally, you will need to change the separator on the CSV. On Excel, you will need to open it, convert text to columns, and then save it again.

Imported Fields & Validation

Upon clicking Next, the following window will provide a statistical summary of information for the block model that will be imported (Figure 4).

Check the parameters carefully.

Figure 4: Validating your data.

Geo-reference system, Origin, Dimension and Rotation

Upon clicking Next, the CSV file will be imported into MiningMath, and show data related to the block model geo-reference system, that can be coordinates or index. The next steps are to place the rotation degrees (Azimuth rotation), origin accordingly with your mining package, and the block dimension as illustrated in Figure 5. The number of blocks is automatically calculated after the origin and dimensions are provided.

The origin of this project was $x=3,475$, $y=6,480$, and $z=285$, and the block dimensions were 30 meters in each coordinate.

Figure 4: Coordinates input.

Rotated models

MiningMath supports the use of block models that have been rotated using an Azimuth rotation (Figure 5). The amount of rotation degrees can be passed as depicted in Figure 6. After importing, you can see the rotated model in the Viewer tab (Figure 7).

Figure 5: Example of Azimuth rotation in the coordinate system.

Figure 6: Azimuth rotation depicted when hovering over the RZ field.

Figure 7: Example of rotated model in the viewer tab.

Field Type Assignment

When Next is selected, the following form will appear (Figure 8), showing correlations between the imported CSV file header and the available field types in MiningMath.

You must associate each imported column to one of the options located just above the table, for instance: block coordinates X, Y, and Z to Coord. X, Y, and Z field types. For more details on how you can correlate each column, access this link. You can also keep the original data from your previous Mining Package, by using this approach.

If you do not already have an Economic Value function, when importing your block model, you will be directed to the Scenario tab. Then, click on the Function tab to calculate your Economic Value function in the internal calculator as explained here.

Figure 8: Assigning each column to the proper field type.

Notes

- * MiningMath has mandatory variables (columns) to be assigned to the proper Field Type:

- 1) Indexes (IX, IY, IZ) or coordinates (X, Y, Z).

- 2) Average

- 3) Economic Values (at least two)

- * Validating data screen might be overlooked, but it is very important to validate one's data based on minimums and maximums. Read more.

- * Each column imported should be assigned to the proper field type in order for MiningMath to treat each variable accordingly. Read more.

- * Typically, MiningMath recognizes some columns automatically when their headers are similar to the Field Type name. Otherwise, the MiningMath will automatically assign them to the Field Type sum.

To enable the Next button, the user needs to assign each one of the mandatory variables to their respective Field Type. Figure 9 shows, in green, a Field Type correctly assigned for the column IY. In red, the grade columns are assigned to the Field Type Other instead of Average. Consequently, the Next button is disabled.

Figure 9: Variables and Field Types.

Grade, Dimension and Origin

After clicking Next, it will demand grade units. As you can see in Figure 10, the copper grade has been defined as a percentage (%), while gold grade was defined as PPM, which stands for parts per million and, in turn, is equivalent to g/ton.

Figure 10: Informing block dimensions, origin, and grade units.

View Your Model and Surfaces

After filling in the required fields, the options View Model and Scenarios will be enabled. Before setting up your first scenario you can view it by clicking in the Viewer and Load scenario. Select all the tooltip options and click in load. This option also allows you to view surfaces created, just place them in the scenario folder before loading and do the first validation.

Warning message

A warning message can appear as no scenario was run before. Just press ok, check your block model and proceed to set up your project.

Evaluate your model

After importing your model, you can view it in the Viewer tab as depicted in Fig. 11-14. This should help you answer questions such as:

- * Where are the high grades distributed?
- * Does the process economic values, above zero, match with the regions identified in the last question?
- * How are waste economic values distributed? Are maximum and minimum values reasonable when you compare them with the process?

Figure 14: Checking your model.

Figure 11: Checking your model.

Figure 12: Checking your model.

Figure 13: Checking your model.

Figure 14: Checking your model.

Figure 11: Checking your model.

Previous image

Next image

Economic Values

MiningMath does not require pre-defined destinations ruled by an arbitrary cut-off grade. Instead, the software uses an Economic Value for each possible destination and for each block. The average grade that delineates whether blocks are classified as ore or waste will be a dynamic consequence of the optimization process.

Destinations required

MiningMath requires two mandatory destinations at least:

Therefore, each block must be associated with:

- * 1 Economic value for the processing plant
- * 1 Economic value for the waste dump
- * 1 Processing stream
- * 1 Waste dump

Notes:

* Even blocks of waste might have processing costs in the economic values of the plant. Therefore, non-profitable blocks would have higher costs when sent to process instead of waste.

* If you have any material that should be forbidden in the plant, you can use economic values to reduce the complexity and runtime, as mentioned here.

Animation (click to see): Simplified flow-chart of blocks' destinations optimization.

Calculation

Each field related to Economic Value (Process/Waste) must report the value of each block as a function of its destination (Process or Waste in this example), grades, recovery, mining cost, haul costs, treatment costs, blasting costs, selling price, etc. The user is not required to pre-set the destination, as the software will determine the best option during the optimization.

To calculate the Economic Values you can use MiningMaths's internal calculator, available at the "Function" option inside the "Model" tab. To illustrate the calculation of economic values, an example is shown below. The calculation parameters are listed in Table 1.

Description

Cu (%)

Au (PPM)

Recovery

0.88

0.6

Selling price (Cu: \$/t, Au: \$/gram)

2000

12

Selling cost (Cu: \$/t, Au: \$/gram)

720

0.2

Processing cost (\$/t)

4

Mining cost (\$/t)

0.9

Discount rate (%)

10

Dimensions of the blocks in X, Y, Z (m)

30, 30, 30

Table 1: Parameters for calculating the economic values.

Figure 1: Internal Calculator.

Block Tonnes

$$* \text{Block Tonnes} = \text{BlockVolume} * \text{BlockDensity}$$

$$* \text{Block Tonnes} = 30 * 30 * 30 * [\text{Density}]$$

Figure 2: Block model calculations.

Tonnes Cu

$$* \text{Tonnes Cu} = \text{Block Tonnes} * (\text{Grade Cu}/100)$$

$$* \text{Tonnes Cu} = [\text{BlockTonnes}] * ([\text{CU}]/100)$$

Figure 3: Block model calculations.

Mass Au

$$* \text{Mass Au} = \text{Block Tonnes} * \text{Grade Au}$$

$$* \text{Mass Au} = [\text{BlockTonnes}] * [\text{AU}]$$

Figure 4: Block model calculations.

Economic Value Process

$$* \text{Economic Value Process} =$$

$$[\text{Tonnes Cu} * \text{Recovery Cu} * (\text{Selling Price Cu} - \text{Selling Cost Cu})] +$$

$$[\text{Mass Au} * \text{Recovery Au} * (\text{Selling Price Au} - \text{Selling Cost Au})] -$$

$$[\text{Block Tonnes} * (\text{Processing Cost} + \text{Mining Cost})]$$

$$* \text{Economic Value Process} = ([\text{TonnesCu}] * 0.88 * (2000 - 720)) + ([\text{MassAu}] * 0.60 * (12 - 0.2)) - ([\text{BlockTonnes}] * (4.00 + 0.90))$$

Figure 5: Process Economic Value calculation.

Economic Value Waste

$$* \text{Economic Value Waste} = -\text{Block Tonnes} * \text{Mining Cost}$$

$$* \text{Economic Value Waste} = -[\text{BlockTonnes}] * 0.9$$

Figure 6: Economic Value Waste calculation.

The example block in Figures 4-6 would generate -299,880\$ if it is sent to the process, and -55,080.1\$ if discarded as waste. Therefore, this block might go to waste, since it would result in less prejudice than when it is processed. MiningMath defines the best destination regarding the set of constraints throughout the time, thus this decision a lot more complex than the example above in most cases.

Data Validation

Running an optimization for complex projects with several constraints may demand hours only to validate if the formatting has been done properly. Therefore, we present here an efficient workflow

to structure your project. It is important to mention that you can run each topic of this content individually at any time you find suitable, but, we strongly recommend you to take it step by step to understand the main idea of the process

The next pages use the example below within Marvin Deposit and aim to exemplify the following set of constraints:

Constraint	
Value	
Processing capacity	
10 Mt per year	
Total movement	
40 Mt per year	
Sum of processing hours	
4,000 per year (detailed estimate of the plant throughput)	
Vertical rate of advance:	
150m per year	
Copper grade	
Limited until 0.7%	
Minimum Mining Width	
50m	
Minimum Bottom Width	
100m	
Restrict Mining Surface	
Surface-RM-offset-400m.csv	
Due to a processing plant in the area	
Fixed Mining (Stockpiling)	
0.9\$/t	
Rehandling cost (Stockpiling)	
0.2\$/t	

Table 1: Set of constraints example.

Validate it First

In order to validate your data and cut its runtime, we strongly recommend running MiningMath Full with the following set up:

1. Process and dumps set with respective recovery values.
2. A bigger production capacity than the expected reserves. In this example, the expected life of mine vs production rate is 35-year producing 10 Mt per year. Hence, a value of 1,000 Mt would be big enough to cover the whole reserve.
3. No discount rate.

4. No stockpiling.
5. Density and slope values.
6. Timeframe: Years (1), since it would all be processed in 1 period.

Figure 1 depicts this set up at MiningMath, with the highlighted fields. This execution will also generate the topography surface so that you can create a force or restrict mining areas.

Figure1: Validation fields in the Overview tab.

Results

Figure 5: Production result

Figure 2: Sequencing

Figure 3: Surface

Figure 4: Surface with blocks

Figure 5: Production result

Figure 2: Sequencing

Previous image

Next image

Questions

- * Did the scenario run properly?
- * Are most of the positive economic values from the process inside this surface?
- * Is the mining happening in reseanable areas?
- * Is there a reasonable number of periods of life of mine?

Figure 5: Scenario's set up

Constraints Validation

Add Constraints to your Project

Continuing the validation, start to add the first constraints related to your project so that you can understand its maximum potential. The surface generated in this case could also be used as a restrict mining in the last period to reduce the complexity of your block model and the runtime of MiningMath, since it includes a set of constraints inputted.

Example:

- * Set up a scenario with 1,000 Mt in the processing plants, which corresponds to a lot more mass than expected in the whole life of the mine.
- * Add the Minimum Bottom Width (100m). This constraint will allow you to have a suitable work-front for your equipment.
- * Restrict Mining surface, if you have this constraint in your project.

* Grade constraint until 0.7%.

* Timeframe: Years (1), since it would all be processed in 1 period.

Note: Sum constraints can restrict the total amount of handling material (ore + waste) of the mine. Therefore, do not use them in the validation.

Figure 3: Surface

Figure 1: Sequencing

Figure 2: Restrict surface

Figure 3: Surface

Figure 1: Sequencing

[Previous image](#)

[Next image](#)

Let's make everything clear

Now that Constraints Validation step is done, you are able to use this final surface as a guide for future optimizations. This approach reduces the runtime and the complexity of the algorithm because when you take it into account, the blocks below this final optimized surface won't be considered and the heuristics inside the interface would be facilitated. Notice that we did not make any change in the discount rate, thus, this first NPV does not represent the reality. If you need an accurate result in this step, make sure to adjust it.

It's important to remember that when we restrict the mining into this surface, the number of periods generated in future runs could be reduced because the average parameters of each one will have to meet the constraints of the overall package. Therefore, to achieve the same parameters in a lower timeframe, some blocks may be discarded due to the mining sequence and the optimization of destinations inside the whole mass.

Having this idea in mind, you should already have enough information to decide and structure the next step of the optimization. Based on the amount mined in the last item and in the processing capacity, define a good timeframe to identify the mining sequence. In this case, we had 231 Mt as the ore total mass to split in almost 23 years, since the processing capacity is 10Mt.

To improve efficiency in the optimization, before working on a yearly basis, we made the decision to consider the first 5 years. It is reasonable to generate a 10-year surface to consider the optimization inside this limit due to observations made before. Remember that each assumption here can be done accordingly with your project's demands and that MiningMath can work with any timeframe to meet your needs.

Exporting Data

Exporting the Model

Select the button Export Model on MiningMath's Model tab, as shown below.

Figure 1: Clicking on Export.

Clicking on Export, a new page will appear, allowing you to select the folder where the block model exported would be saved with its name.

Figure 2: Exporting data.

Just click on "Next" for your model to be exported to the folder selected.

Public Datasets

Freely disclosed datasets

MiningMath allows you to learn, practice, and demonstrate, by showing any scenario previously ran the concepts of Strategy Optimization using the full capabilities of using only Marvin Deposit. This version is freely available to mining professionals, researchers, and students who want to develop their abilities considering this standard block model.

MARVIN DEPOSIT

* Source: MineLib.

* Download the original one.

* Download the predefined one in MiningMath.

DB Information

Below are listed the default parameters for Marvin according to the adaptations made in our formatted model.

Parameter

Value

Block size

23000 m³ (X = 30m, Y=30m, Z=30m)

AU - Selling Price

12 \$/g

AU - Selling Cost

0.2 \$/g

AU - Recovery

0.60

CU - Selling Price

2000 \$/ton

CU - Selling Cost

720 \$/ton

CU - Recovery

0.88

Mining Cost

0.9 \$/ton

Processing Cost

4.0 \$/ton

Discount Rate

10% per year

Default Density

2.75 t/m³

Default Slope Angles

45 degrees

Economic Values

* Process Function = BlockSize * Density * [GradeCU/100 * RecoveryCu * (SellingPriceCU - SellingCostCU) + GradeAU * RecoveryCu * (SellingPriceAU - SellingCostAU) - (ProcessingCost + MiningCost)]

* Waste Function = BlockSize * Density * MiningCost

McLAUGHLIN DEPOSIT

* Source: MineLib.

* Download.

DB Information

Below are listed the default parameters for the McLaughlin deposit according to the adoptions made in our formatted model.

Parameter

Value

Block size

X = 7.62m (25ft), Y = 7.62m (25ft), Z = 6.096m (15ft)

AU - Selling Price

900 \$/oz

AU - Recovery

0.90

Mining Cost

1.32 \$/ton

Processing Cost

12 \$/ton

Discount Rate

15% per year

Default Density

3.0 t/m³

Default Slope Angle

45 degrees

Economic Values

* Process Function = BlockSize * Density * [GradeAU * RecoveryCu * (SellingPriceAU) - (ProcessingCost + MiningCost)]

* Waste Function = BlockSize * Density * MiningCost

Output files

Execution options

The Execution Options or Run Options allow the user to define:

- * Files to be exported.
- * The visual results to be automatically shown on the viewer after each run.

Figure 1 highlights in (A) where the user can trigger this pop-up window and in (B) the options available, among which the user can:

- * Export/not export to CSV files:
 - o The resulting surfaces
 - o The resulting model in two ways: all blocks or only mined blocks, with/without coordinates and/or index information.
- * Set which results to be shown on the viewer:
 - o Surfaces
 - o Model

Figure 1: Execution options.

Outputs from MiningMath

MiningMath automatically produces:

- * Formatted reports (XLSX files).
- * Tables (CSV) whose data feeds the reports.
- * Updated block model (MinedBlocks or AllBlocks).
- * Surfaces as a grid of points (CSV)

How those files are organized?

MiningMath organizes files, as listed below:

SSMOD and SSPRJ are important to report any issues you face.

*

- o Model Folder
 - o MiningMath Model file (.SSMOD).
 - o MiningMath Project file (.SSPRJ).
- o Scenario folder
 - * Output Block Model
 - * MinedBlocks.CSV contains information about the mined blocks.
 - * AllBlocks.CSV, when requested, contains information about all blocks.

* Scenario file (.SSSCN) is a XML file read by the interface. Use it for a quick check on parameters used.

* Report file (.XLSX) summarizes some quantifiable results, including charts such as productions, average grades, and NPV.

* MiningMath also generates independent report files (.CSV) present in the report file (XLSX) as a backup:

* Production Process.

* Production Dump.

* Production Total.

* Grade Process.

* Grade Dump.

* Metal Process.

* NPV.

* Cumulative NPV.

* Surface files (Surface-##.CSV) formatted as a grid of points.

* List Item

SSMOD and SSPRJ are important to report any issues you face.

Block Model Output (MinedBlocks.csv x AllBlocks.csv)

After each optimization, MiningMath exports the block model in one of two formats:

* MinedBlocks.csv: The file presents only the blocks that have been mined from each scenario. Mined Blocks are exported by default, as it is a lighter file.

* AllBlocks.csv: The All Blocks file presents all the blocks, whether mined or not, from each scenario, so it is basically the original Block Model along with resultant information from the optimization.

The resultant model includes all columns imported (except the skipped ones) besides the following information:

* Mined Block shows whether (1) or not (0) a block have been mined.

* Period Mined shows in which period a block have been mined (-99 for blocks that have not been mined).

* Period Processed shows in which period a block have been processed (-99 for blocks that have not been processed).

* Destination informs the destination of each block - according to the order the user has added processing stream(s) and waste dump(s).

How to interchange them?

Figure 2 shows where the user can interchange of these options.

1. Click on the highlighted Execution button (A) to open the Run Options (B).
2. Select All blocks in model or Only mined blocks, as you need.
3. Hit OK, then Run.

By default, MiningMath exports the MinedBlocks file as a block model output

Results

MiningMath will generate a report directly on Microsoft Excel, as shown in the following image, and the optimized pit (blocks and surface) in the viewer in case the user chooses this option (right figure above). The automatic preview shows only the mined blocks, colored according to each mining period defined by the scheduler.

Reports

The results presented in the Excel spreadsheet show, in the Charts tab, the graphs relative to the reported results calculated in the Report tab. The processed mass results, discarded mass, stock development, Au/Cu percentage in the process, Au/Cu percentage in the dump, metal contained in the process, net present value and cumulative net present value are arranged individually in the Production Process 1, Production Dump 1, Stock Process 1, AU/CU - Grade Process 1, AU/CU - Grade Dump 1, AU/CU - Metal Process 1, NPV and Cumulative NPV tabs, respectively.

Figure 2: Results report.

Visual Results

By default, MiningMath exports only the Mined Blocks file showing them by period on the viewer, as in the following illustration. The user can change any exporting options on Run Options menu.

Figure 3: Visual results.

Mined Blocks

If the user chooses to export the model, MiningMath will automatically save the list of the scheduled blocks (MinedBlocks.csv) or all blocks (AllBlocks.csv) in the block model folder, as shown in the figure below, which can be imported into other mining software packages.

The files MinedBlocks.csv and AllBlocks.csv may contain indices and/or block coordinates, and all the imported data/parameters along with the following information:

Figure 4: Mined blocks.

- * Mined Block shows whether (1) or not (0) a block have been mined.
- * Period Mined shows in which period a block have been mined (-99 for blocks that have not been mined).
- * Period Processed shows in which period a block have been processed (-99 for blocks that have not been processed).
- * Destination informs the destination of each block - according to the order the user has added processing stream(s) and waste dump(s).

Video 1: Outputs and files' hierarchy.

Workflow

Articles

- * Super Best Case
- * Optimized Pushbacks
- * NPV Upside Potential
- * Best-Worst Range Analysis
- * Selectivity Analysis
- * Design Enhancement
- * NPV Enhancement
- * Bottleneck Analysis
- * Multivariate Sensitivity Analysis
- * Optimized Schedules
- * Short-term Planning
- * Multi-mine
- * Sustainable analysis
- * Uncertainties at the Beginning

Super Best Case

Get The NPV Upper Bound

In the search for the upside potential for the NPV of a given project, this setup explores the whole solution space without any other constraints but processing capacities, in a global multi-period optimization fully focused on maximizing the project's discounted cashflow.

As MiningMath optimizes all periods simultaneously, without the need for revenue factors, it has the potential to find higher NPVs than traditional procedures based on LG/Pseudoflow nested pits, which do not account for processing capacities (gap problems), cutoff policy optimization and discount rate. Traditionally, these, and many other, real-life aspects are only accounted for later, through a stepwise process, limiting the potentials of the project.

This setup serves as a reference to challenge the Best Case obtained by other means, including more recent academic/commercial DBS technologies available. The block periods and destinations optimized by MiningMath could be imported back into your preferred mining package, for comparison, pushback design or scheduling purposes. This is all available for free!

Example:

- * Processing capacity: 10 Mt per year.

* Stockpiling parameters on.

* Timeframe: Years (1).

Figure 2: Sequencing

Figure 1: Best case sequencing

Figure 2: Sequencing

Figure 1: Best case sequencing

[Previous image](#)

[Next image](#)

Figure 3: Production constraints

Advanced experience and refinements

It is important to mention that if you have multiple destinations, extra processing, or dump routes, it could be added for proper cutoff optimization. Besides that, the surfaces obtained here could be used in further steps or imported back into any mining package for pushback design and scheduling.

A refinement of the best case could be done by adding more constraints, preferably one at the time to evaluate each impact in "reserves", potential conflicts between them, and so on. You can try to follow the suggestions below for this improvement:

* All blending constraints

* All restrict mining aspects due to forbidden areas

* Extra processing or dump routes for proper cutoff optimization

* Sum variables (with caution), just in case some aspect must be controlled for the whole LOM at once.

* In case more efficiency is needed, the resulting surface obtained in the Constraints Validation step could be used as restrict mining for the runs here.

These scenarios might take longer and the main recommendation is to use powerful machine to run it, in parallel, while other optimizations are performed.

Optimized Pushbacks

Pushback Optimization is the process of generating and analyzing optimized pushback scenarios by varying the volumes of ore and waste in each case. This procedure is important to ensure the financial and operational viability of the mining project, as excessively large volumes can render the project unfeasible, while excessively small volumes can result in resource wastage or missed opportunities for ore extraction. By testing different volumes, it is possible to find an optimal point that maximizes the net present value (NPV) of the project.

MiningMath utilizes timeframes to generate pushbacks at different levels of detail. Timeframes are time intervals that divide the mine's lifespan into smaller periods. Different timeframes allow users to perform a fast evaluation of the impact of production volume on the NPV. If necessary, adjustments can be made to optimize production and reduce costs.

In Pushback Optimization, multiple optimized pushback scenarios are created with varying levels of detail, enabling users to have a comprehensive view of the impact of volume variations on the project's performance.

Work Through Different Timeframes

You can Identify timeframe intervals in your project, so that you can work with group periods before getting into a detailed insight. This strategy allows you to run the scenarios faster without losing flexibility or adding dilution for the optimization, which happens when we reblock.

The idea is to make each optimized period represent biennial, triennial, or decennial plans. MiningMath allows you to do it easily by simply adjusting some constraints to fit with the timeframe selected.

Example

Dataset: Preinstalled Marvin deposit. It can also be downloaded [here](#).

To exemplify this idea, consider the base scenario for performing a Pushback Optimization process using the Marvin dataset. The timeframes (in green), process and dump production limits (in green) and the vertical rate (in red) will be adjusted for the process of Pushback Optimization.

By varying the highlighted parameters above, the following decision tree has been constructed for Pushback Optimization.

Three different timeframes are explored: 3 years, 5 years, and 10 years. Each timeframe is associated with specific process and dump production limits. Such limits not only scale with their respective timeframes but also allow for variations that provide flexibility for testing different production scenarios. Finally, the vertical rate is also adjusted to align with the defined timeframe of each scenario. For instance, the vertical rate is set to 450m for the 3-year timeframe, 750m for the 5-year timeframe, and 1500m for the 10-year timeframe.

Afterward, specific results were carefully selected for comparison, focusing on key parameters such as Net Present Value (NPV), production process, and production dump.

Note: Waste control and vertical rate of advance are not recommended if you are just looking for pushback shapes.

More details

The 2 constraints inputted at the production tab are related to the maximum material handling allowed: the third one is about the processing equipment capacity, and the vertical rate of advance is

related to the depth that could be achieved adjusted to this interval. The minimum mining width was added because we are already generating designed surfaces that could be used later as guidance of detailed schedules, thus, it should respect the parameter due to the equipment sizing. Parameters such as average, minimum bottom and restrict mining surface, don't suffer any change in the time frames.

It's important to remember that the packages of time here don't necessarily have to correspond to identical sets of years. You could propose intervals with different constraints until reaching reasonable/achievable shapes for the design of ramps, for example. If you wish to produce more operational results, easier to design, and closer to real-life operations, try to play with wider mining/bottom widths rates. Those changes will not necessarily reduce the NPV of your project.

Considering this approach the discount rate serves just a rough NPV approximation and it doesn't affect much the quality of the solution, given that the best materials following the required constraints will be allocated to the first packages anyway.

Remember all the constraints

NPV Upside Potential

NPV Upside Potential is the process of generating and analyzing scenarios to measure the impact of each constraint on the project's net present value (NPV), from the Super Best Case to a detailed setup. Measuring the impact of each constraint on the NPV is important to assess the financial impact and ensure the project's viability under different scenarios and conditions. Each constraint can have a significant impact on the project's NPV, and it is crucial to understand how they affect the project's financial performance.

By evaluating the impact of each constraint on the project's NPV, it is possible to identify financial bottlenecks and opportunities for improvement, as well as prioritize problem resolution. This can result in better resource allocation and cost reduction, enhancing the project's profitability and viability.

In NPV Upside Potential, scenarios are created that sequentially incorporate each constraint of the project, allowing users to have a comprehensive view of the impact of each constraint on the project's performance. In case more efficiency is needed, the resulting surface obtained on the Constraints Validation or in Best Case refinements could be used as Restrict Mining in the last interval, which might reduce the complexity and the runtime.

Example

Dataset: Preinstalled Marvin deposit. It can also be downloaded [here](#).

To illustrate this process, let's consider the base scenario of the Marvin dataset (shown in the figure below). The highlighted green fields represent all the targeted constraints that need to be controlled in this project: process capacity, minimum average grade of CU in process, dump capacity, bottom minimum width, mining minimum width and maximum vertical rate.

The decision tree depicted below has been constructed for a NPV Upside Potential process, based on the above scenario. In this decision tree, the scenarios progressively introduce each constraint into the project.

The target scenario is the last one, with the following restrictions: Process Production=10mt, Dump Production=30mt, Bottom Width=100m, Mining Width=100m, Vertical Rate=150m, and average CU=0.5. However, the constraints are added interactively, starting with the process production, followed by the dump production, widths, and so on.

Note how the cumulative NPV usually decreases (as expected) when more restrictions are added (see note at the end for exceptions). Without this interactive process, there might be a lack of information to understand the NPV of the final, desired scenario.

Amount of constraints and NPV relationship

More constraints usually lead to lower NPV. The exceptions for that are the geometric constraints, which deal with non-linear restrictions. If you are not happy with the results achieved after adding geometric constraints you might need to perform a Selectivity Analysis or Best-Worst Range Analysis of your project.

Best-Worst Range Analysis

Best-Worst Range Analysis is the process of generating and analyzing scenarios to measure the impact of mine width constraints on the project's net present value (NPV), from no restriction to wide widths. Measuring the impact of mine width constraints is crucial to determine the optimal fleet equipment configuration in mining operations, with the aim of optimizing productivity and maximizing the net present value (NPV) of the project.

By analyzing variations in width constraints, it is possible to identify the effect of space limitations on mining operations and evaluate the influence of different bench widths on fleet performance. Appropriate mining widths can bring a series of benefits: higher amount of material to be simultaneously extracted; higher fleet productivity; more efficient transportation; easier road maintenance and so on. Hence, the search for different widths allows finding the best combination of equipment and mining techniques aimed at maximizing production and profit simultaneously in each scenario.

In a Best-Worst Range Analysis, scenarios are created gradually increasing the mine width up to a feasible maximum, allowing users to have a comprehensive view of the impact of space limitations on the project's performance.

Considering the nature of global optimization and the non-linearity of the problem, it is expected that there will be variations in performance (NPV, production, amount of mine fronts, etc.) as parameter values are modified. Therefore, it is crucial to generate a large number of scenarios to obtain a comprehensive analysis of the impact of these variations on the project. This way, a more precise understanding of how different parameter values affect the overall performance can be achieved.

Example

Dataset: Preinstalled Marvin deposit. It can also be downloaded [here](#).

Consider the following base scenario and decision tree built for a Best-Worst Range Analysis using the Marvin dataset.

The goal in this case is to understand the impact of different values of mining width (in green), which will be tested with a range of different values, from 0m up to 200m.

Note that there is no linear relationship between mining width and NPV. In other words, a higher mining width does not necessarily imply a lower NPV. As previously mentioned, that is due to the non-linearity of the problem.

Considering the nature of global optimization employed in MiningMath, other variables might also be affected by different mining widths. For example, the production could be analyzed for identification of possible issues when employing different mining widths.

Selectivity Analysis

Selectivity Analysis is the process of generating and analyzing scenarios to measure the impact of all geometric constraints on the project's net present value (NPV), from the most selective to the least selective setup. Analyzing the impact of variations in geometric constraints is important to determine the optimal mine configuration and optimize productivity and profit.

By performing such an analysis, it is possible to identify the effect of geometric limitations on mining operations. Moreover, it is possible to evaluate the influence of each parameter and its variation on mine performance. This allows finding the best combination of parameters and mining techniques aimed at maximizing production and profit for each scenario.

In a Selectivity Analysis, scenarios are created including each geometric constraint sequentially and gradually increasing or decreasing their values from the least selective until the desirable requirement. This allows users to have a comprehensive view of the impact of geometric limitations on the project's performance.

Considering the nature of global optimization and the non-linearity of the problem, it is expected that there will be variations in performance (NPV, production, amount of mine fronts, etc.) as parameter values are modified. Therefore, it is crucial to generate a large number of scenarios to perform a comprehensive analysis of the impact of these variations on the project.

Example

Dataset: Preinstalled Marvin deposit. It can also be downloaded [here](#).

Consider the following base scenario and decision tree built for a Selectivity Analysis using the Marvin dataset.

The goal is to understand the impact of different values of geometric constraints (mining width, bottom width, and vertical rate of advance). The geometric parameters (in green) will be tested with

a range of different values: bottom width with values from 0m up to 200m; mining width with values from 0m up to 200m; and vertical rate of advance with values from 50m up to 300m. In this example, 26 different scenarios were evaluated.

Note that there is no linear relationship between geometric constraints and NPV. In other words, a higher width or lower vertical rate of advance do not necessarily imply a lower NPV. As previously mentioned, that is due to the non-linearity of the problem. The cumulative NPV of the scenarios is compared in the graph below.

A diverse range of results can be achieved with a Selectivity Analysis. However, there are usually two possibilities when they are compared:

- * Contrasting geometrics parameters with small NPV variations: note that when the bottom width changes from 0m to 80m, and the remaining parameters are fixed, the NPV drops from 454 M\$ to 444M\$. This indicates that large changes in geometric constraints do not necessarily lead to large changes in the NPV. The same for the scenarios SA_BW000_MW100_VR150 and SA_BW100_MW100_VR250.

- * Similar geometric parameters with larger NPV variations: when comparing scenarios SA_BW080_MW100_VR150 and SA_BW100_MW160_VR150 there is a drop in NPV from 444M\$ to 370M\$, highlighting that the 20m and 60m change in bottom width and mining width respectively can lead to a larger NPV difference in the project.

In conclusion, it is important to create several scenarios in a Selectivity Analysis. As exemplified above, results can be quite similar or quite different due to the non-linearity of the problem. Considering the nature of global optimization employed in MiningMath, it is also important to evaluate other indicators. The figures below depict the tonnage achieved for the production, demonstrating the possible impacts for different geometric constraints.

Design Enhancement

Design Enhancement is the process of creating scenarios to conduct extensive searches for solutions with similar NPV values but with fewer violations and improved shapes. This process allows finding more efficient and sustainable solutions that meet specific mine constraints and needs. Hence, seeking these kinds of scenarios is important for optimizing mining operations and for reducing risks and costs.

In Design Enhancement, scenarios are created with more rigorous geometric constraints without compromising the desirable requirements. The goal is to reduce violations and find better forms for the project. This is possible due to the global nature of optimization and the non-linearity of the problem, enabling the use of stricter requirements for the geometric constraints that could lead to a better performance of the project.

Example

Dataset: Preinstalled Marvin deposit. It can also be downloaded [here](#).

Consider the initial scenario and respective decision tree built for a Design Enhancement process. The goal is to evaluate stricter variations in the geometric constraints (in green).

Note the variation of results for Cumulative NPV and production. The base scenario has a small violation on dump production for period 1. However, when modifying the minimum width to 120m (scenario DE_BW100_MW120_VR150) this violation is not present anymore. Hence, this is an example of how small variations in geometric constraints could lead to less violations.

NPV Enhancement

NPV Enhancement is the process of creating scenarios to conduct extensive searches for solutions with higher net present value (NPV) values and similar violations, while considering minimum requirements for project constraints. Scenarios are created that gradually modify constraints from desirable requirements to minimum requirements, with the goal of increasing project profitability.

Example

Dataset: Preinstalled Marvin deposit. It can also be downloaded [here](#).

Consider the initial scenario and respective decision tree built for a NPV Enhancement process. The constraints in green (production capacities, geometric constraints, and average CU) are considered for modifications, from desirable requirements to minimum requirements, in order to identify solutions with higher net present value (NPV).

Results show a high variation in NPV while the production remains in its limits. Hence, it shows that it is possible to achieve higher NPVs when employing minimum requirements defined by the user.

Bottleneck Analysis

Bottleneck Analysis is the process of creating scenarios to conduct extensive searches for solutions with fewer violations while preserving the NPV and geometries. Seeking solutions with fewer violations is important for optimizing mining operations and reducing risks. This allows for the discovery of more efficient and sustainable solutions that meet specific constraints and needs of the mine.

In Bottleneck Analysis, after analyzing a desirable scenario it is possible to identify the constraint/s with demanding requirements that directly impact the optimization results and cause significant violation issues. Then, scenarios should be created by relaxing these demanding parameters, enabling users to make decisions to mitigate risks and ensure project viability.

Example

Dataset: Preinstalled Marvin deposit. It can also be downloaded [here](#).

Consider the base scenario overview and the respective decision tree built for a Bottleneck Analysis depicted below.

The graphs below depict a comparative analysis of the results for the scenarios in the decision tree.

This analysis shows that the minimum average constraint of CU, production dump and vertical rate of advance are all restricting the base scenario. When relaxing these parameters, there is an increase of approximately 5% in the cumulative NPV, while the productions are still kept within their limits.

Multivariate Sensitivity Analysis

Multivariate Sensitivity Analysis is the process of creating and analyzing scenarios based on a range of possible values for selected constraints. Analyzing the impact of constraints variations is important for determining the optimal mine configuration and for optimizing productivity and profitability.

In Multivariate Sensitivity Analysis, scenarios are created gradually increasing or gradually decreasing the values of the constraints within a desired range, covering all combinations of values. This allows users to have a comprehensive view of the impact of combinations of constraint values on the project's performance.

Considering the nature of global optimization and the nonlinearity of the problem, it is expected that there will be variations in performance as certain parameter values are modified. Therefore, it is crucial to generate a large number of scenarios to obtain a comprehensive analysis of the impact of these variations on the project.

Example

Dataset: Preinstalled Marvin deposit. It can also be downloaded [here](#).

Consider the base scenario overview and the respective decision tree built for a Multivariate Sensitivity Analysis depicted below.

Results are depicted below. Note how a diverse range of cumulative NPVs is reported when compared to the Base scenario. Also not how certain productions are more stable than others, demonstrating the importance of performing a Multivariate Sensitivity Analysis.

Optimized Schedules

In the early years of the project you will find many concerns and the most value in terms of NPV. Knowing that, we decide to consider a 10-year surface to optimize the first 5 years. By a logical assumption, as the surface used corresponds to a decade, it will contain the interval from the 1st to 5th period and represents a lot more mass. However, this simple input can restrict the space where the algorithm has to find a solution, which could reduce the run-time and help it to deliver better results respecting the set of constraints given.

Example:

Figure 4: Processing hours

Figure 1: Sequencing

Figure 2: Surface

Figure 3: Production result

Figure 4: Processing hours

Figure 1: Sequencing

Previous image

Next image

Parameters

Value

Timeframe

Years (1)

Processing capacity

10 Mt per year

Total movement

40 Mt per year

Vertical rate of advance

150m per year

Maximum of processing hours

4,500 processing hours per year

Minimum Mining

50m

Bottom width

100m

Restrict Mining Surface

Surface002 from Exploratory Analysis

Stockpiling parameters

On

Grade copper

Until 0.7%

Figure 8: Sum constraints

Figure 5: Production constraints

Figure 6: Geometric constraints

Figure 7: Average constraints

Figure 8: Sum constraints

Figure 5: Production constraints

Previous image

Next image

Short-term Planning

MiningMath allows the integration between long and short-term. By running the Best Case, surfaces to guide the optimization were generated and they could be used as a guide based on the NPV upper bound. The Exploratory Analysis provides insights on what could be the challenges of our project and also operational designs that could be used in further steps. At last, we obtained a detailed Schedule by using, or not, a surface, which could be the final pit or any intermediary one, as a guide.

Considering this workflow, now you may have enough information on a reasonable long-term view to enhance the adherence/reconciliation of your plans. You could choose a surface and use it as force and restrict mining to refine everything inside it. Remember that Force Mining is responsible for making the mining achieve at least the surface inserted, which means that all the material inside its limits should be extracted, respecting the slope angles, while Restrict Mining aims to prohibit the area below the surface inserted to be mined until the period in which it has been applied.

Thus, MiningMath will reach this exact surface in the time-frame required and enable you to test different geometries, blending constraints, and any other variable that could be required in the short-term planning without interfering in the long-term overview. Additional helpful features in these refinements are the concepts of mining fronts and the design optimization, based on surfaces modification, that could be done respecting all the parameters and generating results accordingly with your needs.

Figure 1: Results generated using different helpful features.

Example

Parameters

Value

Timeframe

Custom factor (0.5)

Processing capacity
5 Mt per semester
Total movement
20 Mt per semester
Vertical rate of advance
60m per semester
Minimum Mining
120m
Bottom width
100m
Force and Restrict Mining Surface
Surface005 from Schedule Optimization

Stockpiling parameters

On

Play with steeper slope angles in the short term?

Yes

Table 1: Set of constraints example (1).

Figure 5: Sum constraints as listed in Table 1.

Figure 2: Production constraints as listed in Table 1.

Figure 3: Geometric constraints as listed in Table 1.

Figure 4: Average constraints as listed in Table 1.

Figure 5: Sum constraints as listed in Table 1.

Figure 2: Production constraints as listed in Table 1.

Previous image

Next image

Results examples

Figure 8: Resulting stock process tonnage from example in Table 1.

Figure 6: Resulting schedule from example in Table 1.

Figure 7: Resulting production from example in Table 1.

Figure 8: Resulting stock process tonnage from example in Table 1.

Figure 6: Resulting schedule from example in Table 1.

Previous image

Next image

Further details

The example above used fewer constraints, geometries were changed and the average grade was let free. It is very helpful to define the early years based on a semester timeframe, which can assist you to manage stocks and any other variables in the firsts 3 years, for instance. Note that the period ranges on MiningMath are based on the timeframe selected, therefore, you should adjust your variables accordingly with this value.

When we use Force+Restrict, we are telling the optimizer to break this volume into pieces and that it must mine this volume entirely, even if it is waste, so that the long-term view is respected. This way, you keep regarding the whole deposit while deciding what to do in the first periods. The approach here is quite different than a set of Revenue Factors for a series of LG/Pseudoflow runs, followed by adjustments to find pushbacks without math optimization criteria. It is worth mentioning that this kind of suggestion must be only applied at the beginning or at the end of the life of mine, since Force+Restrict Mining surfaces used in intermediate periods could interfere directly with the results.

Using timeframes

Another strategy is to optimize the short-term along with the long-term using different timeframes. In this approach, the integration between the short and long term visions is made in the same optimization process, facilitating the analysis and strategic definitions.

It is possible to consider:

- * shorter time horizons (weeks, months, quarters...) for the initial periods of the operation;
- * annual plans as far as needed, for a precise definition of discounted cash flow;
- * less detail for longer time horizons. They need to be considered in the overall view of the mine, up to exhaustion, but they consume optimization processing time that can be more focused on the early years of operation.

Thus, there is value maximization at the strategic level, and feasibility at the tactical level simultaneously. In addition, there's minimizing compliance and reconciliation problems, as well as improving communication between teams, by working in an integrated system.

On this strategy, each period range will represent the time interval chosen in the timeframe, and discount rate will be adjusted in alignment with the time interval choice. Other constraints such as production and vertical rate of advance (VRA) must be adjusted to match each interval on the period ranges.

In order to clarify this strategy, Table 2 and Figure 9 present a possible list of constraints for an example using timeframes:

Parameters

Period range 1-6

Period range 7-12

Period range 13-end

Timeframe

1/12 of a year (each period representing a month)

1 year.

10 years.

Processing capacity

2.5 Mt per semester.

30 Mt per year.

300 Mt per 10 years.
 Vertical rate of advance
 30m (minimum block size, at this case) per month.
 150m per year.
 1500m per 10 years.
 Minimum Mining
 120m
 120m
 120m
 Bottom Width
 100m
 100m
 100m
 Stockpiling parameters
 On
 Off
 Off
 Different/steeper slope angles in the short term
 Yes
 No
 No

Table 2: Set of constraints for a timeframe example.

Figure 9: Constraints chosen in the interface for a timeframe example.

Multi-mine

Multi-mine projects often have hidden opportunities to maximize value. Accordingly with the natural level of complexity, it is pretty common to make a few simplifications that might reduce value, producing sub-optimal solutions from the mathematical perspective, while using the traditional LG/Pseudoflow methodology.

MiningMath has global optimization algorithm which can overcome such challenges, when working with integrated multi-mine projects to find solutions that regards all the pits been mined or mixed at the same time, instead of an individual pit optimization within a multi-mine project, which provides a totally different overview. To handle with such projects, the block model must contain all the mining regions which should be considered to the simultaneous optimization, thus, if you have the pits mapped in different datasets, it is important to follow the steps suggested below:

1. Work with a single block model or single pit first, run the initial tests and understand this region before handling the block model modification.
2. Try to eliminate meaningless blocks, which would not affect the solution and could increase complexity.
3. Join a second model/pit* and understand the manipulation process to work with multi-mine projects.

- Play with surfaces, if you wish to refine the results, filter regions to not mine, and any other guide. Since the surface files on MiningMath has always the same order, a good way to work with it could be using an excel file, available [here](#), which is pretty useful to such modifications.

- Play with Mining Fronts, if you want to control the material which is extracted from each region.

4. Add the other regions and start use everything that you wish.

*This joint block model file should fullfill the same requirments of a single one, as mentioned at the forming data page.

Figure 1: Multiple pits project

The current version of MiningMath consider the same Vertical Rate, Bottom Width and Mining Width for the entire block model. However, in a multiple pit overview, each one could have different geometric parameters that influence the levels of selectivity. Regarding these scenarios, the main recommendation is to set the parameter from one pit, fix solutions of this one, since force+restrict mining has the highest level of priority, and start the optimization of the others which would already consider the mass that would be extracted from the first pit fixed.

Figure 2: Surface obtained in the first optimization

An efficient workflow is to run the first scenario without geometric parameters, which should be the validating or the Best Case, for scheduling optimization. Then set up a scenario with geometric parameters from the most selective mine which means, the smallest widths and biggest vertical rate (VR), which would be the least constrained scenario considering geometric aspects. Therefore, each surface generated by this approach will be used to fix solutions for Mine 1.

For instance, you could take the surface 1 generated and decrease the elevation in the other areas to regard the mass from the period one of mine 1 and what could be mined in the second pit. By having these results you can refine either surfaces or mining fronts to get the best results and do a sensitivity analysis of geometric parameters for multi-projects still preserving the global optimization.

Figure 3: Surfaces set-up

Sustainable analysis

Technology has been developed to incorporate social and environmental factors in the mining project optimization, assessing these impacts whilst maximizing its net present value (NPV). The method can quantify socio-environmental aspects, such as dust, noise, avoidance of springs/caves/tribes, carbon emissions, water consumption, and any parameter that could be controlled by its average or sum. These environmental and social aspects can be assessed following internationally recognized standards (ISO 14044).

Figure 1: social and environmental factors.

Minviro in partnership with MiningMath has developed an approach to integrate such quantitative assessment into strategic mining optimization. This enables socio-environmental impacts to be constrained in the mining optimization, and the economic cost of reducing them to be calculated as a consequence. The way to do it is by inserting these variables linked with each block of your model, following these instructions. Considering this methodology, published here, significant reductions in the global warming impact could be achieved with a small economic cost. For example, using an environmental constraint it was possible to reduce 8.1% of CO2 emission whilst achieving 95.9% of the net present value compared to the baseline, as you can see in the image bellow.

Figure 2: Reduction in enviromental impacts.

Several scenarios for mine development, processing setup, energy/water consumption, CAPEX (content in Spanish), OPEX etc. can be evaluated. It is also possible to include geometric constraints in order to restrict a mining area due to legal and site-specific issues that affects the local population, using this feature. Spatially and temporally explicit socio-environmental risks can be included in mining optimization, providing an opportunity to assess alternative project options or explore a socio-environmental cost benefit analysis. For each aspect considered, decision makers are able to propose a range of possible scenarios and assess the economic cost of constraining these to different levels.

Figure 3: Possible scenarios to assess the economic cost of constraints.

The decision-making board, which previously had access to one or a few scenarios, now has a cloud of possibilities optimized and integrated with the technical and economic aspects of the project, reducing risks and adding sustainable value. The mathematical intelligence behind it is based on modern and well-accepted Data Science and Optimization concepts academically proven. The methodology has been tested in real mining projects with gains in NPV ranging between 15% and 20% on average, where socio-environmental aspects haven't been added yet.

Figure 4: Performance over time.

Uncertainties at the Beginning

One of the many possibilities offered by MiningMath's approach is to have multiple overview scenarios to evaluate different project assumptions, before doing a more detailed work. It does not demand an arbitrary/automated trial-and-error cutoff definition, nor a fixed input in form of pushbacks that will guide further optimization steps within the boundaries of a simplified problem. A subtle but substantial implication is the possibility of seeing a totally different mine development throughout the mine life cycle for each project assumption change. This allows mine managers to have a clearer view of the decision-tree and the possibilities on their hands, to improve economic, technical, and socio-environmental performances.

Considering this context, mine managers can judge greenfield projects to know whether or not they should prioritize a geotechnical study. This could be done by running multiple scenarios, considering the expected variability for slope angles for a given deposit. For example, in a given deposit, benchmarks from similar deposits indicate the overall slope angles might vary between 35-45 degrees. Before reaching the conclusion using an in-depth geotechnical study, multiple scenarios can be used to estimate the economic impact of each possible assumption for the overall slope angle. The conclusion might, then indicate a low economic impact, that could postpone the need for a detailed study.

The same idea applies to any parameter, which ultimately represents a project assumption.

MiningMath conducted an illustrative example with 2000 simulations varying multiple parameters independently. The results produced the chart from Figure 1, showing the probability (Y-axis) and the Project's Value (X-axis). In this case, a detailed geotechnical study might be postponed, as the Project's Value varies between 700 to 1100 MU\$, in function of the OSA.

Figure 1: What would 2000 simulations say about NPV distributions?

Theory

Articles

- * Current Best Practices
- * MiningMath Uniqueness
- * Guaranteed Solutions
- * Theory Validation
- * Mining Optimization Algorithm

Current Best Practices

The mining planning models built with current best practices have developed shortcuts and approximations to try to deliver acceptable results that consider all the project's complexities and constraints. To handle it, powerful machines are required to find a solution and to simultaneously determine the optimum pit limit and mining sequence that deliver the maximum project value.

Figure 1 depicts a stepwise approach used by current best practices.

Figure 1: Current best practices: stepwise approach

These steps may include different strategies, technologies or algorithms. However, they are all usually solved individually in three larger stages:

1. Nested pits: when finding nested pits it is possible to employ the Lerchs-Grossmann (LG) algorithm, the Pseudoflow algorithm, destination optimization, direct block scheduling, or even more recent heuristic mechanisms.
2. Pushback definition: having the nested pits defined, the next step would usually be to perform the definition of pushbacks in a manual way by some expert mine planning engineers using a number of empirical rules. Automatic ways focused on NPV optimization could also be employed for pushback design, but these are usually under resource constraints and do not consider enough geometric requirements.
3. Schedules: finally, starting from a chosen pushback, the scheduling is performed. A myriad of techniques can be employed for that, such as direct block scheduling, genetic algorithms, (fuzzy) clustering algorithms, dynamic programming, and heuristic methods in general. All with different rates of success, but limited variety of solutions due to the single pushback input.

Regardless of the technologies or algorithms, in a stepwise approach the aim is to initially find the final pit limit that maximizes the undiscounted cash flow to then focus on block sequence within this final pit envelope. By constraining the problem and predefining inputs, these shortcuts (approximations) help to save time and computer resources, enabling such software to consider complexities such as ore blending requirements, different processing routes, stockpiling policy, truck fleet considerations, and so on.

With current best practices, thousands of potential schedules can be generated with a multitude of different methods, but they are all based on the same stepwise rationale, with one step guiding the other. Commonly, schedules follow from a set of nested pits and other fixed input parameters such as geotechnics, metallurgical performance, blending constraints, etc. Therefore, the results frequently present similar behaviours and restrict the full exploration of the solution space. MiningMath solves these issues through math optimization models that integrate multiple areas of

the business. It handles all parameters simultaneously and delivers multiple scenarios, accounting for both strategic and tactical aspects. To better understand MiningMath uniqueness please continue reading here.

MiningMath Uniqueness

MiningMath allows mining managers to improve their strategic analysis through risk assessments that are unconstrained by stepwise processes. Through math optimization models that integrate multiple areas of the business, MiningMath handles all parameters simultaneously and delivers multiple scenarios, accounting for both strategic and tactical aspects.

MiningMath optimization is not constrained by arbitrary decisions for cut-off grades or pushbacks, since these decisions are usually guided by prior knowledge or automated trial-and-error. Thus, each set of constraints in our technology has the potential to deliver an entirely new project development, including economic, technical, and socio-environmental indicators, along with a mine schedule, while aiming to maximize the project's NPV.

How can it be used?

MiningMath acknowledges that each project has its own characteristics. Thus, it also allows you to choose which workflow fits best in your demand and decide which one should be used. Straight from block model you can find solutions to your short-term, schedules, optimized pushbacks or super best case, as depicted in Figure 1.

Figure 1: Single-step approach employed in MiningMath. Straight from block model to short-term, schedules, optimized pushbacks or super best case.

Super best case

As MiningMath optimizes all periods simultaneously, without the need for revenue factors, it has the potential to find higher best case's NPVs than traditional best case procedures based on LG/Pseudoflow nested pits, which do not account for processing capacities (gap problems), cutoff policy optimization and discount rate. Usually, these, and many other, real-life aspects are only accounted for later, through a stepwise process, limiting the potentials of the project.

Discounted Cash flow x Undiscounted Cash flow

The use of LG/Pseudoflow methods to perform pit optimization aims to maximize the undiscounted cash flow of the project. On the other hand, MiningMath maximizes the discounted cash flow. Therefore, regions in which MiningMath has decided not to mine are, probably, regions where you have to pay for removing waste on the earlier periods, but the profit obtained by the discounted revenue from the hidden ore does not pay for the extraction.

A proper comparison between this methodology could be done if you import the final pit surface obtained from the other mining package into MiningMath, and use it as Force/Restrict mining. This way, MiningMath will do the schedule optimization using the exact same surface, which will allow you to compare the NPV for each case. Figures 2 and 3 depict two comparisons between undiscounted and discounted cashflows.

Figure 2: Undiscounted versus discounted cash flow optimization.

Figure 3: Undiscounted versus discounted cash flow optimization regarding a minimum mining width.

Pushbacks

MiningMath offers the option of producing Optimized Pushbacks with controlled ore production and operational designs to guide your mine sequencing. Having this broader view in mind, you are already able to begin the scheduling stage. The block periods and destinations optimized by MiningMath could be imported back into your preferred mining package, for comparison, pushback design or scheduling purposes.

Schedules

When using MiningMath, it is possible to define the pit limit and mine schedule simultaneously. That is, to determine which blocks should be mined, when this should happen and to where they should be sent to maximize the NPV, while respecting production and operational constraints, slope angles, discount rate, stockpiles, among others, all performed straight from the block model. This means that the steps of pit optimization, pushback and scheduling are not obtained separately, but in a single and optimized process.

Decision Trees

To help with all that, our software allows you to build Decision Trees which enable a broader view of your project and a deeper understanding of the impacts of each variable. This is all possible because MiningMath works with a global optimization which simultaneously regards all variables, instead of using a step-wise approach. The software provides different views and solutions for the same mine for each parameter changed and each possible objective.

Guaranteed Solutions

Multiple, complex constraints increase the likelihood of not finding or not existing feasible solutions. Nonetheless, MiningMath always delivers a solution, even if it could not honor the entire set of constraints imposed or had to reduce the NPV to find a feasible solution.

When dealing with highly constrained problems, other technologies might take hours or days to realize there is no feasible solution. The reason for that is because they usually employ generic optimization algorithms, not suitable to take decisions in a mining problem. In this case, the only option is to prepare a second execution with more flexible constraints, but still with no guarantee of feasibility.

On MiningMath, once an infeasible solution is detected, the algorithm takes decisions on which (less relevant) constraints should be flexible, returning some warnings to the user at the report. This is performed along the optimization process, without compromising the runtimes.

The constraints priority order, from the highest to the lowest, is depicted in Figure 1.

1. Force+Restrict Mining together using the same surface.
2. Slope Angles.
3. Force or Restrict Mining, same concept as above, but the surfaces here are corrected according to slopes and it might have some differences.
4. Minimum Bottom and Mining width, mining length.
5. Total Production Capacity

6. Vertical rate of advance.
7. Average and Sum, modeled as strong penalties in the objective function
8. Time limit
9. Improve the NPV

Figure 1: Constraints hierarchy order.

Theory Validation

MiningMath's results are only possible due to its proprietary Math Programming Solver (c). It consists of a Mixed Integer Linear Programming (MILP) formulation and linearization methods that tackle the challenging non-linear aspects of the mining optimization. In addition, it has its own Branch & Cut algorithm, which provides more efficiency than standard MILP optimizers since it's fine tuned to this specific optimization problem.

Another major advantage of MiningMath comes from the mathematical formulations based on surfaces (Goodwin et al., 2006; Marinho, 2013), instead of usual block precedences. Block precedence methods might lead to higher errors (Beretta and Marinho, 2014), providing slopes steeper (i.e. riskier, more optimistic) than requested. The use of surfaces eliminates these geotechnical errors and allows for block-by-block geotechnical zones, if needed.

These surface-based formulations allow MiningMath to include geometric constraints, and, consequently, find solutions that are closer to real mining operations. The user can guide geometries by including mining and bottom widths, mining lengths, maximum vertical advance rates, and forcing/restricting mining areas. You can better understand how each constraint interacts with all others here. Such constraints give freedom to the user to work, or not, with predefined cut-offs and pushbacks which might limit the space of potential solutions. An in-depth view of MiningMath's formulations and algorithm can also be seen here.

This approach (Figure 1) has been applied for years by clients, such as Vale, Rio Tinto, Codelco, Kinross, AMSA and MMG, with a growing number of licenses sold, press releases and academic research also proving the consistency of the implementation. With constant developments since 2013, MiningMath has reached a mature and robust state. It is the first and only singlestep mining optimization engine available in the market.

Figure 1: MiningMath's approach. From block model to schedule in a single step solved by its proprietary Math Programming Solver (c).

Mining Optimization Algorithm

MiningMath has a flexible mining optimization algorithm that consists of a Mixed Integer Linear Programming (MILP) formulation and linearization methods that tackle the challenging non-linear aspects of the problem. In addition, MiningMath has its own Branch & Cut algorithm, which provides more efficiency than standard MILP optimizers since it's fine tuned to this specific optimization problem.

One of the major advantages of MiningMath comes from the mathematical formulations based on surfaces 1 2 , instead of usual block precedences. Block precedence methods might lead to higher

errors 3 , providing slopes steeper (i.e. riskier, more optimistic) than requested. The use of surfaces eliminates these geotechnical errors and allows for block-by-block geotechnical zones, if needed.

Another crucial advantage is that MiningMath's formulation includes geometric constraints, allowing its algorithm to find solutions that are closer to real mining operations. The user can guide geometries by including mining and bottom widths, maximum vertical advance rates, and forcing/restricting mining areas. Such constraints give freedom to the user to work, or not, with predefined cut-offs and pushbacks which might limit the space of potential solutions. Hence, the software provides different views and solutions for the same mine for each parameter changed.

Eventually, linear solutions need to be mapped onto an approximate integer (block-by-block) solution that will represent the scheduling of the mining problem in the real-world. The intelligence to convert continuous solutions into integer and non-linear ones are made by MiningMath's Branch & Cut algorithm.

Summary steps of MiningMath algorithm

Algorithm's flowchart and mathematical formulation

MiningMath employs an innovative mathematical formulation and powerful proprietary Branch & Cut algorithm for mining optimization problems. A description of this mathematical formulation and the three main steps of the algorithm employed are given below.

Step 1: Initial assessment

Figure 1: Initial assessment of entire block model and inclusion of likely profitable blocks within an initial surface.

The first step of the mining optimization algorithm is to remove regions that do not add any value to the project. This is an initial assessment that considers slope constraints, reducing the size of the problem and providing a region of interest for the optimization process. Since MiningMath always employs surfaces in its mathematical formulations, this first set of likely profitable blocks are contained within an initial surface as depicted in Figure 1.

Step 2: Problem linearization and mining optimization

Figure 2: Example solution with geometric constraints

In the second step of the mining optimization algorithm, the non-linear, integer problem is approximated to an integer, linear one based on surfaces. For that, it is necessary first to define the common notation across the problem and its variables.

* 1Common notation

* 2Variables and weights

* 3Domain of variables

* S : number of simulated orebody models considered

* s : simulation index, $s = 1, \dots, S$

* D : number of destinations

* d : destination index, $d = 1, \dots, D$

- * Z : number of levels in the orebody model
- * z : level index, $z = 1, \dots, Z$
- * T : number of periods over which the orebody is being scheduled and also defines the number of surfaces considered
- * t : period index, $t = 1, \dots, T$
- * M : number of cells in each surface; where $M = x \times y$ represents the number of mining blocks in x and y dimensions.
- * c : cell index, $c = 1, \dots, M$.
- * G : number of unique destination groups defined. Each group might contain 1, all, or any combination of destinations.
- * g : group index, $g = 1, \dots, G$.
- * $x_{c,t,d}^z$: simulation-independent binary variable that assumes 1 if block (c, z) is being mined in period t and sent to destination d , and 0 otherwise.
- * $e_{c,t}$: simulation-independent continuous variables associated with each cell c for each period t , representing cell elevations.
- * $\overline{f_{t,g,s}}, \underline{f_{t,g,s}}$: continuous variables to penalize sum constraints violated for each period, group of destinations, and simulation. One pair of variables is necessary for each quantifiable parameter modeled block by block whose sum is being constrained. An example would be variables used to control fleet hours spent in different periods, groups of destinations, and simulations. More information about possible parameters here. Note also that the software allows the control of the average of simulations, instead of dealing with each simulation individually, and the control by the sum of destinations, instead of each destination individually.
- * $\overline{\alpha_{t,g}}, \underline{\alpha_{t,g}}$: user defined weights for variables $\overline{f_{t,g,s}}, \underline{f_{t,g,s}}$ with the same destination group g and period t . These can only be defined in the .ssscn files.
- * $\overline{j_{t,g,s}}, \underline{j_{t,g,s}}$: continuous variables to penalize average constraints violated for each period, destination, and simulation. One pair of variables is necessary for each quantifiable parameter modeled block by block whose average is being constrained. An example would be variables used to control the average grade of blocks mined in different periods, destination groups, and simulations. More information about possible parameters here. Note also that the software allows the control of the average of simulations, instead of dealing with each simulation individually, and the control by the sum of destinations, instead of each destination individually.
- * $\overline{\beta_{t,g}}, \underline{\beta_{t,g}}$: user defined weights for variables $\overline{j_{t,g,s}}, \underline{j_{t,g,s}}$ with the same destination d and period t . These can only be defined in the .ssscn files.
- * $e_{c,t} \in \mathbb{R}, \forall c, t$ $t = 1, \dots, T, c = 1, \dots, M$

* $x_{c,t,d}^z \in \{0,1\}$, $c=1,\dots,M$, $t=1,\dots,T$, $z=1,\dots,Z$, $d=1,\dots,D$

* $\overline{f_{t,d,s}}, \underline{f_{t,d,s}} \in \mathbb{R}_{\geq 0}$, $t=1,\dots,T$, $s=1,\dots,S$, $d=1,\dots,D$

* $\overline{j_{t,d,s}}, \underline{j_{t,d,s}} \in \mathbb{R}_{\geq 0}$, $t=1,\dots,T$, $s=1,\dots,S$, $d=1,\dots,D$

Having the set of variables defined, is possible now to define a mathematical model with an objective function and necessary constraints.

Objective function

Intuitive idea

1. Sum of the economic value of blocks mined per period, destination, and simulation.
2. Average the result by the number of simulations.
3. Subtract penalties for certain violated restrictions associated with some user defined parameters.

Requirements

* $V_{c,t,d,s}$: cumulative discounted economic value of block (c,z) in simulation s , period t and destination d . More about this calculation [here](#).

Formulation

$$\max \sum_{s=1}^S \sum_{t=1}^T \sum_{c=1}^M \sum_{z=1}^Z \sum_{d=1}^D (V_{c,t,d,s} x_{c,t,d}^z) - p$$

where

$$p = \sum_{t=1}^T \sum_{g=1}^G (\alpha_t g^{t-1} (\sum_{s=1}^S f_{t,g,s}^{t-1}) + \alpha_t g^t (\sum_{s=1}^S f_{t,g,s}^t) + \beta_t g^{t-1} (\sum_{s=1}^S j_{t,g,s}^{t-1}) + \beta_t g^t (\sum_{s=1}^S j_{t,g,s}^t))$$

Finally, the objective function is constrained by the restrictions below

Surface constraints

* $e_{c,t-1} - e_{c,t} \geq 0$, $c=1,\dots,M$, $t=2,\dots,T$

[caption id="attachment_14191" align="alignnone" width="5532"] Figure 3: Two surfaces (blue and yellow): a) not crossing each other and respecting constraint (2); b) crossing each other and not respecting constraint (2).[/caption]

Slope constraints

Intuitive idea

* Adjacent elevations in a single surface need to respect a maximum difference. This maximum will change based on which direction they are adjacent: x, y, or diagonally.

Requirements

* H_x, H_y, H_d : maximum difference in elevation for adjacent cells in x , y and diagonal directions

* X_c, Y_c, D_c : equivalent to H_x, H_y, H_d concept, the sets of adjacent cells, laterally in x , in y , and diagonally, for a given cell c , respectively.

Formulation

* $e_{c,t} - e_{x,t} \geq H_x, c=1,\dots,M, t=1,\dots,T, x \in X_c$

* $e_{c,t} - e_{y,t} \geq H_y, c=1,\dots,M, t=1,\dots,T, y \in Y_c$

* $e_{c,t} - e_{d,t} \geq H_d, c=1,\dots,M, t=1,\dots,T, d \in D_c$

[caption id="attachment_14202" align="alignnone" width="5260"] Figure 4: Maximum allowed difference (H_x , H_y , and H_d) in elevation between adjacent cells in contact laterally in the x direction (a), in contact laterally in the y direction (b), and in contact diagonally (c).[/caption]

Link constraints

Proprietary constraints not disclosed. Possible examples of constraints of the same type, but not the ones actually employed.

Intuitive idea

* Surfaces will define when blocks will be mined. For example, blocks between surfaces associated with period 1 and 2, will be mined in period two. A block is between two surfaces if its centroid is between the two surfaces.

Requirements

* E_c^z : elevation of centroid for a given block (c, z)

Formulation

* $E_c^z \times \sum_{d=1}^D x_{c,1,d}^z \geq e_{c,1}, c=1,\dots,M, z=1,\dots,Z$

* $e_{c,t-1} \geq E_c^z \times \sum_{d=1}^D x_{c,t,d}^z \geq e_{c,t},$
 $c=1,\dots,M, t=2,\dots,T, z=1,\dots,Z$

[caption id="attachment_14206" align="alignnone" width="4192"] Figure 5: Distance between centroids above surfaces (green lines) and below surfaces (red lines) respecting constraints (6) and (7). Blue blocks are mined in period 1, while yellow blocks are mined in period 2. [/caption]

Destination constraints

Intuitive idea

* Each mined block can only be sent to one destination.

Formulation

* $\sum_{d=1}^D x_{c,t,d}^z = 1, c=1,\dots,M, t=1,\dots,T, z=1,\dots,Z$

Mining constraints

Intuitive idea

* For each period and destination groups there's an upper and lower limit of total tonnage to be extracted. Destination groups might be formed by any unique combination of destinations, with 1, many or all. The sum of the tonnage of mined blocks sent to the same group of destinations in the same period must respect these limits.

Requirements

- * T_c^z : tonnage for a given block (c, z) .
- * $\overline{T}_{t,g}$: upper limits in total tonnage to be extracted during period t and destinations in group g .
- *

Formulation

* $\sum_{c=1}^M \sum_{z=1}^Z \sum_{d \in g} T_c^z x_{c,t,d}^z \leq \overline{T}_{t,g}$, $t = 1, \dots, T$, $g = 1, \dots, G$

Sum constraints

Intuitive idea

* The user can define a certain parameter (i.e. fleet hours spent) associated with each mined block to have its sum controlled. The sum of the values of this parameter associated to each mined block must respect lower and upper bounds for each period, destination groups (optional) and simulation (individually or on average). Destination groups might be formed by any unique combination of destinations, with 1, many or all.

Requirements

- * $\underline{F}_{t,g,s}, \overline{F}_{t,g,s}$: lower and upper limits, respectively, in sum of user defined parameter to be respected in period t , destination group g , and simulation s .
- * $F_{c,d,s}^z$: value of user defined parameter related to a given block (c, z) in destination d and simulation s .

Formulation

* $\underline{F}_{t,g,s} \leq \sum_{c=1}^M \sum_{z=1}^Z \sum_{d \in g} F_{c,d,s}^z x_{c,t,d}^z + \underline{f}_{t,g,s} - \overline{f}_{t,g,s} \leq \overline{F}_{t,g,s}$,
 $t = 1, \dots, T$, $g = 1, \dots, G$, $s = 1, \dots, S$

Average constraints

Intuitive idea

* The user can define a certain parameter (i.e. grade) associated with each mined block to be controlled in average. This average is weighted by the block's tonnage and by an optional, user defined weight. It must respect lower and upper bounds for each period, destination group (optional) and simulation (individually or on average). Destination groups might be formed by any unique combination of destinations, with 1, many or all.

Requirements

* $\underline{J}_{t,g,s}, \overline{J}_{t,g,s}$: lower and upper limits, respectively, for average value of user defined parameter to be respected in period t , simulation s , and destination group g .

* T_c^z : tonnage for a given block (c, z) .

* $J_{c,s,d}^z$: value of user defined parameter of block (c, z) sent to destination d in simulation s

* $P_{c,t,d,s}^z$: user defined weight for block (c, z) in period t , destination d , and simulation s

Formulation

*
$$\frac{\sum_{c=1}^M \sum_{z=1}^Z \sum_{d \in g} P_{c,t,d,s}^z T_c^z J_{c,s,d}^z x_{c,t,d}^z}{\sum_{c=1}^M \sum_{z=1}^Z \sum_{d \in g} P_{c,t,d,s}^z T_c^z} + \underline{j}_{t,g,s} - \overline{j}_{t,g,s} \leq \overline{J}_{t,g,s}$$

$t = 1, \dots, T, g = 1, \dots, G, s = 1, \dots, S$

Geometric constraints

Proprietary constraints not disclosed

Intuitive idea

* Surfaces should respect geometric parameters defined by the user, such as minimum bottom width, minimum mining width, and maximum vertical rate of advance, as depicted here.

Formulation

* $\text{Geometric}(e_{c,t}) \leq \text{text}\{\text{geometric restriction}\}, c=1, \dots, M, t=1, \dots, T$

Step 3: Integer, non-linear solution and evaluation

The next step in the mining optimization algorithm is to convert the linear solution to an integer, non-linear one. MiningMath's Branch & Cut method is responsible for this conversion. Once it is done, the resulting solution can be evaluated, leading to the end of the algorithm's execution or to a new optimization process. This new process might be triggered if one of the two situations arise:

1. restrictions are violated due to transformation from linear to integer, non-linear solution, or due to problem being infeasible.
2. an evaluation of certain restrictions in the transformed integer, non-linear solution concluded that they might not affect the problem and be better discarded or modified.

If any of these are true, the solution at this stage will be sent back to Step 2 for linearization and refinement. Thus, if this refinement is caused by situation 1) then the goal is to improve the solution's feasibility. This feasibility is improved according the constraint hierarchy order depicted in Figure 6.

Figure 6: constraints hierarchy order.

In contrast, if it is caused by situation 2) then the goal is to allow the optimization to focus on the bottlenecks of the problem and improve the current NPV. Once none of these situations have been identified, the current solution is returned. Note that each time the algorithm goes back to Step 2, a new global optimization is performed, thus the new resulting solution might be entirely different.

Pseudo-code

The whole process of the mining optimization algorithm, from input to output is summarized in the pseudo-code below. References are made to previous Steps 1, 2, and 3. This algorithmic flow together with the proposed mathematical formulation exemplifies the innovative methodology applied to solve a single mine scheduling problem.

INPUT: Block model,

Mining parameters,

Optional time limit T

OUTPUT: Excel report summarizing the main results of the optimization,

Outputs of mining optimization, topography, and pit surfaces using

.csv format that can also be imported into other mining packages.

EXECUTE initial assessment // Step 1

CREATE problem linearization P // Step 2

SET CURRENT_SOLUTION to empty

SET FEASIBLE_SOLUTION to empty

REPEAT // Step 3

SOLVE P // Optimization engine + proprietary Branch & Cut algorithm

SET LS to the integer, linear solution of P

TRANSFORM LS to an integer, non-linear solution RS

// Evaluate RS

IF RS has no violated constraints THEN

SET FEASIBLE_SOLUTION to RS

ENDIF

IF RS is better than CURRENT_SOLUTION THEN

SET CURRENT_SOLUTION to RS

ENDIF

```

// Evaluate if new iteration is necessary
IF FEASIBLE_SOLUTION is empty THEN
    // Step 2 and Figure 6
    CREATE new problem linearization P
        with flexible constraints
    CONTINUE // Go back to loop's start
ELSE IF T has been reached
    BREAK // Leave loop
ELSE IF RS has violated constraints that were unviolated in LS OR
    has constraints that can be discarded/modified THEN
    CREATE new problem linearization P // Step 2
    CONTINUE // Go back to loop's start
ELSE
    BREAK // Leave loop
ENDIF

```

WHILE TRUE

EXPORT reports and outputs from CURRENT_SOLUTION

Copy

General Content

Articles

- * Destinations
- * Recovery
- * Stockpiles
- * Stocked blocks
- * Artificial stockpiles
- * Reclaim Policy
- * Variable Mining Costs
- * Production Constraints
- * Average
- * Sum

- * Material Types
- * Mining Fronts
- * Stochastic Models
- * Time Limit

Destinations

Destination Policy of MiningMath

MiningMath aims to maximize the NPV of a mining project, and as such, it uses a discount rate in all calculations, it also considers the value of money through time. The software decides which blocks will be mined, when, and to which destination they must be sent. The mathematical model bases its decision on the economic values of each possible route. It means that MiningMath aims to identify, in a global view, which is the best destination to increase the NPV, respecting, simultaneously, all the constraints and using the priority order listed here.

Adding Destinations

Figure 1 shows, on the right bottom corner of the screen, the buttons responsible for add/remove destinations.

Figure 5: Stockpile limit

Figure 1: Type's panel and buttons

Figure 2: Processes

Figure 3: Recoveries

Figure 4: Economic values

Figure 5: Stockpile limit

Figure 1: Type's panel and buttons

Previous image

Next image

The panel Type shows the type of each destination added. The user must add at least:

- * One (1) processing stream;
- * One (1) waste dump

The numbers at the left of the screen are identifiers for each route at the mined blocks output file.

Read more about multiple destinations.

Renaming

Especially when using multiple destinations, the user should consider using more meaningful names for each route. Figure 2 highlights the panel Name, where the user can rename each one.

Process Recovery

Figure 3 shows the recovery panel, which is intended to input recoveries used during economic value calculation. The difference is that now it is for reporting purposes, which means it is not being considered twice.

Economic Value

Figure 4 highlights the Economic Value panel. Here, the user can assign each economic function to the proper destination.

Stockpile limits

As shown in Figure 5, stockpile limits are available only for processing streams, if activated in the General tab. These limits are valid for the life of mine.

[Read more about stockpiles.](#)

Recovery

Mineral Processing

One of the most important and basic concepts in mineral processing is metallurgical efficiency. Two terms are used to describe this efficiency: recovery and grade.

Recovery

The recovery or percent recovery refers to the ratio of the valuable material (metal or mineral) contained in the concentrate with reference to the amount of the same material in the processing plant feed.

How and where?

The main place where the user input the recovery information is when defining the Economic Values.

However, as this information is implicit in the economic functions, the user need to input this value on the interface for purposes of generating reports.

On the Destination tab (light green), the user can define recoveries for each element in the panel (dark green) from Figure 1.

Figure 1: Destinations tab and recoveries.

Varying Recoveries

Why?

A detailed mine planning is likely to require an iterative process to update a block model with new information.

Considering the usage of specific tools for measurements, analysis, and reporting, mine planners might be interested in using the thorough information acquired on recoveries along the way.

How?

When editing the data set, the user can add as many columns as needed, defining recoveries for each block. Figure 2 shows how it would look like.

Figure 2: Recovery columns in the block model.

Figure 3: Imported recoveries available to the user.

Figure 3 shows a dropdown menu where the user can choose which recovery to use:

- * RecoveryA
- * RecoveryB
- * Constant recovery

NOTE the user does not need to use all the recovery fields imported for each run. This means recovery fields might be created for further scenarios, being used separately. The same concept is valid for columns with slopes and economic values.

Step-by-step

The following video presents how to use a different recovery for each block.

Video 1: Varying recoveries for each block.

Stockpiles

How MiningMath deals with Stockpiles

Stockpiles are a post-processing unit of the algorithm. Therefore, after the optimization, which generates mining surfaces, the algorithm does its analysis among the discarded blocks. MiningMath checks if the block's values (Revenue - Fixed Mining Cost - Rehandling Cost) is higher than the cost of discarding it, with the respective discount rates applied at the period of extraction and period of processing. Therefore, the stocks are considered as an optimization of the blocks that initially went to the dump.

Figure 1: Stockpiles on MiningMath

When choosing the destination, the same logic of maximizing the project value is used and MiningMath will define which discarded blocks will be reclaimed, complementing production shortfalls over time, Figure 1 shows the process.

Stockpiles setup

To enable the stockpiles on the interface the first step is on the General tab (Figure 2) where two inputs are required:

- * Fixed Mining Cost: value used to decompose the economic value while considering stockpiles;
- * Rehandling Cost: represents the cost to reclaim blocks from the stockpile to the process.

After that, on the Destinations tab, you can define stockpile limits (Figure3) for each processing plant added, remembering that this limit is based on the life of mine, not in a period timeframe.

Figure 3: Stockpile limit

Figure 2: General tab

Figure 3: Stockpile limit

Figure 2: General tab

Previous image

Next image

More Information

Defining a Fixed Mining Cost is required because MiningMath does have an interface to enter these parameters when calculating the economic values. It is only used to decompose the block value so that the algorithm can make proper calculations to adjust values considering that part of its costs will occur when mined, while others may be charged when processed.

The Rehandling Cost is the cost to reclaim a block from a stockpile. Therefore, it is the way to break the Economic Values into parts and apply the discount rate at the time a block is processed.

Disclaimer: stockpiles and total tonnage

Since stockpiles are a post-processing unit of the algorithm, total tonnage restrictions might be violated when the stockpiles are processed. Total tonnages restrictions are only considered during the processing unit of the algorithm.

Read More

- * Which blocks have been stocked?
- * Which criteria is used for the reclamation policy?
- * How to incorporate variable mining costs into the economic modeling?
- * How to incorporate a pre-existing stockpile from ongoing operations?
- * How to incorporate underground material into the optimization along with open-pit material?

Stocked blocks

Tracing Stocked Blocks

There are two common ways that mining software packages deal with stockpiles:

- * Reclaiming blocks according to an average grade for the entire stockpile.
- * Reclaiming blocks selectively in any required sequence such as FIFO (first in, first out), LIFO (last in, first out), etc.

Both options are approximations and have their advantages and disadvantages.

Currently, MiningMath reclaims blocks selectively from stockpiles according to the highest Economic Values. Read more about Reclamation Policy.

How to trace blocks?

The user can analyze MiningMath's output in the files MinedBlocks or AllBlocks and report the periods in which a block has been mined and when it has been processed.

Video 1: How to trace stocked blocks.

Artificial stockpiles

Artificial Stockpiles is an advanced operation to incorporate external sources of material to your model and use them as an input on your scheduling.

Related common needs

This artifice may be required for cases when the user needs to include into the optimization, any of the following cases:

- * Pre-existing stockpile from ongoing operations.
- * Underground material to be blended with open-pit material.
- * Ore bought from third-part companies to fulfill production shortfalls.

How to incorporate artificial stockpiles

Two main ways to do it:

- * Modelling the stockpile with its actual geometry.
- * Creating a simplified artificial stockpile.

Modelled Stockpile

Modelling an existing stockpile with its actual geometry is the best alternative to include it to the scheduling for cases where you need an operational control over:

- * The stockpile and its adjacent areas.
- * The stock reclamation.

For this process, use a modelling software to perform the following steps:

1. Use the previous topography for the base of the stockpile.
2. Use the current topography for the top of the stockpile.
3. Create blocks in-between these surfaces. These blocks will have the same size of the block model.
4. Assign an average quality (grade) and density to each block created.
5. Calculate the economic values for these stocked blocks.
6. Import the model back to MiningMath to further scheduling.

Simplified Stockpile

For a quicker process, the user can create as many blocks as needed using a spreadsheet application.

This is an alternative useful for cases where you need:

- * A quicker process and evaluation.
- * Less operational control.

Rows vs Columns

You can model artificial stockpiles as rows or as columns.

Rows

- * To control a sequence, it may require surface constraints.
- * A 1-line row will be affected by minimum widths used for the scenario.

- * Thin rows may cause problems to be mined completely.
- * Easier setup. Rows don't need geotechnical adjustments.
- * Multiple rows will give more flexibility and reduce conflicts with the operational constraints from the open-pit scheduling.
- * If you opt for multiple stockpiles, create them with a 2-cell distance to avoid overlapping interference.

Columns

- * Easier sequence. The precedence is defined by the vertical geometry.
- * A 1-line column will be affected by minimum widths used for the scenario.
- * Long columns may affect how deep the scheduling can go in a single period.
- * Columns require a very vertical slope angle set to them and to the adjacent cells.
- * Multiple columns will give more flexibility and reduce conflicts with the operational constraints from the open-pit scheduling.
- * If you opt for multiple stockpiles, create them with a 2-cell distance to avoid overlapping interference.

Step-by-step (rows)

1. Create rows of blocks above the topography (Figure 1).
2. Assign an average quality (grade) and density to each block created.
3. Calculate the economic values for these new blocks.
4. Import the model back to MiningMath to further schedules.

Figure 1: Illustration of a row of 4 blocks created (in green) above a flat topography.

Requirements and observations

Consider checking the following requirements and observations.

Notes on modelling:

- * The blocks created must have the same size of the ones from original model.
- * Consider increasing densities for the blocks created to represent more material with a few blocks. The trade-off is a reduced selectivity.
- * The more blocks you have, the more selective is the algorithm.

Notes on operational needs:

- * The blocks created will be subject to the operational constraints, such as widths and vertical advance, from your scenarios. This means you need to consider these parameters to define:
 - o The stockpile base width.
 - o The stockpile height.

Notes on the placement within the model

The artificial stockpile should be placed in a peripheral area of your model to not affect the open-pit schedule.

Avoid borders to prevent any geotechnical issue, which will impede mining the artificial stockpile completely.

More information about vertical stockpiles

The following video shows more information on artificial stockpiles.

Video 1: Artificial Stockpiles.

Additional control over sources

Use Other Constraints to have additional control over different sources of material, whether they come from the original model or from a modeled stockpile.

Figure 2 shows an example on how to setup your model to track and material from different sources and control them by inputting minimum and maximum limits.

Figure 2: Example of how to use other constraints to track and control from different sources.

Reclaim Policy

Stock Reclaim Policy: Principles and Decisions

Stockpiles are handled as a post-processing unit of the optimization algorithm (read more). Also, another important point is to understand what guides MiningMath's algorithm decisions for stocked and reclaimed blocks. MiningMath's reclaim policy is based on the economic values. This goal is aligned with the objective of maximizing the NPV. This means, blocks with higher value will be reclaimed first, regardless of the period in which the block has been mined and stocked.

How could we further control stockpiles and make it suitable to different sorts of controls?

Reclaim Policies vs Selectivity

Different mining packages may adopt different conventions for Stock Reclaim Policies, overall, based on the main goal for each application or module. Some of the possibilities may include:

- * FIFO: First In, First Out.
- * FILO: First In, Last Out.
- * Reclaiming an average grade for the entire stockpile.
- * Reclaiming the highest-value blocks first (MiningMath uses this method).

Each one of these possibilities is an approximation of reality, with its pros and cons. FIFO and FILO are quite logic but represent a level of selectivity that is not practical. The angle of repose and the positioning of each block are likely the most intuitive examples of lower selectivity in practice.

An average grade for the entire stockpile is naturally an approximation, considering the amount of material that should be blended to make it close to the reality.

Reclaiming the high-value blocks first is also a selectivity level that is not possible in reality. On the other hand, it is quite aligned with the mathematical goal of maximizing the project's NPV for strategic evaluation.

After all, notice that none of these approaches are fully operational. Here, the decision is still based on the professional in charge of strategy optimization, one's preferences, experience, and skills to add further ways of control.

Preserving Value and Guiding the Reclaim Policy

This section aims to bring a few ideas on how the user might guide the algorithm in order to follow one's preferred reclaim strategy.

Baseline Knowledge

For this article, the user should have prior knowledge in the following concepts.

- * Manipulating the Economic Values to pre-define destinations.
- * Creating multiple economic assumptions to handle fake destinations.
- * Handling different output files (Mined Blocks and All Blocks).

Baseline Scenario

The general idea is to set up and run a baseline scenario to find what is optimal for the long-term value. The solution obtained in this step will guide further executions.

Output Format

The first step is to switch the output format to export the entire block model along with the optimization output information, which is represented by the AllBlocks.csv. This is a basic step for any iteration that requires re-optimizing a previous solution. By default, MiningMath exports only the MinedBlocks.csv.

Results will present a guide of which blocks should be mined, when, and which ones were mined, immediately processed, stocked then processed, or discarded.

Iterating over multiple scenarios

The idea here is to use previous outputs to create new columns of economic values, assuming the use of fake destinations, i.e. creating multiple processing streams that does not coexist and, in fact, represent the same single plant. These fake destinations, along with a pre-definition of the destination through economic values is what allows the user to impose whatever sort of control is preferred.

Along with that, and based on the previous results, the user must manipulate new economic values pre-defining the final destination for each block.

Hands-on Approach

Steps here comprise:

- * Define a criteria for stocked blocks (Period Mined different from Period Processed) that should be reclaimed early or later. This criteria must be based on the previous results from the AllBlocks.csv file (Figure 1), and on the columns Mined Block, Period Mined, Period Processed, and Destination.

* Pre-define destinations, based on the criteria adopted, by manipulating economic values. Use very negative values to prevent a block from going to a given destination, as shown in Figure 2.

* Set up your scenario considering the pre-defined destinations will not coexist, as illustrated in Figure 3. Notice that:

- o Process 1 and its stockpile will be used from Period 1 to Period 5.

- o Process 2 and its stockpile will be used from Period 6 to Period 10.

- o Process 3 and its stockpile will be used from Period 11 to .

* Optimize the new scenario to have a better approximation for the final NPV, considering the strategy of your preference.

FIFO Approach (Example)

For the FIFO approach, a block stocked during the second period on the first run, should be sent to the stockpile of Process 1 to be reclaimed first. Hence, this block must have a very negative value for all other destinations.

Figure 3: Setting up multiple fake destinations on the interface. Each processing plant does not coexist with one another.

Figure 1: Information extract from the AllBlocks.csv file showing the key information to pre-define new destinations.

Figure 2: Economic Values manipulated to pre-define destinations.

Figure 3: Setting up multiple fake destinations on the interface. Each processing plant does not coexist with one another.

Figure 1: Information extract from the AllBlocks.csv file showing the key information to pre-define new destinations.

[Previous image](#)

[Next image](#)

Variable Mining Costs

MiningMath was conceived with some simplifications for current version, such as the Fixed Mining Costs.

The software considers a fixed mining cost, which is applied beforehand in the economic formulation. This value is inputted in the user interface to make the software able to recognize this variable and fragment the previously calculated economic values in order to apply proper discounts to each block, whether (1) processed, (2) discarded, or (3) stocked and processed - here, mining costs are applied earlier than the processing costs and revenue.

However, mining costs might be defined with some variability for purposes of higher detail.

This article shares some ideas to consider Variable Mining Costs, where some goals are based on different:

- * Haul costs, in function of block's depth and/or the destination site.

- * Blasting costs, in function of the rock type.
- * Loading costs, in function of a selective mine
- * Supplies and materials, labor costs, among others.

How to estimate a Fixed Mining Cost

1. Prior to the model import, create a column of mining costs by block - including whatever costs are applicable: blasting, haul, loading, etc.
2. Calculate the Average Mining Cost of all of the blocks.
3. During the model import, assign the Mining Costs column to the field type Other; MiningMath will export this column along with its output.
4. Use the Average Mining Cost as the Fixed Mining Cost for the stockpiles.
5. Run MiningMath.
6. (OPTIONAL, for more accurate approximation) Analyze the MinedBlocks file (or AllBlocks) and calculate the Average Mining Cost just for the stockpiled blocks (Check out, Tracing Stocked Blocks)
7. (OPTIONAL) Run MiningMath again, now using the new Average Mining Cost (from Step 6) as the Fixed Mining Cost for the stockpiles.

Mining Costs in the stockpile context

For mined blocks in the stockpiles context, the possibilities for the material flow are:

1. Mine-to-process
2. Mine-to-waste
3. Mine-to-stock
4. Stock-to-process

The mining costs for flows (1) mine-to-process and (2) mine-to-waste are already embedded in the economic formulation.

The costs for flow (4) stock-to-process consider an additional cost, represented in the user interface by the re-handling cost.

MiningMath assumes the costs for (2) mine-to-waste and (3) mine-to-stock are the same, which might not be always the case due to, for example, different distances at the mine site surface (different haul costs).

How to consider different costs for discarded and stocked blocks

Option 1: Manual NPV Calculation

For which the steps are:

1. Open the MinedBlocks.csv.
2. Edit the stocked blocks' value to add new costs.
3. Calculate the NPV manually.

Option 2: Iterative Process

1. For the first execution 1, set up MiningMath to export the AllBlocks.csv and run.
2. Use the output model (AllBlocks) to update the economic value for stocked blocks. Then, save as a new model.
3. For the execution 2, re-import this model with fixed surfaces for all periods. As in the Figure 1.

Observation

This process does not guarantee solution will not change, i.e. the stocked blocks might change, which might require further iterations for a better approximation.

As stockpiles are treated as a post-processing unit in our algorithm, not being part of the optimization, Option 1 is more suitable for a more assertive NPV calculation.

DEVELOPMENT NOTES

The following features are already in our pipeline:

- * Optimized Stockpiles: future versions will consider Stockpiles as destinations when optimizing the NPV.
- * Varying mining costs: future versions will have a field such Density, Slope and Recoveries, which varies by block.

We have a partner working on a calculator and post-processing analysis tools to be integrated with MiningMath in the future. For more details, contact our support.

Production Constraints

Introduction

The Production tab allows the user to set period ranges and its corresponding production limits. This functionality allows the user to consider options such as pre-stripping, production ramp-up, prices changing over time, among others. It is worth mentioning that the discount rate on MiningMath is already applied in the first period. At the training presentation on this page, you can check the NPV formulas used.

Period Ranges

Figure 1 presents a panel where the user can define period ranges.

The Period Ranges feature allows the user to vary correspondent variables over time (see the last section of this page) such as:

- * Production limits
- * Limiting surfaces
- * Blending constraints
- * Other constraints

The user is able to edit only the field To. Subsequent periods will have their From field adjusted automatically.

Figure 5: Multiple periods

Figure 1: Period ranges

Figure 2: Timeframes

Figure 3: Production limits

Figure 4: Add/remove intervals

Figure 5: Multiple periods

Figure 1: Period ranges

Previous image

Next image

Timeframes

Figure 2 highlights the timeframe panel, where the user can choose different values for their projects. The values are used to deliver more accurate sequencing and have their values attached to a yearly basis calculation. It is possible to select a predefined value or to input a customized one.

In this example, we used a timeframe of 1 year, meaning that each generated period on this sequencing represents one year.

Production Limits

Figure 3 highlights the Production panel where the user can define limits for any destination added.

In this example, we have the following limits:

- * Process 1: 30,000,000 t

- * Dump 1:

- * Total: 60,000,000 t

These limits are being considered from Period 1 up to the end of the life of mine.

Add/Remove Intervals

Figure 4 highlights buttons to add and/or remove intervals, which are shown in rows in Figure 4.

Multiple Periods

Figure 5 presents an example where multiple periods were added.

A user has defined two period ranges as follows:

- * From 1 to 4

- * From 5 to <end>

For the first range, the user clicked on the field To (A) and set a value of 4. Automatically, the field From (B) of the second range, changed to 5.

The user can repeat this process to fragment the life of mine, and change parameters over time, as much as required. In this example, production limits increased 100% for Process 1 and 50% increase in Total from Period 5

Notes

The example from Figure 2 shows that the user does not need to set an explicit limit for all destinations available. In this case, Dump 1 has an <unlimited> upper bound. This means MiningMath will have enough flexibility to reduce process throughput - and increase dump tonnages up to the Total limit - whenever it finds this is a solution that increases the project's NPV. For users willing to have more control and reduce this flexibility, simply define a limit for each destination.

Average

What is an Average constraint?

Average constraints are based on the average of any quantifiable parameter modeled block by block. To use this feature on MiningMath, the dataset must contain an auxiliary field/column which considers a value of what you wish to limit regarding the value of it in each block. Therefore, this feature controls the average value of the variable that has been modeled considering blocks that were mined in that single period. Since this feature is based on average parameters, the algorithm can use lower values to respect this target and increase the NPV with higher ones.

This feature is usually applied on blending to combine low-grade and high-grade blocks in order to increase the profitability. Although, it could have a lot of other applications. Basically, any variable which could be modeled considering these assumptions could be controlled.

Video 1: Blending and other constraints.

Some examples using average are listed below:

- * Grade of a contaminant on the plant.
- * Haulage distance, based on the destination each block.
- * Blasting material consumptions.
- * CO2 and noise emissions, energy and water consumption and other socio-environmental.

The user can define:

- * Minimum and maximum average limits.
- * Different limits for different materials.
- * Different limits for different intervals.
- * Different limits for different destinations.

How to use Average?

Step-by-step

1. Create auxiliary fields in the block model, quantifying the information to be controlled.
2. During the importation, assign the column to be blended to Grade (Figure1).

3. On the Average tab, input minimum and maximum limits for each variable (Figure 2a), period range (Figure 2b), other weights to be considered (Figure 2c) and destination (Figure 2d).

Figure 1: During the importation, Cu and Au are assigned to "Average".

Figure 2: Fields where the user can input limits (A), for each period range (B), Weights (C) and each destination (D).

Sum

What is a Sum constraint?

Sum constraints are based on the sum of any quantifiable parameter modeled block by block. To use this feature on MiningMath, the dataset must contain an auxiliary field/column which considers a value of what you wish to limit regarding the value of it in each block. Therefore, this feature controls the total amount of the variable that has been modeled based on blocks that were mined in that single period. Basically, any variable which could be modeled considering these assumptions could be controlled.

Video 1: Blending and other constraints.

Some examples are listed below:

- * Tonnages and proportions of rock type and metal production.
- * Consumption of inputs such as energy spent during comminution, and fleet hours spent to mobilize material.
- * Contaminants control on the processing plant during each period.

The user can define:

- * Different sum limits for each material.
- * Different sum limits for each intervals.
- * Different sum limits for each destinations.
- * Combine all the options above in order to achieve globally optimized results.

How to use Sum Constraints?

Step-by-step

1. Create auxiliary fields in the block model, quantifying the information to be controlled (Figure 1).
2. During importation, assign these auxiliary columns to Sum (Figure 2).
3. On the Other tab, input minimum and maximum limits for each variable, period range and destination (Figure 3).

Figure 3: Sum tab

Figure 1: Block model setup to incorporate information of tonnage for each lithology.

Figure 2: During the importation, Tonnes A and Tonnes B are assigned to "Sum".

Figure 3: Sum tab

Figure 1: Block model setup to incorporate information of tonnage for each lithology.

[Previous image](#)

[Next image](#)

Material Types

Controlling materials

Usual definition and modeling

Mining fronts, rock type, or lithotype are usually defined in the block model as strings or converted to integer numbers. As the next step, the user defines which ranges of material types should be allowed, avoided or forbidden in the processing plant.

The ideal way to model material types for further control of this variable is to create tonnage columns for each material type. Therefore, you will be allowed to:

- * Control material types to be allowed, avoided, or forbidden in any destination.
- * Control the proportions of different material types, if applicable.
- * Analyze scenarios with different levels of flexibility.
- * Understand how the project development changes in face of each hypothesis tested.
- * Assess the impacts of the flexibility level given on economic and technical variables of the system.

Practical Steps

Block Model Setup

The general idea is to create auxiliary columns in the block model to control any variable through their sum. Then, use the same idea of if-then-else statements, example:

- * If variable value matches condition
- * Then auxiliary column equals to X
- * Else auxiliary column equals to Y

Example 1: Controlling material type

To control the amount of material by rock type, create columns for tonnages of each lithotype, as shown in Figure 1, where:

- * Lithotype A has its tonnage inputted in the field Tonnage A.
- * Lithotype A equals zero for fields Tonnage B and Tonnage C, as it does not match the specified condition, i.e. being a lithotype B or C.

The same concept is used to the other material types.

Figure 2 shows the same concept being applied to measured, indicated, and inferred resources.

Figure 2: Auxiliary fields to control measured, indicated, and inferred resources.

Figure 1: Auxiliary fields to control tonnages of different lithotypes.

Figure 2: Auxiliary fields to control measured, indicated, and inferred resources.

Figure 1: Auxiliary fields to control tonnages of different lithotypes.

[Previous image](#)

[Next image](#)

Importation and Setup

Whatever is the variable being modeled, the columns created with a similar purpose should be assigned to the field type "Sum", during the importation, as shown in Figure 3.

On the interface, the user will need to insert General and Destinations parameters in order to enable the Sum tab (Figure 4). The next step is to define the limits to be imposed to each variable, to which destinations, and to which period intervals.

Figure 3: Importation screen where variables to be controlled through sums are properly assigned to the field type "Sum".

Figure 4: Illustrates the interface and options available.

Figure 4: Sum interface

Figure 3: Importation screen

Figure 4: Sum interface

Figure 3: Importation screen

[Previous image](#)

[Next image](#)

Mining Fronts

Control your Mining Fronts

The 'mining fronts' approach, as shown in Figure 1 and 2, is a good way to refine results, control regions, and understand which are the best results considering different amounts extracted from a specific place. Using this methodology, you have the possibility to categorize masses by depth, a specific coordinate interval, even into sectors based on into 360 degrees vision, of your project.

The general idea is pretty similar to what is used to define material types. The first step is to identify the area which you want to control. Then create an additional field regarding the mass on that region and assign it as a sum constraint while importing it. At last, go to the sum tab and control the minimum or maximum amount at the process or dump destinations.

Note that by using this feature you are able to control what, how much, and when the blocks should be mined according to what you want to analyze. This concept is also very useful in the context of mine design, although it increases the complexity while compared with the use of force/restrict mining surfaces, it can also guide solutions, geometrically speaking. It is important to remember that the sum constraint has a high priority parameter in the algorithm and it can also influence the other inputs based on the hierarchy order.

Figure 2: Mining Front 2

Figure 1: Mining Front 1

Figure 2: Mining Front 2

Figure 1: Mining Front 1

[Previous image](#)

[Next image](#)

[Practical example](#)

Identify and define your regions as you wish. For instance, use this excel file, place the z from the validation data scenario, calculate the elevation difference with the topography, filter the region which has a 0 result, and create a dispersion chart to identify the final pit. Then, take the coordinates/indexes, which will be the limits of your mining fronts, in this case, the boundary between the mining fronts was at the index 35. Figures 3 and 4 have some visual information to help you.

Figure 4: Identify the regions

Figure 3: Filter the values

Figure 4: Identify the regions

Figure 3: Filter the values

[Previous image](#)

[Next image](#)

Calculate the tonnage of your mining front in an additional field and import it as a sum parameter. In this case, the mining front 1 is in the region above or equal to the limit chosen and the mining front 2 is what is below it, as seen in Figure 4.

Use the sum tab to control the Mining Front of each destination. Play with the minimum or maximum that should be extracted from each mining front and understand how the results change. The scenarios, that are shown in Figures 5 to 9, used the production, geometric and stockpiling parameters from the Data Validation page.

Download this modified Marvin Deposit file and play on your own.

Figure 9: Results of a minimum of 7Mt in Mining Front 1

Figure 5: Using the Sum Tab to control the destination

Figure 6: Mining front 1 with minimum 7M

Figure 7: Results of a minimum of 9Mt in Mining Front 1

Figure 8: Mining front 1 with minimum 7M

Figure 9: Results of a minimum of 7Mt in Mining Front 1

Figure 5: Using the Sum Tab to control the destination

[Previous image](#)

[Next image](#)

Stochastic Models

Stochastic simulations requires equiprobable models to consider uncertainties related to geological aspects, such as grade and/or volume of ore.

While single scenarios of distinct models are run separately, a stochastic scenario consist of obeying all the single scenarios at once.

This is achieved through an adapted resource block model that contains equiprobable values for a given set of variables containing a certain level of uncertainty.

As a consequence, MiningMath produces reports with the risk-profile of indicators presenting the minimum, maximum, expected, and percentiles P10 and P90 (these are threshold values, indicating that 10% of the indicators are below the P10 and 90% of the indicators are below the P90). Fig. 1 and 2 depicts the graphs for the NPV and cumulative NPV respectively.

Maximum expected NPV for period 2

Minimum NPV expected for Period 2

P90 NPV for Period 3. 90% of possible expected values for Period 3 are below this point.

P10 NPV for Period 3. 10% of possible expected values for Period 3 are below this point.

Expected NPV for Period 6.

Fig 1: Report on NPV for stochastic model.

Maximum expected Cumulative NPV for Period 2.

Minimum Cumulative NPV expected for Period 3

P90 Cumulative NPV for Period 4. 90% of possible expected values for Period 3 are below this point.

Expected Cumulative NPV for Period 5.

Fig 2: Report on cumulative NPV for stochastic model.

The purpose of this page is to briefly explain how to import data and manage stochastic constraints using MiningMath.

Scope

Stochastic mine planning is a wide topic and relates to long academic debates beyond the scope of this technical material. Please, check [here](#), and especially [here](#), for a deeper understanding of stochastic mine planning.

Formatting Uncertain Fields

Uncertain-fields are those which might vary from simulation to simulation. By definition stochastic models have uncertain fields. Typically, grade fields contain uncertain information. Therefore, the user will need to format each equiprobable possibility in a specific way: name each uncertain column as the same adding {#} (where # is a number from 1 up to n). The list below highlight how grade headers should look like for example:

* Copper {1}

* Copper {2}

* Copper {3}

* Copper {4}

* Copper {5}

Note that the grade information will influence the economic values for the processing stream. Therefore, the user will need to calculate the Economic Values for each possible grade information, as highlighted in Figure 3. This figure illustrates parts of a simulated copper deposit that can be downloaded [here](#).

Figure 3: Stochastic values for copper grade (green) and respective process (blue) for 20 simulations.

Stochastic Constraints

Once you import your stochastic block model, the tabs Average and Sum will allow for constraints both on:

1. Expected values to control the averages over all simulations.

These constraints will guarantee that, in average, the indicators will be within the defined ranges. For example, take Expected Min = 0.60 and Expected Max = 0.65 for a certain constraint. If there are 3 simulations returning 0.59, 0.62 and 0.65, the average is 0.62, so this is within the range defined.

Fig 4: Example to control the average of all simulations. This option is only available when databases containing stochastic data are imported.

2. All simulations to guarantee that each one of them respect certain criteria

These constraints control the variability, or the spread, of the results to be within a certain acceptable range. Let's take an example where such a range has Min = 0.60 and Max = 0.65, and again three simulations returning 0.59, 0.62, and 0.65. In this case the solution will be penalized by the optimizer, as $0.59 < 0.60$. Learn more about penalized solutions [here](#).

Fig 5: Example to control all simulations individually. This option is only available when databases containing stochastic data are imported.

Stochastic optimization is an optimized way to combine all these modelled uncertainties into one schedule that maximizes the Expected NPV of the project.

Violated constraints

After executing the optimization with constraints for simulated indicators, it is possible that such constraints will not be respected due to some infeasibility in the problem (more about infeasibilities [here](#)).

MiningMath will try to solve any violated constraints following the hierarchy order depicted in Fig 6. Stochastic constraining are average and sum constraints. They have higher priority than any NPV improvements and time limits imposed by the user.

Figure 6: constraints hierarchy order.

Time Limit

It is possible to indicate a time limit in hours before running a scenario in the "Run" tab as depicted in Fig. 1. The time limit is defined in hours due to the usual complexity of mining projects and by the fact that MiningMath will always try to deliver a reasonable solution.

Figure1: Time limit option in the interface

How does it work?

MiningMath is built via a global and interactive algorithm. It solves the entire mining optimization after formulating a global mathematical model. The result of such optimization might deliver a solution with room for improvement, due to necessary approximations for solving complex non-linear restrictions, such as the geometric ones, or due to infeasibilities identified in the problem's restrictions. In turn, if an improvement is possible, another iteration of the global algorithm is prepared and executed.

Therefore, in order to deliver any solution, the whole mining problem needs to be solved at least once, making a more fine-grained time limit (i.e. seconds or minutes) not possible to be set. In other words, the time limit is evaluated before each iteration of a global optimization that executes multiple times as depicted in Fig. 2.

Fig 2: Illustration of when time limit is evaluated. From steps 1 to 4.

The algorithm is designed in such a way that it is able to adjust subsequent iterations once it has identified that the time limit becomes restricted. However, it is important to highlight two aspects of such adjustment:

1. It will not interrupt the current iteration of the algorithm. Hence, while it is expected that this adjustment will help the execution to achieve the desired time limit, it is still possible that it will take more than what was defined.
2. Once an adjustment is made, a different problem will be defined and consequently new solutions will be explored. Thus, while unlikely, there is a chance that solutions will end up better than those unrestricted in relation to time. Therefore, despite not being implemented for this purpose, the time limit might be used to find more diverse solutions. For instance, you might build decision trees with different time limits. Even if better results are not obtained, fast solutions will still give you a quicker assessment of your project.

Academy

Articles

* Teaching

* Press Releases

Teaching

Training Presentation

Previous image

Next image

Assignment

John is a mine planning engineer working on Marvin deposit. He's conducting the long-term production scheduling plan and is concerned on what would be the best scenario to optimize Marvin's operations.

The mine has some constraints, such as slope angles defined block-by-block and a default value of 45 degrees for blocks missing information. The total movement is limited to 60 Mt/yr.

Part 1

* Format, economic values, and import this Marvin data. Save it as MMCI_YourName_Marvin.

* Choose your own economic parameters and fill the Table 1b to simulate a process B.

Remember air blocks!

Remember that if you used spreadsheet software to remove the Air Blocks, you might need to join all data in a unique column. A great tip on this is to join the columns by using "&","&":

=X&","&Y&","&Z&","&Cu&","&Au&","&Density,Slope

Figure 1: Block model format

Part 2

Use the following steps to evaluate your project:

1. Validate your data
2. Constraints Validation
3. Integrated workflow

Now that you know MiningMath is able to easily run multiple scenarios, you should play with the remaining parameters, evaluate the results and bring a proposal for the board. You should:

1. Download Screencast O-Matic or equivalent screen recording software.
2. Prepare a presentation (ppt or docx), which could be structured by answering the following questions:
 - a) What is the goal of your evaluation?
 - b) How did you use MiningMath to achieve your goal? Which were the features used? What was the methodology used to get to your results
 - c) Why have you got such results?
3. Record it as if you were reporting your analysis to the board. Upload it on YouTube as public. The video should be limited to 5 minutes.
4. Send the link, materials, and the Marvin data used to your teacher.

Questions

- * What are the main resources of MiningMath viewer and which are the output files?
- * Could market changes be considered based on different ore selling prices throughout time? If yes, how?
- * Is it possible to forbid the optimization to access some specific region? If yes, which feature could be used?
- * Which geometric features could be used to get results accordingly with the operational needs?
- * What are the benefits of building decision trees?
- * What are the main differences between Lerchs-Grossmann/Pseudoflow and DBS methodologies?
- * What is the importance of considering a discounted cash flow on the optimization decision-making process? Does the value of money through time impact the mining sequence?
- * Could the optimization handle multiple destinations? Over 2 processing plants? One stockpile for each with maximum capacity?
- * Could production ramp-ups be part of the optimization constraints?
- * Is it possible to control the average of a given variable, such as grade, haulage distance and etc.? How many properties can be controlled? Is it possible to change them over the mine's life?
- * How does the cut-off grade policy work on MiningMath?
- * What are the main validations before running a scenario?
- * Can MiningMath run more than one scenario simultaneously?
- * Which are the 3 main suggestions of the integrated workflow?
- * Is it possible to generate a short term planning using MiningMath?

Press Releases

MiningMath software name MiningMath was previously name SimSched. Some news might have been published with this name. × Dismiss this alert.

Strategic optimization

Is mining an ordinary business?

November 22, 2019.

"Mining is a business. This may sound obvious. But when one tries to look at how we operate that becomes much less evident. Are we really looking at what would be important for any business? I've had this discussions throughout my career in mining and I must say in many cases technical people, i.e. geologists, mining engineers, processing engineers - those who run the mines - do not look at business metrics. At all!!! Let's look at a miner. "Let's move tonnes and increase equipment utilization", one would say! This person will see the efficiency improvements in higher utilization. Does it help increase shareholder returns? No, not really. It has nothing to do with it."

Alexey Tsoy, LinkedIn Article

Startup wants to reduce dam risks

March 21, 2019

"The last two tragedies that desolated Brazil, leaving hundreds of deaths and miles of environmental devastation, from Brumadinho and Mariana, raised a great question: it is possible to reduce the risks of tailing dams and to guarantee everyone's safety in an efficient and economically viable way?

MiningMath, startup from Belo Horizonte, says yes. The company has created a software, SimSched, which, through the union of modern programming, and data science, allows the combination of any variables from a mining project to generate analysis, hypothesis and possible results. The objective is to contribute to improve the decision-making processes in the mining companies, in order to effectively consider the economic, social and environmental aspects of the business."

Diário do Comércio · Gira Betim

Startups from SEED visit the Mining Hub

March 18, 2019

"Five Startups of the Minas Gerais Government Accelerator (SEED - Startups and Entrepreneurship Ecosystem Development) from several segments, with solutions for business areas, visited last Tuesday (March 12th) the Mining Hub in Belo Horizonte (MG), to strengthen the relationship and exchange experiences on the market.

Among the startups were: MiningMath, which markets a solution that supports decision making at a strategic level specific to the mining industry, from designing a project to optimizing the value chain during the development phases; Recrutamento Inteligente ("Intelligent Recruitment"), which facilitates the management of intellectual capital; The Mindset, which acts in the management of mental health professionals; Cargo Sapiens, which offers management and compliance solution for international logistics; VG Resíduos, which connects producers and consumers of waste."

Portal da Mineração

Startup develops software that facilitates data management of deposits

December, 2018

"MiningMath has developed mine data management software that promises to reduce costs and anticipate the environmental consequences of mining. The tool crosses and organizes data from different departments of a mining company to generate scenarios with the possible geological, environmental, economic and social impacts of an action in a given location."

DCI

From left to right: the engineer Matheus Ulhoa and the partners of MiningMath, Fabrício Ceolin and Alexandre Marinho (MiningMath).

Industry 4.0 increasingly present in areas of the primary sector

November 6, 2018

"With the advent of industry 4.0, traditional and grassroots activities such as mining start to gain new ground in Brazil and the world. The use of innovative technologies such as data science, internet of things and optimization of processes breathe new life into the industry, promoting added value gain to the minerals extracted in the country, such as iron ore, whose largest production is concentrated in Minas Gerais.

Proof of this is that mining startup MiningMath recently won first place at the MineTech Mining Solutions Technical Challenges Challenge with an innovative technology to simulate scenarios to support strategic mining decisions using modern techniques of programming."

Diário do Comercio

Event in the Capital discussed a more sustainable performance, besides the optimization of processes and improvements in the performance of the sector (Vale Agency).

Mining Software developed by UFMG Alumni is awarded in competition in Russia

October 25, 2018

"The software combines variables ranging from geological aspects to economic data and legal, environmental and social constraints.

An innovative technology designed to simulate scenarios to support strategic decisions in the mining field, developed by former students of UFMG, Alexandre Marinho and Fabrício Ceolin, won the first place in the MineTech: Technical Challenges and Mining Solutions Challenge in early October, held in Moscow during the 14th Russia Exploration and Mining Forum."

UFMG · Mining.com · Hoje em Dia · SIMI · Notícias de Mineração · FUNDEP

Fabrício Ceolin (left) and Alexandre Marinho (right) are the founders of the MiningMath startup (Personal Files).

Alexey Tsoy to present at the MiningMath 2018 Creating Value in Mining Conference

October 16, 2018

CSA Global Principal Consultant-Corporate, Alexey Tsoy will present at the upcoming MiningMath Creating Value in Mining; Strategy Optimisation through Data Science Conference on November 6 2018 at the Museum of Mines and Metals in Belo Horizonte, Brazil.

Alexey will present on Strategic Schedule Optimisation.

Alexey Tsoy to present at the MiningMath 2018 Creating Value in Mining Conference - CSA Global Winner of the 2018 MineTech Competition

October 16, 2018

Alexey Tsoy, Principal Consultant - Corporate and Business Development at CSA Global, is the winner of the 2018 MineTech Competition: 2nd Mining Technical Challenges and Solutions Competition's at the 14th Russian Mining and Exploration Forum held between 2-4 October in Moscow, Russia.

"Alexey presented on Strategic Schedule Optimisation; forming part of this year's MINEX conference 'Building Up Innovative Excellence in Mining and Exploration' [...]. The proposed approach is based on a software called SimSched Direct Block Scheduling (DBS) [...]. The software applies a simplistic approach where a block model is optimized on economic values assigned to each block in pre-processing. The simplicity allows a great degree of flexibility on assigning the economic values. Moreover it allows definition of variable costs based on time, residence time, or indeed any other parameter that can be calculated and limited in time."

Congratulations to Alexey Tsoy: Winner of the 2018 MineTech Competition - CSA Global

"The software that could revolutionize mine planning"

April 9, 2018

This publication states that the traditional planning includes the discount rate, production limits, blending and other technique-economical variables during advanced stages of planning and SimSched brings the possibility to reduce the number of iterations for planning.

"A unique and flexible tool for planning, which currently doesn't exist in the industry. " - Fabián Lemus, Senior Long-term Mine Planning Engineer.

The Scheduling The Sequences (STS) project, conducted by Antofagasta Minerals, includes a co-development of SimSched DBS, where major tests will focus on Minera Centinela, a multi-pit project with multiple processing streams that might benefit from an integrated plan. This will provide flexibility to analyze and evaluate development scenarios and the inclusion of new projects.

Access the original content.

Integrated approaches in the industry

How digital innovation can improve mining productivity

With profits down, miners are focused on improving their productivity. Digital innovation could provide a breakthrough.

[Read more.](#)

Digital Transformation-The future of Mining

In a challenging market, the digital transformation of mining companies has become a business imperative-leveraging technology to improve processes aligned to value.

[Read more.](#)

A tool for these times

Considering the current resurgence in commodity prices, every mining company should concentrate on strategy optimization to ensure that operations move from a cost-focused mindset to one centred on value maximization in order to reap the benefits of the upturn in the mining industry.

[Read more.](#)

Waste Dump Sequencing with SimSched

SimSched Direct Block Scheduler (DBS) is an open pit optimization package that selects maximum NPV pit shell while generating a mining schedule. [Read more.](#)

Innovation and Technology to Improve Open Pit Mine Plan and Design Optimisation

In the mining the change in technology, i.e., processes and software, does not happen fast but it takes its time, around 10 to 15 years as miners are very. (LinkedIn article)

Academic Partners

Become a partner

Mine Optimizations and SimSched DBS is a friend of yours now. SimSched DBS is the most powerful tool to achieve and compare NPV results of the pit. [Read more.](#)

Publications

2018 Sensitivity Analysis applied to operational parameters, IFG, Brazil (Portuguese)

2017 NPV Analysis as a function of the discount rate and cost of rehandling implementing SimSched DBS to open pit mining, Universidad Nacional de Colombia, Colombia (Spanish)

Courses

2017 Universidad del Azuay

2017 Universidad de San Luis, Argentina (Spanish)

We are on LinkedIn

[Join the DBS group!](#)

Source: What is Direct Block Scheduling?

During decades, the mining industry has dealt with Mine Planning as a step-by-step process. This traditional technology has been established in an intelligent manner in face of the technological limitations of that time.

Source: Lerchs-Grossmann, rest in peace.

We'd like to share the news that the main limitations of open pit mine plan and design based on Lerchs-Grossmann have been surpassed by a novel and innovative technology.

Spanish | Portuguese

Source: Is Pit Optimization Still Necessary?

The concept of Pit Optimization is becoming obsolete. Here we bring a comparison showing the variety of outputs you can obtain from a single optimization.

Spanish | Portuguese

Common Issues

Articles

- * Floating-point numbers
- * Get Started
- * Practice First, Optimizing Scenarios, Decision Trees, Theory Behind, Interface overview
- * Economic Values
- * Formatted example
- * Importing Data
- * Progress Bar
- * Constraints Validation
- * Integrated Workflow, Best Case, Exploratory Analysis, Schedule Optimization, Short-term Planning
- * CSV Formatting Issues
- * Warning Messages

Floating-point numbers

Same scenario, different results?

Is it possible to find different results for the same scenario running in different computers. Algorithms based on Mixed Integer Linear Programming (MILP) depend on third-party solvers and their results may differ in terms of floating-point precision from hardware to hardware.

Given that MILP is based on multiple LP executions, precision differences may accumulate over the sequence of operations performed with floating-point numbers. It's expected that results may differ, but they should be equivalent in terms of NPV. If the physical results are too different, this means this mine has the flexibility to operate in both ways without big impacts in NPV.

For an in-depth explanation on floating-point arithmetic from a computer science perspective please see [here](#).

Get Started

Get Started

The main issues faced here are based on the System Requirements, shown in Error Warning, which often are forgotten to try to check them again.

There were improvements in the importation and exportation of coordinates after version 2.0, which were required due to the implementation of the viewer. Therefore, if you have run anything on previous versions, it is required to import your block model and run it again in the current version, available [here](#).

Another common issue is related to the Licensing System. MiningMath relies on online activation based on the internet connection, or through an identification code from your hardware as a contingency. Therefore, you can solve your issue by understanding the main information disclosed on the "licensing screen", from Licensing Error, in the following order:

Figure 1: Error Warning

1. Identify your error number/message.
2. Get your hardware identification code (Host ID), by the two following options:
 - o Copy the text disclosed at the "Informations for support", if it is available for you.
 - o Execute this procedure bellow, explained in this video.
3. Send us the error number/message generated and the identification (Host ID) by filling this form.

Figure 2: Licensing Error

Note: If your error number is -3001 get your solution at [this page](#).

Note 2: The revoke license procedure, available only in commercial licenses, started after version v2.0.24. Therefore, make sure to have an updated version before revoking it on your computer and activate it in another one.

Practice First, Optimizing Scenarios, Decision Trees, Theory Behind, Interface overview

The pre-defined scenarios with Marvin deposit can usually appear with Red Warnings. This means that within the installation process MiningMath could not find a folder to place it. Therefore, to run it you just have to click in the one scenario, choose where the files should be and save them. It is important to mention that Marvin block model is disclosed at the link bellow, thus if lost it or want to import it again feel free to do so.

DOWNLOAD MARVIN DEPOSIT

Figure 3: Recent projects.

While Running the pre-defined scenarios, you can already face some errors which happens when MiningMath tries to communicate with Excel and fails. The reason why it happens is that there are additional windows (like a login screen, an activation failure, etc.) being opened before the worksheet, which is interfering in MiningMath reports generation.

To find out, close the Excel instances completely and reopen it. If an additional window appears, before the worksheet, try to solve what this message asks so that it would fix this interference.

If you have more than one Excel version in place or it's impossible to use due to a lack of license, it is recommended to uninstall and use another platform, to open the .csv reports generated and do your own reports.

Figure 2: Pass through this figure sequence to identify your warning.

Figure 1: Pass through this figure sequence to identify your warning.

Figure 2: Pass through this figure sequence to identify your warning.

Figure 1: Pass through this figure sequence to identify your warning.

Previous image

Next image

Formatted example

The Figure aside provides an example of a block model prepared and ready to be imported into MiningMath using the Marvin block model - based on data available in the MineLib1 repository.

Optional fields inside the block model are listed below:

- * Density (t/m3).
- * Slope angle (degrees).
- * Process recoveries (values from 0 to 1).

Figure 1: Block model example.

Importing Data

There are a lot of reports coming from this page, the first trick here is to check the tooltips when any question appears. A common issue occurs when we assign the variables to the wrong field type. The second trending mistake happens when we forgot to place the origin accordingly with the Mining Packages previously used, which generates surfaces totally disconnected from your original data. To solve these problems just import your block model again. The last one is related to the "Next" button which can be explained here.

If you are working with Marvin Deposit along with MiningMath, be aware that the inclusion of new fields is allowed, as long as there is no inclusion or exclusion of blocks in the file, and the SLOPE, DENSITY, and INDEX or COORDINATES fields are not modified. The error message displayed at the aside occurs when modifications are detected in the basic structure of the standard Marvin deposit. Therefore, use the original CSV file disclosed at item 2 of this page.

Figure 2: Pass through this figure sequence to identify your warning.

Figure 1: Pass through this figure sequence to identify your warning.

Figure 2: Pass through this figure sequence to identify your warning.

Figure 1: Pass through this figure sequence to identify your warning.

Previous image

Next image

Progress Bar

During some steps, such as Data Validation, the main question should be about the runtime and the progress bar. It is worth mention that this is a rough guess, since math programming isn't something very predictable. The pre-processing steps, in which the algorithm eliminates the useless material, might keep it stuck in the initial percentage (2%, 4%, etc) for a while, but after that, the optimization can get faster.

MiningMath can virtually handle any model size. It has successfully run models from clients beyond 10M blocks without reblocking, which might take a few hours to finish. The runtime is directly proportional to the number of blocks, destinations, periods, constraints in use, and variables imported. Therefore, the combination of multiple aspects, are directly related to the complexity of the deposit.

Try to check if there is any "floating blocks" without any connection with the model topography. These regions might affect the optimization, thus, eliminate them so that MiningMath can work well.

Figure 1: Floating blocks.

Integrated Workflow, Best Case, Exploratory Analysis, Schedule Optimization, Short-term Planning

By running these scenarios you are already mastering MiningMath and the main use of surfaces and refinements. The main suggestions to enhance your workflow here are:

- * Avoid using the minimum of a sum constraint equal processing capacity, which could cause some complexity.
- * Take special attention on surfaces so that a force mining does not surpass the limits of a restrict mining area.
- * Try to avoid the use of force+restrict mining in the intermediate intervals of your timeframe.

Sometimes the steps suggested on the integrated workflow, after the validation, could be a huge challenge to block models which has a high complexity in place. Therefore, you can try an intermediary step by splitting the mass obtained in the last executions on 2 or 3 periods, so that you could understand if everything is running properly and how much time such optimization is taking on average. It is important to be aware that geometric constraints as minimum widths and vertical rate of advance increase the complexity of the algorithm since they are 3-dimensional non-linear parameters. Beyond that, the number of periods, destinations, and constraint in place also influence such results. Thus, to identify what are the main challenges to the runtime read this page here.

CSV Formatting Issues

The main focus here is on the requirements. Try to pay attention to the header names, which should be short (less than 13 characters) with no special symbols*. This nomenclature recommendation is also applied to the name of folders and files. Beyond that avoid a high number of subfolders to place your files, which can interfere in the viewer while loading it.

The data format is also a common issue, always remember to remove the air blocks, which could also be checked on the viewer. Note that a CSV file (Comma Separated Value), usually contains just one column* with all the information, when it is opened on excel. Therefore, if you are facing issues to import it, check if it has a similar format as the Marvin_Strategy_Optimization.csv disclosed in the installation process.

*Some Mining Packages export data with letters and values between quotation marks(""), check if this kind of format is interfering in the importation process.

**If you need to join columns a useful tip used in Excel is the formula: =A1&" "&B1&" "&C1

Figure 2: Pass through this figure sequence to identify your warning.

Figure 1: Pass through this figure sequence to identify your warning.

Figure 2: Pass through this figure sequence to identify your warning.

Figure 1: Pass through this figure sequence to identify your warning.

Previous image

Next image

Warning Messages

100 periods

The "100 periods" warning message (Figure 1) aims to prevent the user from spending hours with the wrong scenario setup. This is a hint that some parameters might be inconsistent with the information from the block model.

Common examples are typos, where the values inputted are in a lower magnitude order than they should be, as follows:

- * Production should be 10 Mt/year.
- * The user mistyped 10 kt/year.
- * Other parameters imported range from 10 k to 10 M.
- * The user mistyped 1k as a maximum limit for Other Constraints.

Before each scenario optimization, MiningMath roughly validates the parameters inputted. The math consists of comparing production limits, blending, and other parameters with simplistic calculations based on the amount of material available. In case values inputted are in a lower magnitude order than expected, the life of mine will likely be over 100 years, which triggers the warning message.

Figure 1: Warning message "The production capacities informed may force a mine schedule with more than 100 periods. Do you want to continue anyway?"

Geometry

Articles

- * Slope Angles
- * Geometric Constraints
- * Widths and lengths
- * Vertical Rate
- * Force Mining
- * Restrict Mining
- * Combining Force and Restrict
- * Creating Surfaces
- * Validating Surfaces
- * Surfaces as a Guide
- * Surface Constraints

Slope Angles

How slope angles are generated

The slope angles (Figure 1) are one of the main concerns in the mining industry due to security, and all the operational parameters that it can influence. Therefore, this is one of the most important parameters when considering the constraints hierarchy. Usually, it can vary based on the time frames (short/long-term), rock type, lithologies, mine sector, depth, geotechnical domains, etc. Thus, it is important to have these assumptions clear to use this parameter wisely following what you are expecting to your project.

The traditional workflow for open-pit optimization, design, and production schedule may incur in a discrepancy between parameters used for the optimization stage and the design parameters. Ramp design and positioning is a common example on a step that will affect the Overall Slope Angle (OSA). This often brings the need for an iterative process of re-optimizing the same scenario based on the pit designed.

MiningMath works with "surface-constrained production scheduling", defining surfaces that describe the group of blocks that should be mined, or not, instead of the "block precedence" method. By this approach each surface has a feasible solution, considering productions required, and each point could be placed anywhere along the Z-axis. Therefore, this flexibility allows the elevation to be above, below, or matching a block's centroid, which ensures MiningMath's algorithm to control the OSA precisely, with no error, which would have an even higher impact on transition zones.

The following video illustrates how MiningMath's algorithm handles the definition of slope angles, especially in transition zones.

Figure 1: Slope angles definition

Video 1: Discover how slope angles are generated.

Note

Technologies based on Lerchs-Grossmann/Pseudoflow approaches also present a broadly-known "slope error" which is the difference between the optimization input parameters for OSA and what is measured from output pit shells. This is a consequence of the "block precedence" methodology, in which calculations are based on connecting the block's centroids, which might not be achievable when considering the block model dimensions. Therefore, the decision is always on mining or not a block entirely, which affects the OSA and might create additional difficulty to generate a pit shell that matches the designed pit.

Setting up Slope angles

Block by Block field

This field represents the column(s) that has/have been assigned to the slope during the importation, for varying slope angles on each block. This allows a high level of flexibility to use any specific criteria. These possibilities can also comprise bi-dimensional and tri-dimensional variations, beyond linear and non-linear functions (Figure 2).

Default values

On the interface, you have the option of selecting the information field, which will be the main rule for variable slope angles or to select <none> to define a constant value that will be used for the entire model. The "Default value" settled is also when you use a field that has missing information in the column (Figure 3).

Figure 3: Slopes default value assumption

Figure 2: Slopes defined block-by-block inside the block model

Figure 3: Slopes default value assumption

Figure 2: Slopes defined block-by-block inside the block model

[Previous image](#)

[Next image](#)

Slope angles in the Short-term

Short-term Planning is a great opportunity to use the same platform that the strategic mine planning team is using, which allows the company to enhance the adherence/reconciliation of projects by choosing a surface and using it as force and restrict mining to refine everything inside it.

The approach could consider a surface already designed with ramps or any other from MiningMath which respects the Overall Slope Angle (Figure 4) in the time frame required, so that you could use the steeper Bench Face Angles based on the operational parameters of blasting. To use such a feature apply the item 2.2. suggestion and play with different angles accordingly with your project capabilities, this methodology allows you to give flexibility to the algorithm in the constraints hierarchy to find better results.

Figure 4: Operational Bench Face Angle and Overall Slope Angle difference.

Geometric Constraints

Introduction

For a mining project, the mine planner needs to dimension each unit operation to assign which set of equipment will best suit the existing conditions. On MiningMath, the operational parameters are constraints to the objective function, instead of being applied after pit optimization. This approach allows for solutions that follow either some operational criteria and maximizes NPV, resulting in a better use of the data and identifying opportunities that could be missed by an approach with manual steps and arbitrary assumptions.

Geometries & the User Interface

The Geometric tab is the place to set minimum mining & bottom widths, mining length and vertical rate of advance, whose values are applicable to every period. The user can also use surfaces to define operational constraints in compliance with period ranges, which can limit, force or achieve an exact shape, based on the constraints hierarchy.

Minimum Widths

Figure 1 shows the fields to define widths:

- * Mining Width: Distance from a pit to another.

- * Bottom Width: Bottom minimum area.

Currently, MiningMath does not mine partial blocks. As a consequence, the software will round up any widths to cover the next integer block.

Figure 1: Minimum widths

Vertical rate of advance (VR)

Figure 2 shows the field to define the VR

As said before, the MiningMath will round up the VR to cover the next integer block.

Figure 2: Vertical rate of advance

Mining length (ML)

Figure 3 shows the field to define the ML. Currently, this is only available in the insider version.

Figure 3: Mining length.

Figures 4-7 show a simplistic meaning of each width available and the vertical rate of advance.

Figure 7: Mining length.

Figure 4: Minimum bottom width.

Figure 5: Minimum mining width.

Figure 6: Vertical rate of advance.

Figure 7: Mining length.

Figure 4: Minimum bottom width.

[Previous image](#)

[Next image](#)

Surface Mining Constraints

For each period range, the user can consider:

- * 1 force mining surface.
- * 1 restrict mining surface.

Each surface file is valid from period A up to final of period B, as highlighted in Figure 8.

Figure 8: Surface mining limits: forcing and restrict mining.

Practical Overview: How to play with operational parameters

The following video shows how the variation of operational constraints impacts your solution and how you can take advantage of these parameters to find results more closer to the reality.

Video 1: Operational constraints

Widths and lengths

On MiningMath, the widths and lengths are constraints within the objective function, which means that they are regarded in the optimization instead of being considered just on the pit design stage, when roads, access are drawn to conceive an operational feasible to the pit.

The definition of minimum widths is a pretty useful feature to obtain operational results and play with different geometries accordingly with the project requirements. It is important to understand that they are very complex parameters to be respected while considering 3-dimensional non-linear models, which also influence the runtime of each scenario. Therefore, it is not possible to always guarantee that all minimum widths are respected due to the deposit geometry and also the constraints hierarchy. Testing different values is a great strategy to identify opportunities that could bring the best mining sequence and NPV.

Types of widths and lengths

Bottom widths

Bottom widths (BW) are the minimum horizontal distance, on the lowest floor of the pit, as seen in Figure 1. It is required to allow mining operations based on the equipment sizing. This parameter is considered the same for all periods and is related to adjacent slopes and applicable in other areas regarded as pit bottoms.

Figure 1: Bottom width area in subsequent periods

Mining widths

Mining widths (MW) are the minimum horizontal distance that should be respected from a pit to another in every period, which means that this is the horizontal distance between the walls of two surfaces that belonged to consecutive periods, as shown in Figure 2.

This feature is more complex than the minimum bottom width since it would drastically change the pit shapes to identify the best regions. Note that wider values provide greater mining fronts and also better designs to nested pits, pushbacks, schedules, or any other result that you are looking for.

Figure 2: Mining width in subsequent periods

Mining lengths

The Mining Length (ML) represents a minimum distance that must exist between at least two points amidst the walls of surfaces among two consecutive mining periods. This distance is already respected for any values smaller or equal to the MW. Thus, this parameter extends such distance between any two points for a value greater than MW. Figure 3 depicts an example

Figure 3: Example of MW and ML.

Identifying geometric parameters

Figure 4 shows a section view of the McLaughlin deposit, where each color represents a given period. The horizontal arrows highlight the bottom and the mining width, while others identify the vertical advance. Mining lengths cannot be depicted in this 2D representation.

Figure 4: Bottom width, Mining width and vertical rate of advance.

Testing different widths and lengths

When adding more restrictions to a project, it is common for the net present value (NPV) to decrease as the project becomes more restricted. However, since geometric constraints are not linear, it is possible that the results may not follow this trend as new values are tested.

To conduct an in-depth analysis of the impact of widths and lengths, it is advisable to evaluate more flexible values first. This will provide an indication of the upper bound for the NPV for certain geometric constraints, even if the solution is not fully operational yet. These initial results can then be improved until they are feasible enough. The steps below detail this idea:

1. Create a scenario with an initial flexible width/length, such as a small bottom width or mining width.
2. Evaluate the variance in NPV compared to a scenario with no geometric constraints.
3. If the NPV is good enough, consider the following options:
 - a) Can it be implemented with the current fleet? If the current fleet is not adequate, can smaller trucks be acquired to travel on narrower roads?
 - b) If it is not possible to continue with this scenario, try to further tighten the geometric constraints. Run the scenario again and evaluate the variance in NPV.
4. If the NPV is not good enough, consider loosening the geometric constraints by a small extent and running a new scenario.

By following the steps 1 to 4, you can gain better insights on how geometric constraints impact your project as a whole. Note that some iterations between steps 3 and 4 might be necessary to achieve good results.

How are widths defined?

The widths inputted in the interface define a diameter (d) of a circle. As MiningMath does not mine partial blocks, the software will consider the block size as a reference to define whether d should be rounded up to the next integer multiple. The approximated circle results in a polygon, whose objective is to select centroids of adjacent cells. Then, MiningMath will assign the same elevation value to the selected cells to define mining surfaces.

Figure 5 shows, in sequence, an example of how the minimum width of 25 meters and possible mining length of 50 meters could be defined over a 10 x 10 meters grid (block's dimension in x and y direction).

Figure 5: How MiningMath defines the mining width and mining length.

Vertical Rate

Introduction

The vertical rate of advance (VR) or sinking rate is also considered within the objective function on MiningMath, which means that it will be used to deliver the best result considering constraints hierarchy, even though it has a low priority. This parameter is defined as the vertical distance mined on each period, which depends on the pit depth and equipment available.

It is important to understand that this is also a 3-dimensional non-linear model, which means that it is a complex parameter within the optimization. The VR works as an upper bound to avoid operationally unfeasible solutions. Therefore, testing different values is a great strategy to identify opportunities that could bring the best mining sequence and NPV.

The Mining Width (MW) is an important parameter when defining the Vertical Rate. The MW and VR, together, define volumes of material for each mining period. A reduced MW might create additional challenges for the algorithm to comply with the VR. Therefore, it is important to play with different values, especially when VR is not being fully respected. Figure 1 illustrates a McLaughlin profile where the vertical rate of advance can be observed. Sub-vertical advances are also contemplated by this constraint.

Video 1: Vertical Rate: Definition, Hierarchy of Constraints & Complexity

Figure 1: Bottom width, Mining width and vertical rate of advance.

Prioritizing it

The vertical rate of advance is one of the first constraints to be relaxed within MiningMath's Hierarchy of Constraints (read more). To ensure VR is at least closer to what you need, relax low-priority constraints manually. This way you will lead the algorithm to a more flexible scenario and a broader solution space, which may help it to find a feasible solution for the new set of constraints.

Besides, if the user aims to force a maximum vertical rate for a given period, it can be created a flat surface constraint regarding the achievable depth and input it as a Restrict Mining. Beyond that is important to notice that even the goal of achieving "process full" could result in a non-feasible solution while the VR is still respected, therefore, this feature is very sensitive to any other parameter.

Video 1: Vertical rates controlled by surface constraints

Force Mining

Introduction

Force mining, is used to deplete the material of the entire area (Figure 1) in a specific time-frame. This feature is responsible for making MiningMath achieve at least the surface inserted, which means that all the material inside its limits should be extracted, whether is ore or waste. Thus, this feature could be also understood as a minimum depth that should be mined at a specific timeframe.

Keep in mind that these surfaces might be adjusted during the optimization to respect the slope angles (Figure 2), which has a higher priority order on the algorithm, while the optimization is done. Therefore, more material can be mined either to correct the overall slope angle or to increase the NPV.

This functionality is commonly used to refine/keep the mining amount of a previous good surface in early periods, force a specific depth which the deposit should achieve, create custom advances, extract material to make a region available to allocate equipment, etc.

Figure 1: Golden arrows disclosing the areas that should be forced.

As MiningMath aims to assist the users to apply their project knowledge to guide the algorithm into the best decisions, that is why surfaces are one of the most important constraints hierarchy, which enables the implementation of custom geometries and operational parameters based on this smart hints imposed. Thus, forcing-surfaces might be the reason for disrespecting production limits, blending constraints, geometries, and so on, which require the user to be careful by using these functionalities. To sum up, the material above a Force Mining imported surface will be certainly mined until the specified period of time and what is below it will be mined only if the blocks respect all the other constraints and generate profitable results.

Figure 2: Force Mining surface and the slope angle adjust which could happen.

Practical example

The approach here was the attempt to deplete a Mining Front with high grade ores in the first period on the optimization, which considers the region which has IY higher than 35, inside the final pit of the Data Validation (Figure 3) and limited until the elevation 250. To build such surface the first step was to place the Z coordinates in this excel file, then use conditional functions to define these limits (Figure 4 and 5).

By having a surface suitable to use, a scenario of Exploratory Analysis was run using the set up of disclosed in Figure 6. Note that a scenario without Force Mining (Figure 7) was also run to have a good comparison.

As a result, the Force Mining surface totally mined along additional material to fulfill the processing capacity and also adjustments on the slope angles which generated the result disclosed in Figures 8 to 10. Besides, it is always useful to compare these scenarios with the ones that did not use this approach, which generates different sequences (Figure 11 and 12) and illustrates how powerful could be user assumptions to generate better results or explore possibilities.

Download the Mining_Front-FM.csv surface file and play on your own.

Figure 12: Surface from period 1 respecting Force Mining

Figure 3: Final pit surface and high-grade ores filtered.

Figure 4: Mining_Front-FM.csv context.

Figure 5: Force Mining Surface and high-grade ores.

Figure 6: Scenario Force Mining

Figure 7: Scenario without Force Mining

Figure 8: Mining sequence of the Force Mining scenario.

Figure 9: Scenario without Force Mining.

Figure 10: Force Mining Surface.

Figure 11: Surface001.csv from Force Mining optimization

Figure 12: Surface from period 1 respecting Force Mining

Figure 3: Final pit surface and high-grade ores filtered.

[Previous image](#)

[Next image](#)

[Restrict Mining](#)

[Introduction](#)

Restrict mining is used to prohibit access to any area (Figure 1) in a specific time-frame. This feature is responsible for making MiningMath ignore what is outside the surface inserted, which means that only the material above it is available to be extracted, whether is ore or waste. Thus, this feature could be also understood as a maximum depth that could be mined at a specific timeframe.

Keep in mind that these surfaces might be adjusted during the optimization to respect the slope angles (Figure 2), which has a higher priority order on the algorithm, while the optimization is done. Therefore, less material can be mined either to correct the overall slope angle or to increase the NPV, since mining, useless waste material does not increase the revenue.

This functionality is commonly used to refine/optimize the mining amount of a previous surface in any period, restrict depth to a specific value which the deposit could achieve, extract the best ore in custom advances. It locks/prohibits an area due to concession rights, environmental issues, or even due to an already built stockpile, waste dump or structure in general, etc.

Figure 1: Red X's disclosing the areas that should be restricted.

As MiningMath aims to assist the users to apply their project knowledge to guide the algorithm into the best decisions, that is why surfaces are one of the most important constraints hierarchy, which enables the implementation of custom geometries and operational parameters based on this smart hints imposed. To sum up, the material above a Restrict Mining imported surface will be available to be mined until the specified period of time and what is inside could be mined if the blocks respect all the constraints and generate profitable results. Regarding what is outside, it will not be mined whether profitable or not.

Figure 2: Restrict Mining surface and a possible pit shape respecting this constraint.

[Practical example](#)

The image aside illustrates an example as the standard scenario scn21-PriceUp-RampUp-Protection300 available on optimizing scenarios. It is possible to identify that in the free case (left), the pit limit advanced beyond the limiting surface on the west portion, defined by the higher elevations of the cells (equal to the topography). On the other hand, applying the limiting surface (right) to restrict mining until the last period, a new pit limit is obtained and obeying the constraint imposed by the surface, as shown in Figure 3.

Figure 3: Comparison between a free run and the use of a restrict mining scenario.

The second example is based on the attempt to extract the best material inside the Mining_Front-FM.csv, which mapped high-grade ores in the first period (Figure 4 to 6), as mentioned on this page. Using the suitable surface file, in a scenario of Exploratory Analysis was run using the set up disclosed in Figure 4. Note that a scenario without Force Mining (Figures 7 and 8) was also run to have a good comparison.

As a result, the Restrict Mining surface mined the best material within the area available (Figure 9), which generates different sequences (Figures 10 and 11) and illustrates how powerful could be user assumptions to generate suitable results or explore possibilities.

Download the Mining_Front-FM.csv surface file and play on your own.

Figure 11: Scenario without Restrict Mining.

Figure 4: Final pit surface and high-grade ores filtered.

Figure 5: Mining_Front-FM.csv context.

Figure 6: Restrict Mining Surface and high-grade ores.

Figure 7: Scenario with restrict mining

Figure 8: Scenario without restrict mining

Figure 9: Restrict Mining optimization.

Figure 10: Mining sequence of the Restrict Mining scenario.

Figure 11: Scenario without Restrict Mining.

Figure 4: Final pit surface and high-grade ores filtered.

[Previous image](#)

[Next image](#)

[Combining Force and Restrict](#)

[Introduction](#)

MiningMath allows the user to combine Force and Restrict mining by using surfaces on each field as you already read in the previous pages. These features allow us to use different arrangements due to concession rights, exchanging of land with adjacent mining companies, allocation of waste material inside exhausted areas, environmental issues, and so forth.

By using them together, the user can either reach the exact shape of a pit if you input the same surface as Force and Restrict mining at the same time frame (Figure 1), which has the highest priority constraint in the hierarchy order. It is also possible to optimize the material between surfaces if you add different surfaces in these two fields (Figure 2), which might be adjusted either to correct the overall slope angle or to increase the NPV, as mentioned before.

Figure 1: Using the same surface as Force and Restrict mining to reach an exact shape.

Based on these concepts, MiningMath allows you to export surfaces from the best scenario to a bigger mining package, design it with your mining package, create the grid of points, import the designed pit, and, finally, optimize and refine as much as you can by using smart constraints.

Therefore, the user has the advantage to control the results by guiding results accordingly with the project requirements.

Figure 2: Using different surfaces on Force and Restrict mining to optimize a volume within its limits.

Practical examples

Surfaces used simultaneously in both fields, force and restrict mining, are interpreted as follows:

- * Different surfaces: MiningMath will force mining according to the forcing-surface and will restrict mining according to the restricting-surface.

- * Same surface: MiningMath will achieve the same format proposed by the surface in use until the end of the time frame in which they have been applied.

Different surface example

This approach allows you to use different surfaces in the same time frame or split them accordingly to the goals that you want to achieve since this feature works close to what is shown in Figure 2 above.

Figure 3: Mining sequence between force and restrict mining surfaces.

In this example, it was used the same constraints were mentioned on the schedule optimization page. Besides, the surface 2 of this scenario was add on the force mining field in the second time frame, which means that by the end of the second period the mining should reach at least the shape imposed. The restricting surface used came from the constraints validation page and was added on the last interval, which means that the algorithm could not surpass this limit. Figure 3 discloses the optimized mining sequence volume between the surfaces inserted, and Figure 4 shows the set up of the scenario.

As result, the Figures 5 and 6 illustrate how the force mining has influenced the the firsts periods of the optimization, which mined more material due to its profitability.

Figure 7 disclose the constraint validation surface, used as restrict mining.

Figure 8 shows the final result regarding the force and restrict mining features, which respected the surface constraints and demonstrates the capability of these features to guide results.

Figure 4: Scenario setup.

Figure 8: Final results of the optimization.

Figure 5: Surface 2 from Schedule Optimization.

Figure 6: Force mining optimization of the 2 first years.

Figure 7: Surface 1 from Constraints Validation.

Figure 8: Final results of the optimization.

Figure 5: Surface 2 from Schedule Optimization.

[Previous image](#)

[Next image](#)

[Same surface example](#)

The following figure exemplifies how the user can take into account any pit design to make MiningMath iteratively produce more operational results, detailing a previous scenario.

In this case, the user needs to use the same surface in both fields, force and restrict mining, and during the same period of time to reach the exact shape of the designed surface.

Although the workflow on Figure 9 uses a designed pit, it is possible to use pits from previous scenarios as well, so that you can freeze good results and optimize further periods. Below are some examples of it:

- * Getting the same: Achieve the same final pit of a previous scenario.
- * Using the traditional approach: Define a pushback from 5 to 5 years.

Figure 9: Using a designed surface as force and restrict mining.

This powerful workflow allow a lot of flexibility so that the user can guide solutions based on insights and previous knowledge of the deposit. The concept is a pretty unique feature of MiningMath and such approach could be easily done by following the steps of creating surfaces and validating them on the footer below.

Creating Surfaces

Introduction

The surface files on MiningMath are a set of points (Figure 1), which are aligned with blocks centroids on X and Y axes (Figure 2). The easiest way to avoid any error message is by using a topography surface created by MiningMath on the Data Validation, for instance, and then manipulating only the Z coordinates.

Figure 2: The points aligned with the block centroids on the X and Y axes.

Figure 1: Point set file generated by MiningMath which corresponds to the topography surface.

Figure 2: The points aligned with the block centroids on the X and Y axes.

Figure 1: Point set file generated by MiningMath which corresponds to the topography surface.

Previous image

Next image

Usually, the designed surfaces on traditional mining CADs are based on contour/drawn lines and point triangularizations (Figures 3 and 4). Therefore, they are continuous figures which can not be recognized on MiningMath.

Figure 4: Point triangularization done to generate surfaces on the CAD of Mining Packages.

Figure 3: Common designed surface file generated by a Mining Package.

Figure 4: Point triangularization done to generate surfaces on the CAD of Mining Packages.

Figure 3: Common designed surface file generated by a Mining Package.

Previous image

Next image

Creating surfaces

Here, it's the step-by-step to create surfaces on MiningMath:

- * Import the TopographySurface.csv (Figure 5), which is a grid of points, in the CAD of your Mining Package. The CSV file mentioned might be obtained by validating scenarios or any other execution on MiningMath.

- * Manipulate only the Z coordinates and project them in a way that fits your needs. There are 3 main options at this step:

- o Use a polyline (Figure 6) to draw the region, select the points inside the polygon drew, and find the option on your Mining Package which allows you to place them at the elevation that you want.

- o Use your triangularized designed surface (Figure 7) by placing it on your CAD viewer along the TopographySurface.csv generated by MiningMath. Select the point set imported in your CAD and find the option on your Mining Package which allows you to project all points at the same elevation of your designed surface (Figure 8).

- o Using only Excel or any spreadsheet program. Open the TopographySurface.csv, filter the regions in X and Y that you want to change the elevation and manipulate them.

Figure 8: Place all the points at the same elevation of your designed surface.

Figure 5: Import the Topography surface generated by MiningMath at your Mining Package.

Figure 6: Draw a polyline and project the elevation inside it at the elevation that you wish.

Figure 7: Place your designed surface along with the topography surface generated by MiningMath at your Mining Package.

Figure 8: Place all the points at the same elevation of your designed surface.

Figure 5: Import the Topography surface generated by MiningMath at your Mining Package.

[Previous image](#)

[Next image](#)

Export the modified point set (Figure 9) as a CSV using a different name. Open the file on excel or the notepad to make sure that the header is correct (Figure 10) before importing on MiningMath since it is pretty common that these exported come along with meaningless information that can interfere with the importation.

Figure 9: Surface file which could be imported in MiningMath.

Figure 10: Header and disclosed data on a standard surface file of MiningMath.

Main differences

Figure 13: Surface restricting the processing plant area.

Figure 11: Topography point set and a designed surface.

Figure 12: Processing plant area to be restricted.

Figure 13: Surface restricting the processing plant area.

Figure 11: Topography point set and a designed surface.

[Previous image](#)

[Next image](#)

Restrict Mining

- * Select the points in the area that should be restricted, place them at the highest elevation of the topography, and set rest of them at the bottom.

- * Use as a restricted mining surface.

Figures 11 through 13 illustrate the process.

Note: The X and Y coordinates must remain the same.

Figure 16: Surface which can force the desired shape at any time.

Figure 14: Topography point set and a designed surface.

Figure 15: Desired pit to be reached.

Figure 16: Surface which can force the desired shape at any time.

Figure 14: Topography point set and a designed surface.

[Previous image](#)

[Next image](#)

Force Mining

- * Choose the points inside the area you want to force mine, place them at the elevation that you wish, and let the rest of them at the topography.

- * Use as a force mining surface.

Figures 14 through 16 illustrate the process.

Note: The X and Y coordinates must remain the same.

Figure 19: Surface of Force+Restrict mining to achieve pre-defined shape.

Figure 17: Topography point set and a designed surface.

Figure 18: Exactly pit design to be achieved.

Figure 19: Surface of Force+Restrict mining to achieve pre-defined shape.

Figure 17: Topography point set and a designed surface.

[Previous image](#)

[Next image](#)

Reaching an exact surface

- * Define your polygons or use designed surfaces, by following the methodology presented in 2. Creating surfaces Step-by-step.

* Use as a force and restrict the mining surface.

Figures 17 through 19 illustrate the process.

Note: The X and Y coordinates must remain the same.

Validating Surfaces

Introduction

The best way to generate surfaces is by using a topography surface created by MiningMath, which is created after the Data Validation, then manipulating only the Z coordinates. Make sure to meet all the surface requirements disclosed here.

Surface Requirements

- * Headers must be named as X, Y, and Z. These files have to obey an ascending value order in each one of the axes.
- * Same size of the block model, item 1.1 of this page explains it.
- * Its points must be aligned with blocks' centroids, items 1.1 and 1.2 help you to understand it.
- * Defined as a grid of points, the visual validation, item 1.2 shows it.
- * To be in the CSV format.

Figure 1: Surface file format of MiningMath

Surface and Block Model boundaries

By using the values of a Marvin Deposit file in Figure 2, we find the Block model centroids boundaries which begin at XMin=15; YMin=15 and has the maximum centroid value of XMax=5,295; YMax=8,265, which could be confirmed by checking the topography generated by MiningMath.

Then, its time to search for your surface limits, at this example the file chosen was "Surface-RM-offset-300m". The easy way to find it, is by filtering the axis values, as shown in Figure 3, which disclosed XMin= 15; YMin=15 and XMax= 1,815 and YMax= 1,785. Therefore, even though this file was also based on Marvin Deposit, it is not possible to use it since the surface smaller than the block model in place, which means that it does not have the same size as the block model file.

It is always worthwhile to check the limits of the designed surface (Figures 4 to 8) if we face an error. Remember that everything, even elevations, must be in the boundaries of the block model and check the recommendations for your surface.

Figure 3: Maximum value filter of Surface-RM-offset-300m

Figure 2: Check your block model information

Figure 3: Maximum value filter of Surface-RM-offset-300m

Figure 2: Check your block model information

Previous image

Next image

Figure 4: Surface (grid of points) smaller than the block model.

Figure 6: Surface (grid of points) bigger than the block model.

Figure 7: Surface (grid of points) with missing values.

Figure 5: Surface (grid of points) smaller than the block model.

Figure 8: Surface (grid of points) unaligned with the centroids.

Figure 9: Correct Surface (grid of points) aligned with the centroids and same block model size.

Maximum boundaries formula

The centroids maximum limit by using the equation for each axis:

Maximum centroid value = $OX + (NX * DX) - (DX/2)$

Where:

* OX is the origin of the X-axis;

* DX is the block dimension of the X-axis;

* NX is the number of blocks in the X-axis.

Note: $(DX/2)$ is related to the distance which should be summed on the origin to find the centroid of the first block, or reduced at the block model limit to find the last centroid, since origins are based on the corner of the block model.

Numeric and visual validation

The following video presents how to validate surfaces numerically and visually. This initial verification is what enables you to understand what might be happening and where the error is. The example used regards to the message: "Error parsing surface: Coordinates aren't properly spaced" but it fits any case where the surfaces used are causing a problem, especially when a red box error shows up.

In this case, as there are a lot of values that do not match the correct ones for X and Y, the quicker and easier way to fix it is restarting from the beginning.

Video 1: Validating surfaces.

Surfaces as a Guide

General Rule

The easier way to work with surfaces is by using a topography generated by MiningMath, then manipulating only the Z coordinates. The surfaces generated in MiningMath has always the same format, which means that each one of them has an equivalent number of lines and the data follows the same order from the first row to the last one. Therefore, you can choose the topography file and use it as a guide, which could facilitate conversions to original coordinates from your other software, the filter of regions and pits, the creation of force and restrict mining files, and a lot of options using this concept. It is worth mentioning that this approach could be easily done by using a simple worksheet and loading the CSV files generated in the viewer for further analysis.

Be careful with geometries

If you are considering geometries, mainly the Mining Width with surfaces imported from a different software package, there will be conflicts between the geometric criteria of MiningMath and the geometric criteria from the surfaces imported.

Offer a freedom in the last interval

You must give some freedom in the last interval so that MiningMath can also optimize the number of periods. It means that you should use <end> instead of using a lock period range, such as from 16 to 16 for instance.

Guide Mining In Deeper Areas

In order to add more material from deeper areas, you can use a base surface to increase its depth where you wish to get closer to what the project requires. Focusing on the main areas in early periods, you provide tips to the algorithm so that it can understand your approach by using surfaces. The following steps will reproduce an efficient workflow to promote this optimization:

Maximum Potential In The Bottom

1. Download this excel file to use as a guidance advance in the steps.
2. Insert the data in the yellow cells based on your block model information.
3. Paste the coordinates of the topography surface
4. Drag the formulas IX, IV, Z_Bottom, and Z_Top until the end of your data.
5. Define a plan Z to create a surface plan, import it on the viewer, and identify the elevation in which you want to force the bottom area below it.
6. Create a restrict mining surface to identify the maximum amount of ore, respecting the geotechnical aspects, could be extracted.
7. Set up your scenario and run.

[Previous image](#)

[Next image](#)

Note: As a result, you will get the maximum potential that can be extracted bellow the elevation that was chosen.

Mix Surfaces To Add Material From The Bottom

1. Place the Z coordinate from the base surface that you what to increase the depth to extract more material in the bottom
2. Paste the coordinate from the maximum potential scenario

3. Calculate the difference between them.
4. If the area of the maximum potential surface is below the surface base, it will consider the elevation lowest one. Therefore, it will consider the additional ore that can be extracted in the bottom along with the base surface used.
5. Use the mixed surface as force and restrict mining at the range that you want to achieve it.

[Previous image](#)

[Next image](#)

Note: After this step, MiningMath will generate operational surfaces so that you can use it on your projects considering the timeframes required.

Surface Constraints

Introduction to Surface Constraints

MiningMath uses a surface-constrained mine production scheduling, which is an improvement of the idea proposed by Marinho (2013). Surfaces are one of the most important constraints, allowing the user to impose its manipulations and knowledge to guide the optimization process. It can be used to force areas to back-fill operations, to allocate an in-pit crusher, to restrict an area considering different offsets, and show the economic impacts of preserving or not a given area or community. It also allows to incorporate an operational mine design as a requirement for a given time frame. For example, the mine design for the current year could be a mandatory requirement for the first period, while the rest of the mine sequence would have a new chance to be re-optimized and find a different sequence that finds more long-term value. The following pages unlock all the possibilities of the use of such features.

Internally in the algorithm:

- * To define slope angles and eliminate geotechnical errors, present in the blocks precedence method (Beretta & Marinho, 2014, 2015).

- * To handle geometric parameters and comply with minimum widths and maximum vertical rate.

Figure 1: Force mining could be understood as a minimum depth to be mined.

As optimization inputs:

- * To force mining and achieve a minimum depth, geometry, or area within a given time frame.

- * To restrict mining and ensure unavailable areas will not be considered as part of the optimization within a given time frame.

- * To force and restrict mining to achieve an specific design o guide the optmization.

Figure 2: Restrict mining could be understood as a maximum depth achievable.

As optimization outputs:

- * To outline the mine sequence throughout the Life of Mine that maximizes the Net Present Value.
- * Outputs will be a consequence of the optimization, which implies each set of project assumptions, constraints, and parameters, since it is unconstrained by pushbacks it will produce a different sequence of extraction, unlocking hidden opportunities.

Figure 3: Force and restrict mining used together could represents minimum depth to be mined and maximum achievable.

Surface Requirements

Surface formatting is simple and any surface output from MiningMath might serve as a start point for further manipulations or even for validations. It is important mentioning that they are exported/imported from/at MiningMath in Coordinates.

- * To have headers named as X, Y, and Z. These files also obey an ascending value order in each one of the axes.
- * To have the same size of the block model, which means it should not exceed the block model dimensions.
- * To have its points aligned with blocks' centroids in the X-Y plane.
- * To be defined as a grid of points.
- * To be in the CSV format.

VALIDATE YOUR SURFACES

How to create surfaces

To avoid any mistakes, manipulate an output surface from MiningMath instead of creating one from scratch.

1. Run any scenario to obtain the topography file in MiningMath's format.
2. Import the topography.csv, created by MiningMath, on a software able to manipulate it graphically.
3. Select points inside/outside a polygon. Move them up/down accordingly to the objectives to force or restrict mining. Points should be moved only up and down, along the Z direction.
4. Once the surface is ready, move it back to the original coordinate system.
5. Use it on MiningMath.

GET FURTHER INFORMATION HERE

Notes

- * X and Y coordinates should remain the same, with the same spacing between each pair of points.

- * For rectangular areas, a spreadsheet application is suitable for this task.

How to import surfaces on MiningMath

Surfaces are imported in two tabs of MiningMath: Geometric and Overview. Figure 1 zooms in the operational constraints from the Geometric tab. The main variables to use this feature are mentioned on Figure 4, which illustrates that surfaces are imported considering:

- * The purpose of forcing/restricting mining.

- * The period range when each surface is applicable.

- o MiningMath automatically defines a single period range from "1" to the "" and the user can also add custom intervals.

Figure 4: Geometric constraints.

In the example from Figure 5, the image highlights the fields to apply:

- * A restricting-surface valid for periods 1 and 2 (in green), which means that it would be respected until the end of the second year.

- * A forcing-surface valid for periods 1 to 5 (in blue), which means that the area has to be mined until the end of period 5.

Figure 5: General constraints and fields related to surfaces.

Video 1: Surface Constraints: The ultimate guide

Tips and Tricks

Articles

- * Reblocking

- * Cut-Off Grades

- * Integration

- * Coordinates to index conversion

- * Dilution

- * Keep your Original Data

- * Operational Solutions

- * Optimization Hints

- * Optimization Runtime

- * Optimizing your Workflow

- * Percent Models

- * Imperial System

- * Fixing Destinations

Reblocking

Reblocking is a method used to decrease the number of blocks in a block model by combining some of the smaller blocks to create larger ones. By doing so, the overall size of the blocks is increased, resulting in a reduction of the total number of blocks in the model.

MiningMath is 100% based on 64-bit technology and it is capable of handling tens of millions of blocks without requiring reblocking. Hence, it works only with regularized block models, and it does not offer reblocking options. Nonetheless, some research works have employed MiningMath with reblocked models, showing that processing time may be reduced without significant loss of mining selectivity.

Read more about this topic with the article 'Selective Reblocking of Block Models for Optimisations with MiningMath Software', written by our expert partner Karol Bartsch, [accessing it here](#).

Cut-Off Grades

Global cut-off grade overview

The concept of cut-off grade has been conceived to delineate what is ore and waste, considering the life of mine. Usually, manual assumptions are made to pre-define what must be considered ore or not, while using the LG or Pseudoflow methodologies. These approaches do not consider the value of the money through time in the decision-making process, which could generate a whole different mining sequence due to the choices of what and when it should be mined. Another challenge, that is faced when planning projects which involve multiple destinations, blending constraints, restrict mining areas, and all the complexities present in a real global optimization. MiningMath allows for such Global Optimization and you can still combine it with your current strategies!

MiningMath has no pre-defined assumptions in order to identify the cut-off limit, which is based on a math optimization considering a discounted cash flow while respecting production capacity, blending constraints, vertical advances, widths, and any other assumption. Another key aspect in this regard is that MiningMath is not constrained by fixed pits, pushbacks, or phases. Instead, the mining sequence is an optimized output, which is a consequence of each set of parameters used, allowing more flexibility to find completely new solutions. The advantages of these differences are even more evident for more complex cases with multiple destinations, or complex constraints that could be neglected, hiding opportunities.

As the algorithm here has no manual destinations appointments, it always sends the less valuable blocks to the dump, considering the set of constraints imposed, what means that MiningMath tries to comply with all the constraints inserted, by respecting this priority order, to define an optimized cut-off that can meet all of the requirements and increase the NPV as a consequence of a global optimization. Meanwhile, blocks that have positive values when processed can also be discarded to increase NPV based on the minimum economic value (economic value cut-off) going to the plant at that a specific period. Scenarios without stockpiling policy could have even higher positive values going to the waste since they do not have any other destination. Considering these examples, MiningMath can deliver quite different results from what you were expecting from your previous assumptions. However, you can also get closer, as much as you wish, to any solution by using the approaches suggested below.

Forcing a cut-off grade

Forcing a cut-off grade on MiningMath will likely make you lose part of the advantages it can offer. However, for many reasons, mining professionals might still be willing to use either to compare different approaches, to understand the practical effects of using it, or not, etc. The approaches mentioned could also be used to forbid any material type on the plant, and the use of the item 2.1. could even reduce the complexity of the algorithm, since it could use fewer constraints to deal with it.

Using Economic Values

You can create multiple columns of economic values, each one for a cut-off you want to test. Then, force MiningMath to use this limit by defining very negative values for the destination you want to avoid, as it is shown in Figure 1, for a cut-off of 0.5. The math is:

Economic Value Process = If [Ore_Grade] > [0.5], then [f(Economic Value)], else [-999,999,999.00]

Figure 1: Block model setup to incorporate cut-off grades when defining economic values.

Using Average tab

Grades in MiningMath are controlled as a minimum and/or maximum average calculation, which means that these limits do not represent cut-off values since the algorithm can use lower values to blend higher ones. Thus, to use this approach, just set a very negative value on the grades below the cut-off so that these blocks would reduce substantially the average when processed. It can also work to constraint a contaminant maximum limit by adding a high grade on it, as it can be viewed in Figure 2. Once again, the math is:

Ore_Grade1 = If [Ore_Grade] < [0.5], then [-999,999,999.00], else [Ore_Grade]

Figure 2: Block model setup to incorporate cut-off grades using average.

Using Sum tab

Another option is by using the sum tab to control material types. Therefore, it would be required to create a field to calculate only waste blocks mass and set the constraint of the maximum limit of it in the plant as zero, as it is seen in Figure 3. It is worth mentioning that this approach could increase the complexity of the optimization due to the priority order within the algorithm.

Tonnage_Waste = If [Ore_Grade] < [0.5], then [Volume*density], else [0]

Figure 3: Block model setup to incorporate cut-off grades using sum.

Understanding results

The mined blocks file, as seen in Figure 4, is the main output to trace the blocks from each destination, to understand the results, and to find the best way to enhance your reporting based on

any detail that you wish to disclose. There are many useful tips to identify, and understand the results generated, some of them, are listed below:

- * Filter results where the period mined is equal to the period processed. Check process economic values of those blocks, which were processed, identify the lower one (which means the cut-off value at the plant), and compare it with higher processing economic value of those who went to the dump.

- * Calculate the average grade of any material in period mined equal to the period processed filter. Check if the blocks, which are going to dump, would have exceeded any limit in the plant, if so, even having good economic values, they would not comprise the constraints in place.

To sum up, there are a lot of validations that can be done to understand why the algorithm is taking such decisions. It is also worth mentioning that any constraint can influence the results, even geometric ones, which could change the sequence and change the destination of the block at any period.

Figure 4: Mined blocks output file.

Integration

MiningMath Integration

MiningMath doesn't necessarily compete against mine scheduling optimization packages. The only concept we and all research centers worldwide recommend mining companies to overcome is the one related to Pit Optimization, due to the set of problems you have to face when dealing with such technology.

Figure 1: Lerchs-Grossmann/Pseudoflow

Therefore, even our simplest version has more features to generate nested pits with better control so that you could design better pushbacks and define a mine schedule using your preferred tool. The reason why this software can deliver such results is the Direct Block Scheduling methodology based on Mixed Integer Linear Programming (MILP) model and proprietary heuristics. Check other technical details and related research in our theory section.

Figure 2: Direct to block scheduling.

MiningMath also allows you to generate optimized pushbacks, which could facilitate your design process and guide your mine schedule while using other software packages. Notice that our tool is an optimizer that simply breaks the whole deposit (your block model) into smaller pieces, aiming for maximum Net Present Value, but respecting as many constraints as you wish:

Figure 3: Optimized pushbacks and optimized schedules.

A usual application of our technology is basically on strategy optimization for building decision trees. Once we run dozens/hundreds of scenarios of the yearly schedule optimization and fine-tune their parameters/constraints, our users take some of the resulting surfaces of MiningMath and use them to design some pushbacks so that they could integrate with other packages, such as MSSO, COMET, etc. This procedure could be accelerated/simplified by working with packages of years and finding shapes closer to pushbacks you're used to.

The outputs of our software will serve basically as optimized pushbacks, searching for maximum NPV and controlling whatever variable you consider necessary. Once we manage to import MiningMath surfaces into the other package, they will serve as guidance and they should assist the other package

in finding higher NPVs. Most of these packages also allow us to predefine the blocks' destination, if we wish to use MiningMath optimized cutoff policy. Finally, the package should have "only" the duty to do the bench scheduling, according to your short-term operational/tactical needs.

Even if you decide, for any internal reason, that you have to use LG/Pseudoflow to define final pit limits, there is no problem at all. MiningMath is the only tool available in the market capable of performing complete strategic analysis by building decision trees unconstrained by predefined pushbacks. Please, check this short example (in Spanish) with dozens of scenarios just for the decision on CAPEX regarding processing capacities. Check also the second half of this video for a broader view on how to use the same concept to take strategic decisions on many other aspects related to mine projects or ongoing operations. I assure your managers will get much more interested in your reports once you start adding this sort of strategic analysis. Notice you could perform this sort of analysis either free of constraints or respecting any pre-existing (designed) ultimate pit or pushbacks.

Figure 4: Multiple scenarios to build.

Going one step beyond, we also have clients improving their adherence and reconciliation between long and short-term mine plans by using MiningMath as a complementary tool. Notice that, by using MiningMath in strategic mine planning, you could add more constraints from real-life operations, even if you decide just to check your current long-term plans. Also, notice you could place some surface limits, such as the designed surface of the next five years plan for example, and give some controlled freedom to short-term planners to rerun their mine plans, including more operational details, as long as they don't change anything from period 6 on and they don't affect the NPV negatively. Whenever they find an issue or an opportunity, short and long term teams have a way to collaborate and generate new joint configurations that account for all the strategic and tactical needs of the project simultaneously. All the remaining details, such as the designs, could be adjusted using the current mining packages available.

If you wish to skip such steps and go straight to your final designed plans, we can guide you through this process, which includes a loop of running MiningMath and designing surfaces, until reaching a reasonable and operational sequence. This is a much more innovative procedure, which tends to achieve higher NPVs.

Coordinates to index conversion

Index to Coordinates

The indices of each block represent its position in the model, indicating in which column, line, and level (IX, IY, and IZ) it is.

The indices must be integer values, starting with any value (for Marvin model, it was adopted the indices 1,1,1 for the first block).

The model's origin must be placed at the bottom portion, starting to count from the minimum coordinates at X, Y, and Z.

Figure 1 highlights the origin of the Marvin block model and the first block index coordinates (1,1,1).

Figure 1: Block's Matrix.

However, if the block model contains geo-referenced information based on coordinates, they could be converted into indices before being imported to MiningMath.

To perform this conversion, check the following demonstration on how to convert coordinates into indices using data from Figure 2 and the equation from Figure 3.

Figure 2: Sample data to convert coordinates into indices.

Figure 3 exemplifies using the X-axis but the process is the same for Y and Z axes by just using the corresponding information.

Figure 3: Equation to convert coordinates into indexes.

[Click here to download a spreadsheet to convert the coordinates into indices and calculate the economic values.](#)

Figure 4: Resultant coordinates converted to indexes.

Do you want more details on Index Coordinates?

The video below exemplifies the conversion process in case you have any doubts.

Video 1: How to convert block coordinates into block indices.

Dilution

This is an informal video on how to consider dilution and mining recovery.

Video 1: Dilution and Mining Recovery.

READ MORE

* Percent Models

Keep your Original Data

Keep your data from different software

A common question while using MiningMath integrated with other mining packages is: How to keep the data from one to another?

In order to facilitate this format exchange, you can keep any field from the other software and import it as a Other parameter. By following this approach MiningMath will keep the data in the reports and MinedBlocks.csv generated after the optimization. Thus, you will be able to import it on your mining package by following the same parameters as to when it was exported.

This approach is quite useful when we have to keep the original coordinates X, Y, and Z or any other data that would help to identify the blocks in the platform which it came from.

Operational Solutions

Common questions

A block-by-block schedule is likely very difficult to achieve in operations - example: operations does not allow using multiple types of equipment on the same bench. How to cope with it?

Although MiningMath works with blocks as inputs, the level of connectivity is user-defined by playing with geometrical parameters in the interface.

The following image shows results for the Marvin deposit when changing the Minimum Widths (filtered view after Period 2).

Figure 1: Marvin deposit and visual comparison across scenarios differing their operational widths.

Figure 2: Marvin deposit and NPV impact from scenarios differing their operational widths.

Note everything changes when playing with a single parameter, including the Life of Mine and geometries. Such impacts are also possible when playing with economic aspects, slope angles, environmental and boundary constraints, fleet size, processing setups, blending requirements, etc.

Watch the following video on how to play with operational constraints to achieve results closer to the reality of any project.

Video 1: Operational Constraints.

Optimization Hints

The general relation between the number of Constraints and NPV tends to be inversely proportional, which means that scenarios with fewer constraints imposed, the optimization tends to find higher NPV since the algorithm free to search for better solutions. Therefore, it is always recommended that the full potentials of the mining project are tested first, as recommended in the optimized workflow, which starts at the Data Validation generating results that could be used in further steps. Regarding these aspects, a pretty useful approach was done by the MiningMath team throughout 2000 simulations (Figure 1) on a project seeking to identify the main impacts that the constraints had.

The firsts executions considered the individual variation of each constraint, therefore, Copper Selling Price, Mining Widths, Vertical Rate of Advance, and the rest of the parameters were modified to understand the tendency of each one. As we can see, most of them, when individually varied, have a high probability of generating NPVs around 900M\$.

Figure 1: Project evaluation over 2000 executions.

The second round was performed by running "Overall" scenarios, which consider a random value of each one of the 11 variables at the same time on the project. As you can see, the combination of this amount of constraints has a higher probability of generating NPVs around 200M\$.

In summary, these results illustrate the intuitive assumption that by adding more and more constraints the probability of getting higher NPVs decreases drastically that is why the users have to apply "smart" concepts and constraints to achieve such results.

MiningMath deals with multiple constraints and aims to maximize NPV simultaneously. Remember that it is computationally impossible to guarantee that all possible solutions are evaluated by any algorithm and, therefore, it is impossible to guarantee that the maximum NPV will be delivered by a single run. Having such concepts in mind, a simple modification on a single parameter can generate

unexpected effects on NPV. Thus, it is always worthwhile to identify tendency, what it is not possible to find with only two runs. Note that it is required to create a curve with a set of runs if we wish to find tendencies of an algorithm that includes such heuristics.

The logic behind such analysis is quite simple: for any optimization algorithm that aims to solve such complex problems, if we put too many constraints, we give no freedom for the algorithm to explore the solution space. Therefore, we tend to find solutions with lower NPV. On the other hand, when we give too much freedom, the algorithm will have the whole space to search for solutions, with no guidance on what would be reasonable in real life, so good solutions might not be found. The best compromise is when the user starts free of constraints and, then, starts giving some reasonable guidance (Figure 2). Therefore, smart constraints might help the algorithm in focusing only on a certain part of the space and actually deliver higher NPVs.

Figure 2: Applying smart constraints to reduce a broader solution space and find best results on a limited one.

The main suggestion to explore the potential is by running the first iterations and evaluations with as few constraints as possible and without geometries (most complex in the mathematical model). Then, start to move gradually to detailed scenarios, by adding parameters one-by-one, which will allow you to understand the impact of each constraint in the NPV, and guide decisions through an Integrated Workflow. Thus, by measuring how much each assumption "costs" to the project in the long-term, the managers will be able to choose the best way to use them.

MiningMath is a technology composed of mathematical programming and heuristics, which do not disrespect the constraints imposed to increase the NPV as the hierarchy (Figure 3) disclosed on this content explains. Experienced users take advantage of this aspect and slowly place constraints that might even help the optimizer to find solutions with higher NPV. A scenario with the total movement/production free, for instance, results in the better distribution of waste over time while adding/modifying a total limit makes the algorithm search the best result within these capacities and even find surprising NPVs among them. Therefore, to assess tendencies, it's recommended to draw a curve of scenarios, such as free, 100 Mtpa, 90 Mtpa, 80 Mtpa, and so forth versus NPV.

The user always can guide the optimization by relaxing constraints on the hierarchy whenever possible, which works as hints to the algorithm to reach feasible outcomes. Flexibilizing the Slope Angles in the short term could bring better designs to your project, for instance. Whereas the user has an understanding of the optimization problem, and how the constraints are acting on it, it is possible to insert smart hints to the algorithm to reach a reasonable result.

Figure 3: Constraints hierarchy order.

Key takeaways

One of the main advantages of MiningMath's algorithm is the capability of finding intermediary solutions between the extremes outcomes. Therefore, the user can give hints to guide the algorithm, such as:

* Letting the algorithm free, with fewer constraints (bigger solution space), is great for exploring the economic potential of a project.

- * When dealing with more constraints, reducing the solution space might give some hints to the algorithm and help it to achieve the goals set.
- * Starting to optimize from the most free scenario up to the most constrained one.
- * Relaxing low-priority constraints manually to guide the algorithm.
- * Breaking complex scenarios into smaller problems and iterating over multiple scenarios.
- * Avoid using constraints for the same purpose with the same priority in the hierarchy order. |

Relaxed Constraints vs Violated Constraints

These are subtle nuances that should be evaluated along about this topic as: When are constraints disrespected a consequence of the algorithm pursue to an intermediary outcome? When such deviations should be considered as a violation or an improper solution?

Constraints relaxed to achieve a feasible outcome. Examples:

- * Processing stream shortfalls.
- * Operational constraints.
- * Average constraints (depending on the deviation).
- * Constraints based on Sums (depending on the deviation).

Constraints violated that would represent, in fact, an unfeasible project. Examples:

- * Production limits exceeded.
- * Slope constraints.
- * Average constraints (depending on the deviation).
- * Constraints based on Sums (depending on the deviation).

This information are usually reported at the report tab on the excel output file.

Optimization Runtime

Introduction

The optimization run time is a common concern for professionals dealing with robust models. This page aims to provide context and guidance to improve run times, which might be quite useful for having a big picture of the project's behavior under different assumptions and hypotheses.

Runtime Barriers

The runtime depends on a combination of multiple aspects. It is directly related to the complexity of the deposit. The runtime is directly proportional to the number of:

- * Blocks.
- * Multiple destinations (+3)
- * Constraints in use and conflicting goals with the same hierarchy order.
- * Variables imported.

- * Period ranges.
- * Parameters changing over time.
- * Multi-mine deposits.

Model size limit

Often, users are concerned with the limits to handle models with +20M blocks. MiningMath can virtually handle any model size. It has successfully made tests with models up to 240M blocks without reblocking, which took three weeks, and over a 32 Gb desktop machine.

Typically, datasets with 5 million blocks take a few hours (in an 8GB RAM machine). The technology can also execute multiple scenarios in parallel, using the same computer. There is no need for special servers with extra RAM capabilities for deposits of average size.

Hardware Improvements

Memory

Overall, the main bottleneck for MiningMath is memory consumption. Hardware upgrades that most positively impact the optimization run time are:

- * RAM capacity.
- * RAM frequency.

Cores and threads

MiningMath is a single-thread application, which means:

- * Additional cores and threads do not affect the optimization run time.
- * Processors with higher clock speeds improve the run time.

Additionally, the user can open several instances of MiningMath to run multiple scenarios in parallel. While a single scenario might not run faster, it is possible to improve the average of scenarios tested per unit of time.

Strategies to improve the runtime

Use surfaces

The most recommended strategy is passing through the tutorial steps of validating data and constraints validations then starting using the surfaces as a guide to reduce the complexity, without losing dilution aspects on your approach.

To get such guidance on a broader view with a reduced runtime use the Exploratory Analysis to get pushbacks and insights on what could be used. The last step is to get a detailed Schedule since the model has such complexity. If such approaches do not offer a proper runtime, try to get intermediate results by splitting the total production into 2 or 3 periods.

Reblocking

This is another approach that could be done, which is not recommended since we lose dilution aspects by increasing the block size. Therefore, if your blocks have the dimensions of 5x5x5 and you

could increase it to 10x10x10, which could reduce the number of blocks to half of its standard dataset size. However, according to user's feedback, a multi-mine project with 32M blocks for the final integrated model, which mining complex considered various processing routes and operational constraints involving certain mine infrastructure within the final pit boundaries, obtained the following runtimes:

- * Quadruple reblocking in each direction took about 4-5 hours for each run.
- * Triple reblocking took 12 hours.
- * Double reblocking took 36 hours.

As the solution became clearer, reblocking was gradually reduced to have more flexibility.

Optimizing your Workflow

Tips and Tricks

The following video comprises some tips and tricks to optimize your workflow when running multiple scenarios. The options comprise:

- * Altering multiple scenarios through the SSSCN files, which are XML files that could be parsed using scripts created by the user, then running scenarios through the interface.
- * Running scenarios from the command prompt, without recurring to the User Interface.

Video 1: Optimizing your workflow.

Percent Models

Formatting Percent Models

1. Defining the lithology

Although this is not a mandatory step for the optimization process, the lithology can be defined considering: I in terms of tonnage.

- * The tonnage of a block.
- * The value of a block.

Figure 2 shows the information of a block that could be classified as:

- * MX, if considering the greatest parcel in terms of tonnage.
- * MX, if considering the greatest parcel in terms of tonnage.

Figure 1: Illustration of a block with different parcels.

Figure 2: Block information divided into lithologies OX, MX, PM, Waste.

2. Density

MiningMath calculates tonnages based on [block size × densities]. The average density of a block should be equal to the weighted average based on lithologies and their respective percents.

3. Recovery

Recoveries should also be calculated considering the amount of material recovered from each parcel of the block.

4. Economic Value

The economic value is calculated considering the amount of material recovered from each parcel, along with their respective revenues and costs.

There are two ways to calculate the economic value of a block:

- * Without dilution (Option 1): only ore parcels feed the plant (Figure 3).
- * With dilution (Option 2): the entire block feeds the plant (Figure 4).

4.1 Option 1 without dilution

In this case, the Economic Value for the process will consist of Revenue - Costs, where:

- * Revenue refers to the ore parcel (70%).
- * Processing Costs refer to the ore parcel (70%).
- * Mining Costs refer to the entire block (100%).

As MiningMath will process the entire block, input a greater value for the process limit, assuming the algorithm will feed the plant with the remaining parcel of waste (30%).

Create auxiliary columns to track and control tonnages limits of ore, waste, and any specific lithotype you want (Figure 6).

Ignore default production charts and consider the tonnage ones being used as Other Constraints.

Figure 3: Strategy for economic values considering only ore feeds the plant.

4.2 Option 2 with dilution

In this case, the Economic Value for the process will consist of Revenue - Costs, where:

- * Revenue refer to the ore parcel (70%).
- * Processing Costs refer to the entire block (100%).
- * Mining Costs refer to the entire block (100%).

In this case, as there is dilution, the processing limit inputted in the interface, should be the real plant limit.

Again, auxiliary columns will provide further control of tonnages for each parcel (Figure 6).

Figure 4: Strategy for economic values considering ore and waste feed the plant.

Figure 5: Example of calculations for a block composed of different lithotypes and its respective economic values assuming no dilution and diluted material.

5. Other Constraints

- * Create auxiliary columns for tonnages of each lithotype, as demonstrated in Figure 6.

- * During the importation, set them as Other.

This step will make you able to track and control tonnages of each material.

Figure 6: Example on how to track specific information of a block.

READ MORE

- * Dilution.

- * Other Constraints.

Imperial System

For importing databases, MiningMath uses the metric system exclusively. In case your database is in the imperial system, it should be modified to a counterpart in the metric system. This page provides a script and further instructions for you to do this conversion successfully.

For this process, python and pandas packages are required (see installation guidelines below). In case you find any issues using the script, or if you make any modifications to it that could be useful for the community as a whole, please share your discoveries at our Forum.

Script download

If you're already familiar with python scripts, just copy or download this simple python script and run it in your machine. Otherwise, keep scrolling for further instructions.

DOWNLOAD SCRIPT HERE.

```
import pandas
```

```
# define conversion consts
```

```
# foot to metter const
```

```
ft_to_m = 0.3048
```

```
# short tonne to metric tonne const
```

```
st_to_t = 0.907184
```

```
# ounce to gram const
```

```
oz_to_g = 28.34952
```

```
# import imperial model
```

```
imperial_model = pandas.read_csv("imperial_model.csv")
```

```
# create metric model
```

```
metric_model = pandas.DataFrame(columns=['X', 'Y', 'Z'])
```

```
# set coordinates to metric model (foot to meter)
```

```
metric_model['X'] = imperial_model['X'] * ft_to_m
```

```
metric_model['Y'] = imperial_model['Y'] * ft_to_m
```

```
metric_model['Z'] = imperial_model['Z'] * ft_to_m
```

```
# set dimension to metric model (foot to meter)
```

```
metric_model['!DIM_X'] = imperial_model['DIM_X'] * ft_to_m
```

```
metric_model['!DIM_Y'] = imperial_model['DIM_Y'] * ft_to_m
```

```
metric_model['!DIM_Z'] = imperial_model['DIM_Z'] * ft_to_m
```

```
# set mass to metric model (short tonne to metric tonne)
```

```
metric_model['!MASS'] = imperial_model['MASS'] * st_to_t
```

```
# set volume to metric model (cubic metter)
```

```
metric_model['!VOLUME'] = metric_model['!DIM_X'] * metric_model['!DIM_Y'] *
```

```
metric_model['!DIM_Z']
```

```
# set density to metric model (metric tonne per cubic metter)
```

```
metric_model['%DENSITY'] = metric_model['!MASS'] / metric_model['!VOLUME']
```

```
# set grades to metric model (ounces per short tonne to grams per metric tonne or ppm)
```

```
metric_model['@GRADE_AU'] = imperial_model['GRADE_AU'] * oz_to_g / st_to_t
```

```
metric_model['@GRADE_CU'] = imperial_model['GRADE_CU'] * oz_to_g / st_to_t
```

```
# set recovery values to metric model
```

```
metric_model['*REC_AU'] = imperial_model['REC_AU']
```

```
metric_model['*REC_CU'] = imperial_model['REC_CU']
```

```
# set economic values to metric model
```

```
metric_model['$PROCESS'] = imperial_model['PROCESS']
```

```
metric_model['$WASTE'] = imperial_model['WASTE']
```

```
# export metric model to csv
```

```
metric_model.to_csv("metric_model.csv", index = False)
```

Copy

1) Installing Python

1. Download Python's latest version at <https://www.python.org/downloads/>.

Python download webpage

2. Once the download is complete, open the .exe and follow the instructions for a default installation. Make sure to select "Add Python to Path" before proceeding, as depicted below.

Python installation screen

3. At this point, the installation should be concluded. You can check if Python has been correctly installed by running the command `python --version` at Windows PowerShell.

Python version on Windows PowerShell

2) Installing Pandas

Pandas is an open source data analysis and manipulation tool, built on top of Python. Follow the steps below to install it:

1. Open the Windows PowerShell and run the command "pip install pandas".

Pandas install command

2. Once the installation is complete, you're able to run Pandas inside your Python programs. You can check if Pandas has been correctly installed by running the command "pip show pandas" at Windows PowerShell.

Pandas version

3) Converting your database

This script can be used to convert foot to meter; short tonne to metric tonne; and ounce to gram.

It works with the columns: X, Y, Z, DIM_X, DIM_Y, DIM_Z, MASS, VOLUME, DENSITY, GRADE_AU, GRADE_CU, REC_AU, REC_CU, PROCESS, WASTE.

Follow the steps below:

1. Save your database in a file named `imperial_model.csv`, at the same folder where your script is located.
2. Run the command `python imperial.py` at the Windows PowerShell from the folder where the script is located. The example below shows the script in the Downloads folder.

Run script example

3. Open the output file named `metric_model.csv`, and that's it! Your data has been converted to the metric system.

Output file example

Fixing Destinations

Introduction

We have already seen that MiningMath works with Economic Values for each destination, taking each one of them into account to decide whether or not a block should be mined and where it should be sent to. Thus, fixing destinations or pre-defining them is not a concern anymore, especially with MiningMath technology. However, it might still be necessary if you are using MiningMath to define pushbacks while making use of other constraints.

Applications

- * Fixing destinations to define pushbacks.
- * Preventing a block with a certain characteristic (e.g. rock type) to be sent to a given destination (e.g. processing plant).
- * Among others.

How to lock destinations down?

The concept behind fixing destinations is very simple and relies on the Economic Values. These are the parameters considered by MiningMath to decide whether to mine a block or not.

The user simply needs to define very negative values (e.g. -999,999,999) for a destination that a block should not be sent to.

The figure below presents an example of blocks with very negative values for the Waste Dump, which means they will be sent to the destination with the highest economic value: the processing plant (if mined).

Figure 1: Blocks with very negative values for the Waste Dump.

DISCLAIMER: By doing this procedure to reproduce the traditional way of calculating economic values you are impeding MiningMath to reach its full potential.

A- A+

MiningMath (c) 2023

Contact

- * Contact us

- * WhatsApp Chat
- * Telegram Chat
- * +55 (31) 99887-8394

Company

- * About us
- * Press
- * Terms & Policies
- * Our plans
- * Blog

Resources

- * Downloads
- * Certification
- * Forum
- * Knowledge base
- * Teaching
- * Research

Translation

Knowledge Base Contact

Top of Form

Bottom of Form

1. Home
2. Docs
3. Knowledge Base
4. Ask GPT

Ask GPT

You can use ChatGPT to help you with our knowledge base. First, you will need to...

1. Home
2. Docs
3. Knowledge Base
4. Workflow

5. Multivariate Sensitivity Analysis

Multivariate Sensitivity Analysis

Multivariate Sensitivity Analysis is the process of creating and analyzing scena...

1. Home

2. Docs

3. Knowledge Base

4. Workflow

5. Bottleneck Analysis

Bottleneck Analysis

Bottleneck Analysis is the process of creating scenarios to conduct extensive se...

1. Home

2. Docs

3. Knowledge Base

4. Workflow

5. NPV Enhancement

NPV Enhancement

NPV Enhancement is the process of creating scenarios to conduct extensive search...

1. Home

2. Docs

3. Knowledge Base

4. Workflow

5. Design Enhancement

Design Enhancement

Design Enhancement is the process of creating scenarios to conduct extensive sea...

1. Home

2. Docs

3. Knowledge Base

4. Workflow

5. Selectivity Analysis

Selectivity Analysis

Selectivity Analysis is the process of generating and analyzing scenarios to mea...

1. Home
2. Docs
3. Knowledge Base
4. Workflow
5. Best-Worst Range Analysis

Best-Worst Range Analysis

Best-Worst Range Analysis is the process of generating and analyzing scenarios t...

1. Home
2. Docs
3. Knowledge Base
4. Workflow
5. NPV Upside Potential

NPV Upside Potential

NPV Upside Potential is the process of generating and analyzing scenarios to mea...

1. Home
2. Docs
3. Protected: Getting Started

Protected: Getting Started

There is no excerpt because this is a protected post.

1. Home
2. Docs
3. Protected: Getting Started
4. Protected: Validation stage

Protected: Validation stage

There is no excerpt because this is a protected post.

1. Home
2. Docs
3. Protected: Getting Started
4. Protected: Formatting stage

Protected: Formatting stage

There is no excerpt because this is a protected post.

1. Home

2. Docs

3. Protected: Getting Started

4. Protected: Destinations and economic values

Protected: Destinations and economic values

There is no excerpt because this is a protected post.

Top of Form

Bottom of Form

Notifications