A Comparative Study of NPV Open Pit Schedule Optimisation Using Mixed Integer Programming and Evolutionary Algorithms

TSOUCHLAKIS, D., BREGU, E., KAPAGERIDIS, I.

DEPARTMENT OF ENVIRONMENTAL AND POLLUTION CONTROL ENGINEERING, TECHNOLOGICAL EDUCATIONAL INSTITUTE OF WESTERN MACEDONIA, GREECE

AACHEN INTERNATIONAL MINING SYMPOSIA
Second International Conference
MINING IN EUROPE
June 7-8 2017 Aachen, Germany

Introduction to the Scheduling Problem

- Increasing the value of a mine schedule is one of the main concerns of mine planning engineers.
- Mine schedules are generated at different stages of a mining project, from prefeasibility studies all the way to the last few months of its life.
- The software tools available to engineers for scheduling allow quick development of different production scenarios that meet production targets and satisfy operational constraints.
- Most of the scheduling software packages incorporate some optimisation system that enables engineers to minimise or maximise an objective function while constraining other parameters, both quantitative (tonnages, volumes, commodity units, time, etc.) and qualitative (grades, ratios, etc.).

NPV Schedule Optimisation Software Compared

- A special and very common objective function for schedule optimisation is the **net present value (NPV)** of the schedule.
- Traditional optimisation algorithms for mine scheduling are mostly based on some form of linear, dynamic or mixed integer programming and are commonly quite time consuming to setup. Evolutionary methods such as genetic programming have been tried by various researchers in the past.
- We examine two scheduling solutions capable of NPV optimisation:
  - **Maptek Chronos**, a spreadsheet scheduler that uses IBM ILOG CPLEX mixed integer programming for optimisation, and
  - **Maptek Evolution**, a block scheduler and one of the first commercial products based on evolutionary algorithms for open pit mine scheduling.

Mixed Integer Programming

- A mixed integer programming (MIP) problem may contain both ***integer*** and ***continuous variables***.
- Integer variables may be restricted to the values 0 (zero) and 1 (one), in which case they are referred to as ***binary variables*** or they may take on any integer values, in which case they are referred to as ***general integer variables***.
- A variable of any MIP that may take either the value 0 (zero) or a value between a lower and an upper bound is referred to as ***semi-continuous***.
- ***Continuous variables*** in a MIP problem are those which are not restricted in any of these ways, and are thus permitted to take any solution value within their (possibly infinite) lower and upper bounds.

Mixed Integer Programming Example

Maximize **$x_1 + 2x_2 + 3x_3 + x_4$**

subject to

$-x_1 + x_2 + x_3 + 10x_4 \leq 20$

$x_1 - 3x_2 + x_3 \leq 30$

$x_2 - 3.5x_4 = 0$

with these bounds

$0 \leq x_1 \leq 0$

$0 \leq x_2 \leq +\infty$

$0 \leq x_3 \leq +\infty$

$2 \leq x_4 \leq 3$

x_4 integer

Mixed Integer Optimisation

- The IBM ILOG CPLEX Mixed Integer Optimiser solves MIP models using a very general and robust branch & cut algorithm.
- Optimising a MIP model involves:
  1. Finding a succession of improving integer feasible solutions (solutions satisfying the linear constraints and the integrality conditions), while
  2. Also working toward a proof that no better feasible solution exists and is undiscovered.
- In the branch & cut algorithm, CPLEX solves a series of continuous subproblems.
- CPLEX builds a tree in which each subproblem is a node.

Maptek Chronos & IBM ILOG CPLEX

mixed integer programming

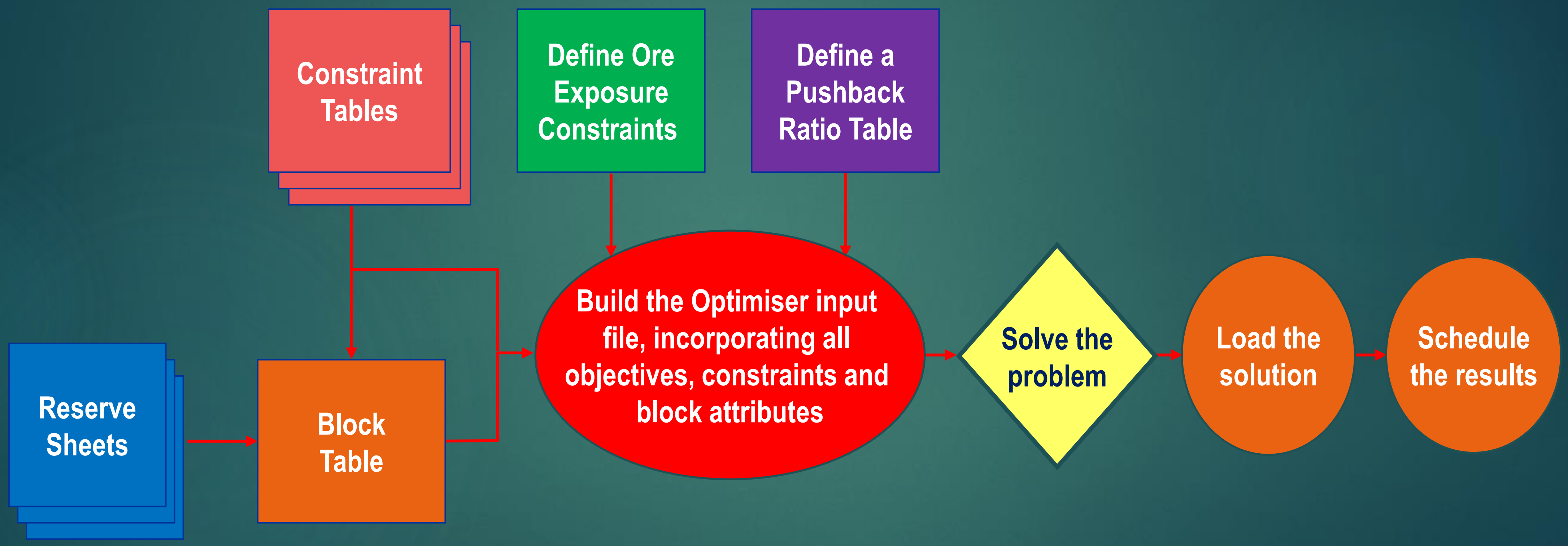
- Maptek Chronos, the first of the two packages considered, works within an object-oriented spreadsheet environment to store, manipulate and report mine production information.
- It can be used for short, mid and long term scheduling and works on the basis of mining blocks for which reserve information has been calculated in advance.
- The mining blocks can be anything that the user designs and models (pushbacks, benches, blasts, etc.) and are reserved against a resource model which is either a block model or a stratigraphic model based on grids.
- The mining blocks and corresponding reserves information are transferred to a fully customisable spreadsheet environment and scheduled using some user defined sequence.

Maptek Chronos & IBM ILOG CPLEX

mixed integer programming

- Block exposure rules need to be defined by the user depending on the type of mining blocks used and the type of mining (surface or underground).
- Optimisation functionality is provided through an interface to an optimisation system based on IBM ILOG CPLEX.
- The Chronos optimisation interface formulates the scheduling problem to a mixed integer programming problem which is passed on to CPLEX for solving.
- The solution is then translated back to a scheduling sequence and is stored in the spreadsheet environment.

Maptek Chronos Optimisation



Evolutionary Algorithms

- An evolutionary algorithm (EA) is a subset of evolutionary computation, a generic population-based metaheuristic optimisation algorithm.
- An EA uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection.
- Candidate solutions to the optimisation problem play the role of individuals in a population, and the fitness function determines the quality of the solutions.
- Evolution of the population then takes place after the repeated application of the above operators.
- Evolutionary algorithms often perform well approximating solutions to all types of problems because they ideally do not make any assumption about the underlying fitness landscape.

Evolutionary Algorithm Process

1. Generate the initial population of individuals randomly. (First generation)
2. Evaluate the fitness of each individual in that population (time limit, sufficient fitness achieved, etc.)
3. Repeat the following regenerational steps until termination:
   - Select the best-fit individuals for reproduction. (Parents)
   - Breed new individuals through crossover and mutation operations to give birth to offspring.
   - Evaluate the individual fitness of new individuals.
   - Replace least-fit population with new individuals.

Maptek Evolution

evolutionary algorithms – cloud based

- It is one of the most recent scheduling systems commercially available and probably the only one based on evolutionary algorithms.
- As it is focused to open pit mining, it is more straightforward to setup as the user doesn't need to deal with block exposure rules.
- It is also a block scheduler – the units considered for scheduling are regular blocks derived from the resource block model.
- There is no spreadsheet holding the reserve information – the block model itself is imported and manipulated for scheduling purposes.

Maptek Evolution

evolutionary algorithms – cloud based

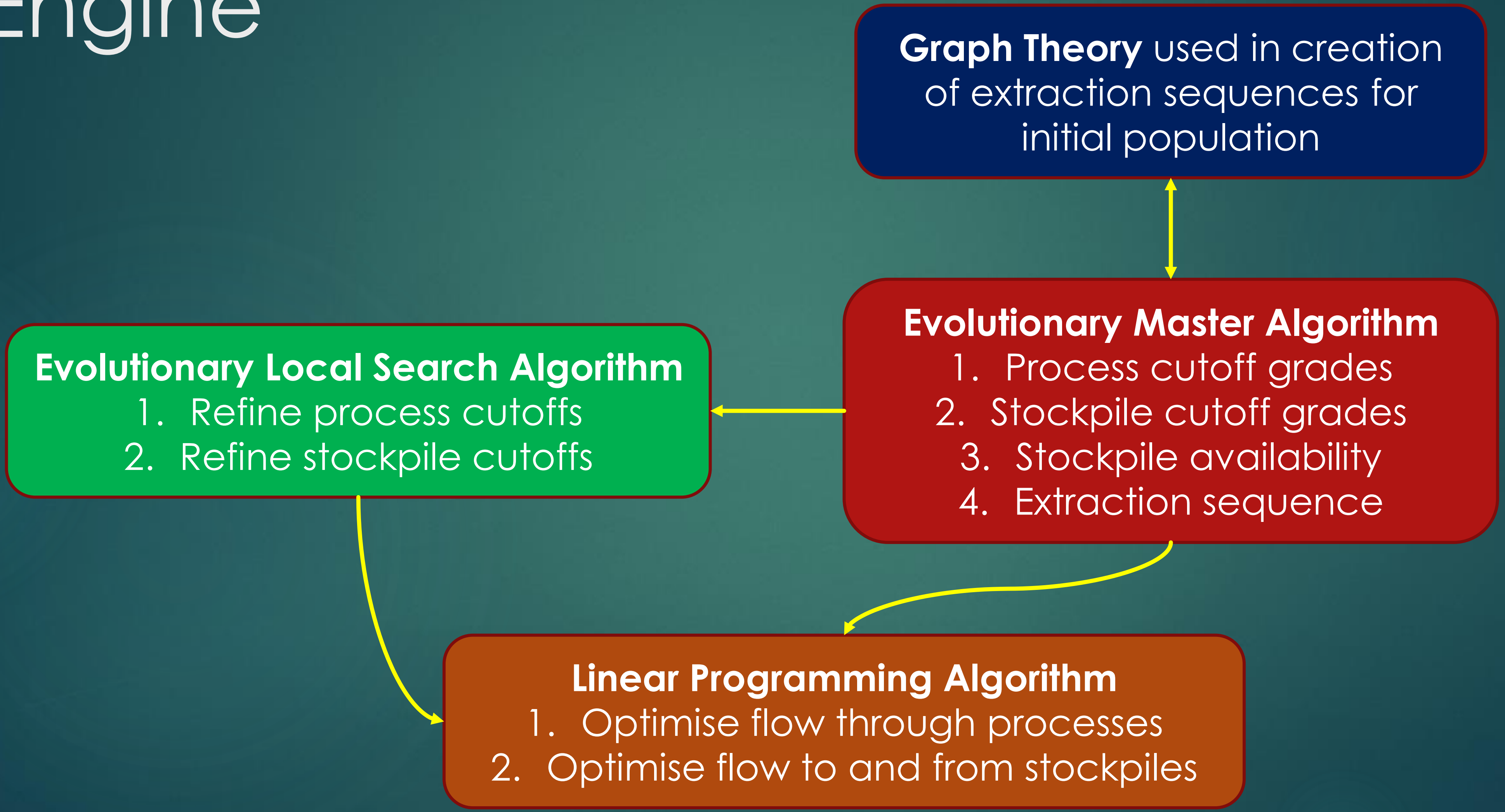
- Blocks need to be flagged in advance relative to modelled pushbacks or phases of the overall pit.
- The scheduling and optimisation functionality of Evolution is cloud-based – the block model and schedule setup are transmitted to a cloud facility for processing.
- The scheduling solutions found are transmitted back to the user for further analysis and approval.
- Optimisation is based on a hybrid system consisting of a core evolutionary algorithm, a local search evolutionary algorithm and a linear programming algorithm, each with different responsibilities.

Maptek Evolution Optimisation Engine

The engine consists of an effective hybridization of two evolutionary and one classical optimisation algorithm:

- **Master evolutionary algorithm**
  - Exploring process cut-off grade search space.
  - Exploring stockpile cut-off grade search space.
  - Exploring extraction sequence search space.
  - Manage Local Search Evolutionary algorithm.
  - Manage Linear Programming Algorithm.
- **Local search evolutionary algorithm**
  - Exploring the immediate neighbourhood of process and stockpile cut-off space for a given extraction sequence.
- **Linear programming algorithm**
  - Optimises the flow of material through available processes.
  - Responsible for optimal reclaim strategy from stockpiles.

Maptek Evolution Optimisation Engine



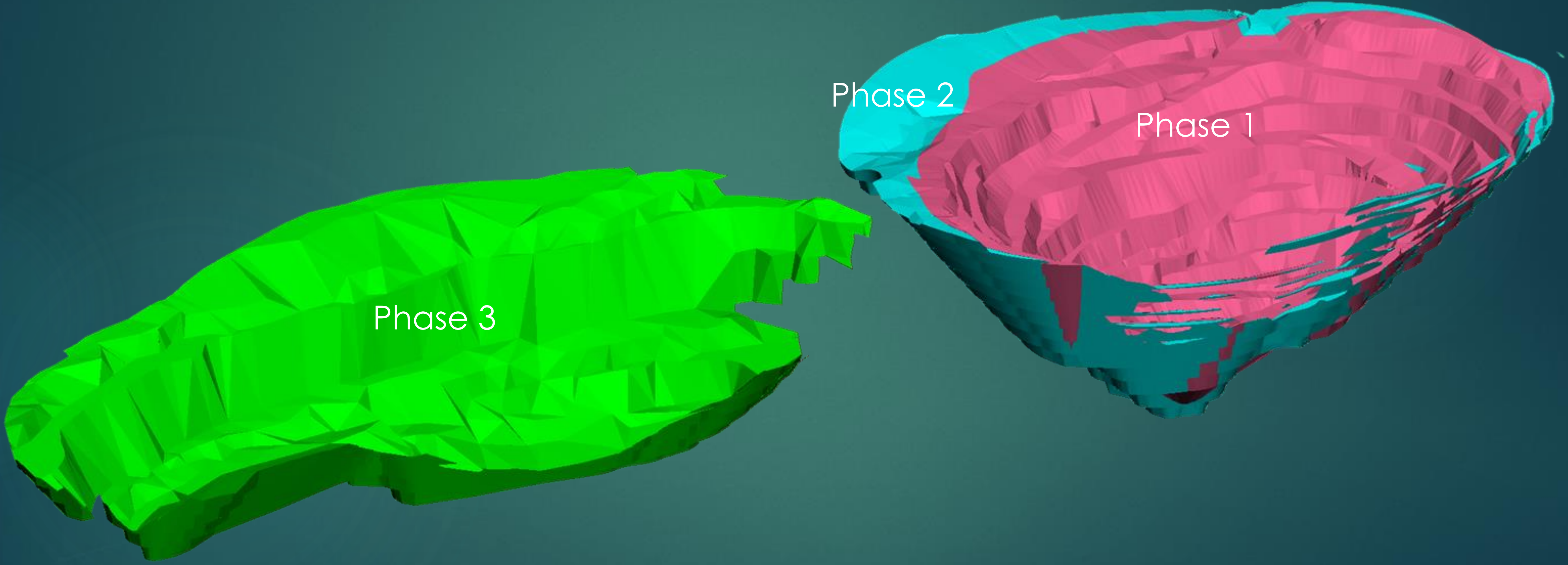
Maptek Evolution Optimisation Steps

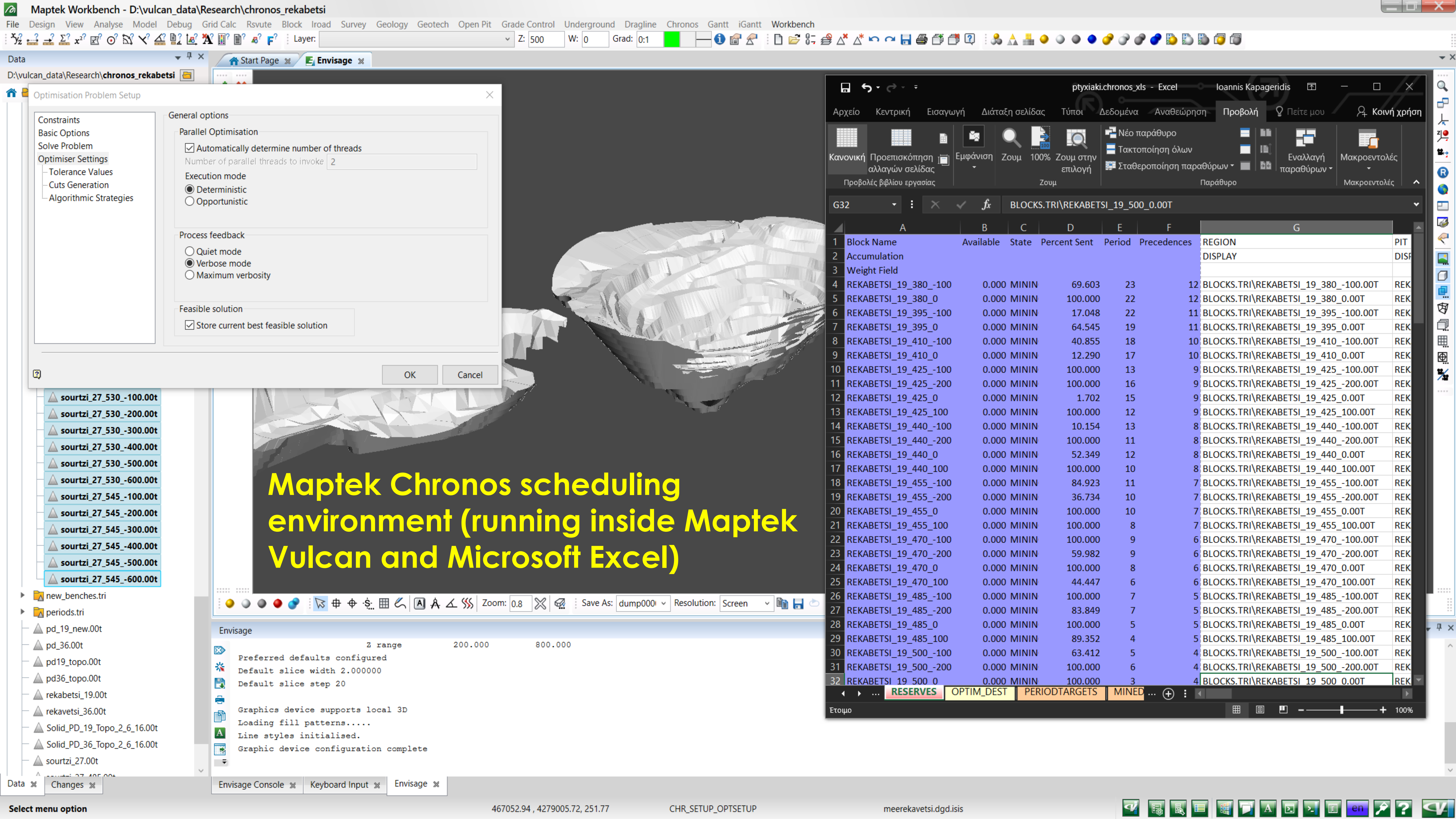
1. Creation of the initial population including a geometrically correct extraction sequence. (**Graph Theory**)
2. Calculation of the fitness of each individual and ranking of the population based on fitness (NPV). (**Master and Local Search Evolutionary Algorithms**)
3. Iteration through successive generations by generating an offspring population where each child competes with the parents for the privilege to progress to the next generation. (**Master Evolutionary Algorithm**)
4. The master algorithm calls on the secondary local search algorithm to boost the best individual found so far, by manipulating the threads through cut-off grade space whilst keeping the extraction sequence static. The improved individual is then sent back to the master where it replaces or upgrades its old self (analogue to exploring the local neighbourhood). (**Local Search Evolutionary Algorithm**)
5. Steps 2 to 4 are repeated until no improvement in NPV is registered, in other words when the population loses diversity and converges on a single high quality NPV.

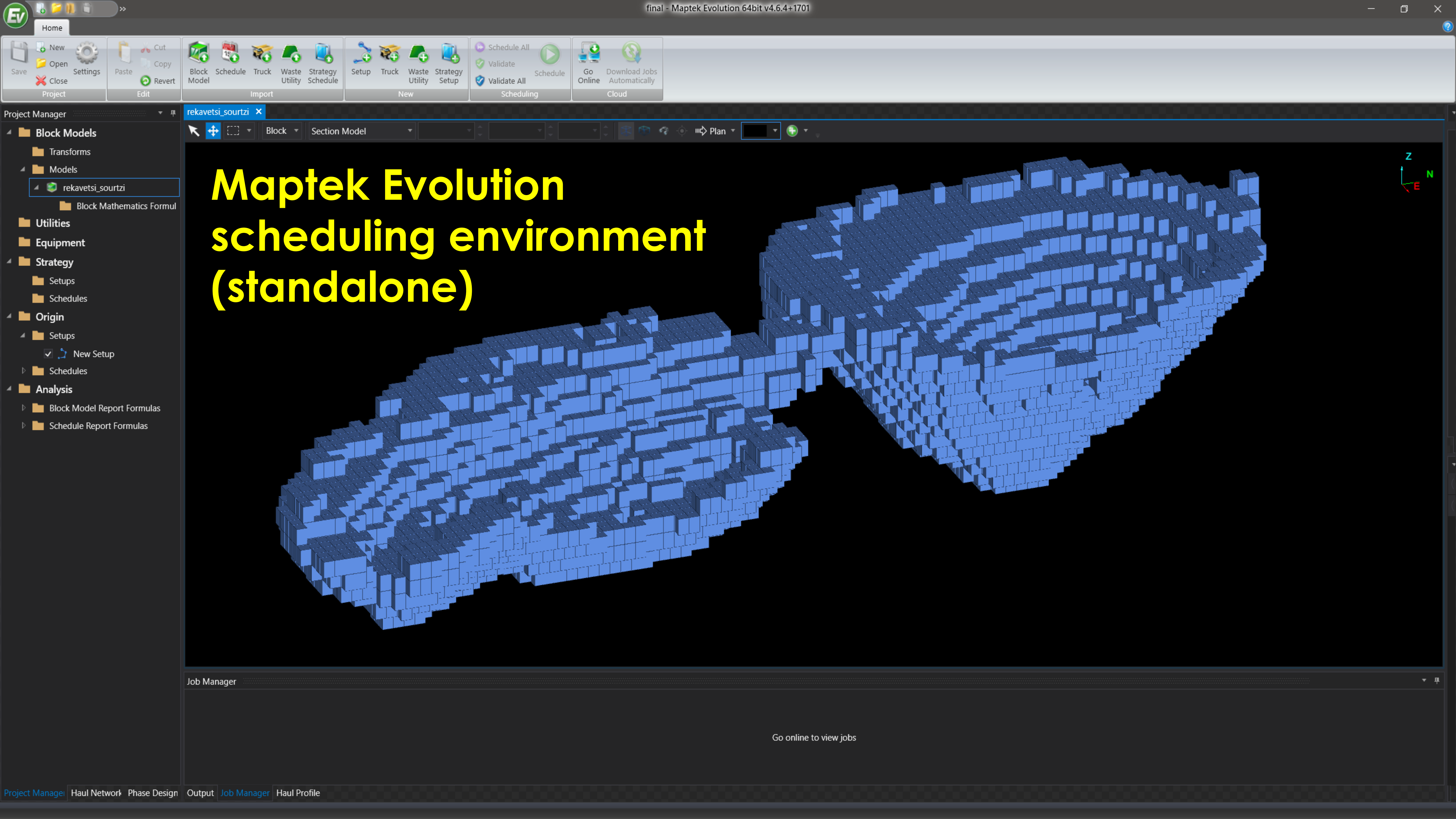
Case Study Details

- For the comparison study, two adjacent nickel open pit mines from central Greece were selected, which are mined in parallel.
- They have both been operational for some time.
- The material left for mining in the first is split in two pushbacks while the second is considered as a single phase.
- The corresponding solid models of the three pushbacks were split into benches and mining blocks, for Chronos, and at the same time they were used to flag the blocks in the resource block model for Evolution.
- Appropriate reserve fields were calculated to ensure that the same quantities and qualities are scheduled in both cases, as Evolution would be scheduling blocks and not solids.
- The schedule setups were as similar as possible between the two systems to ensure that the optimisation algorithms will be compared on as equal basis as possible.

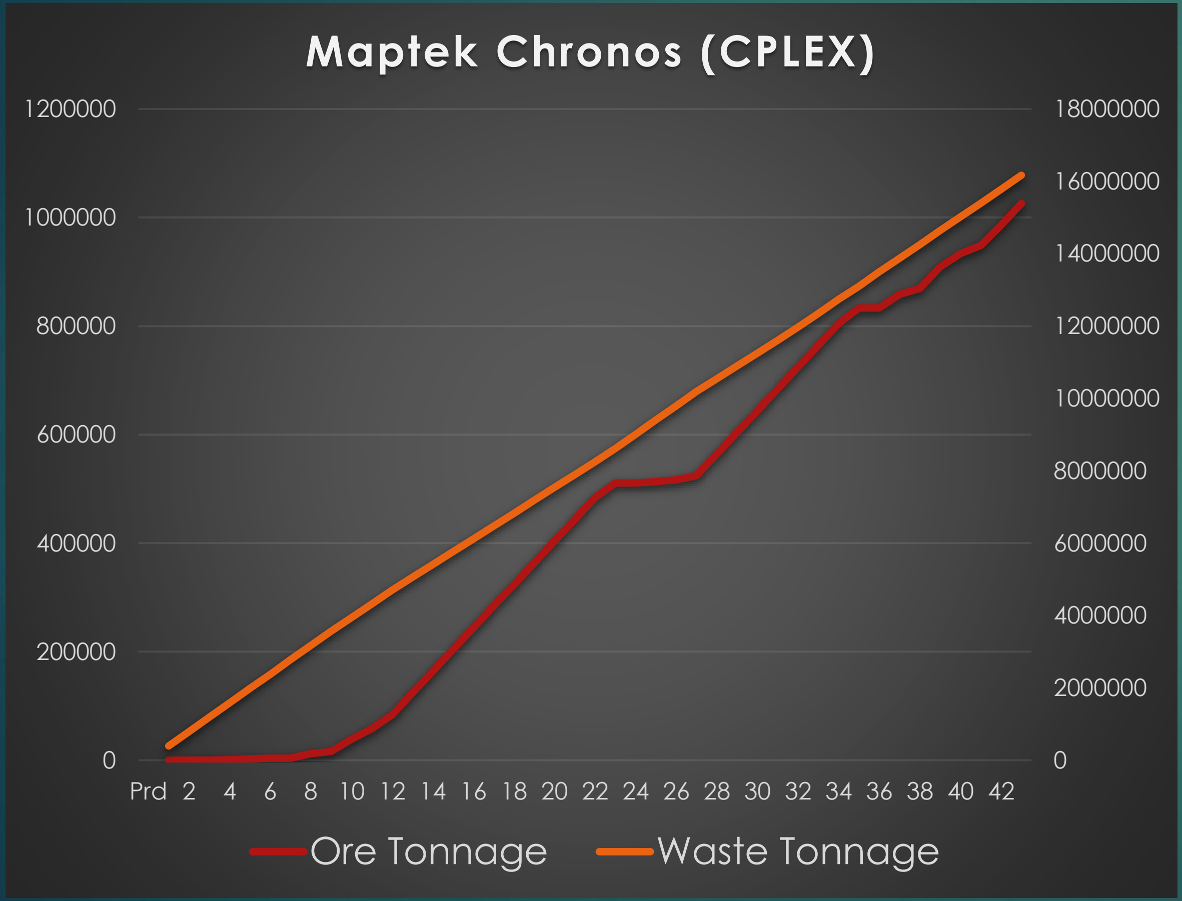
Case Study – Open Pits

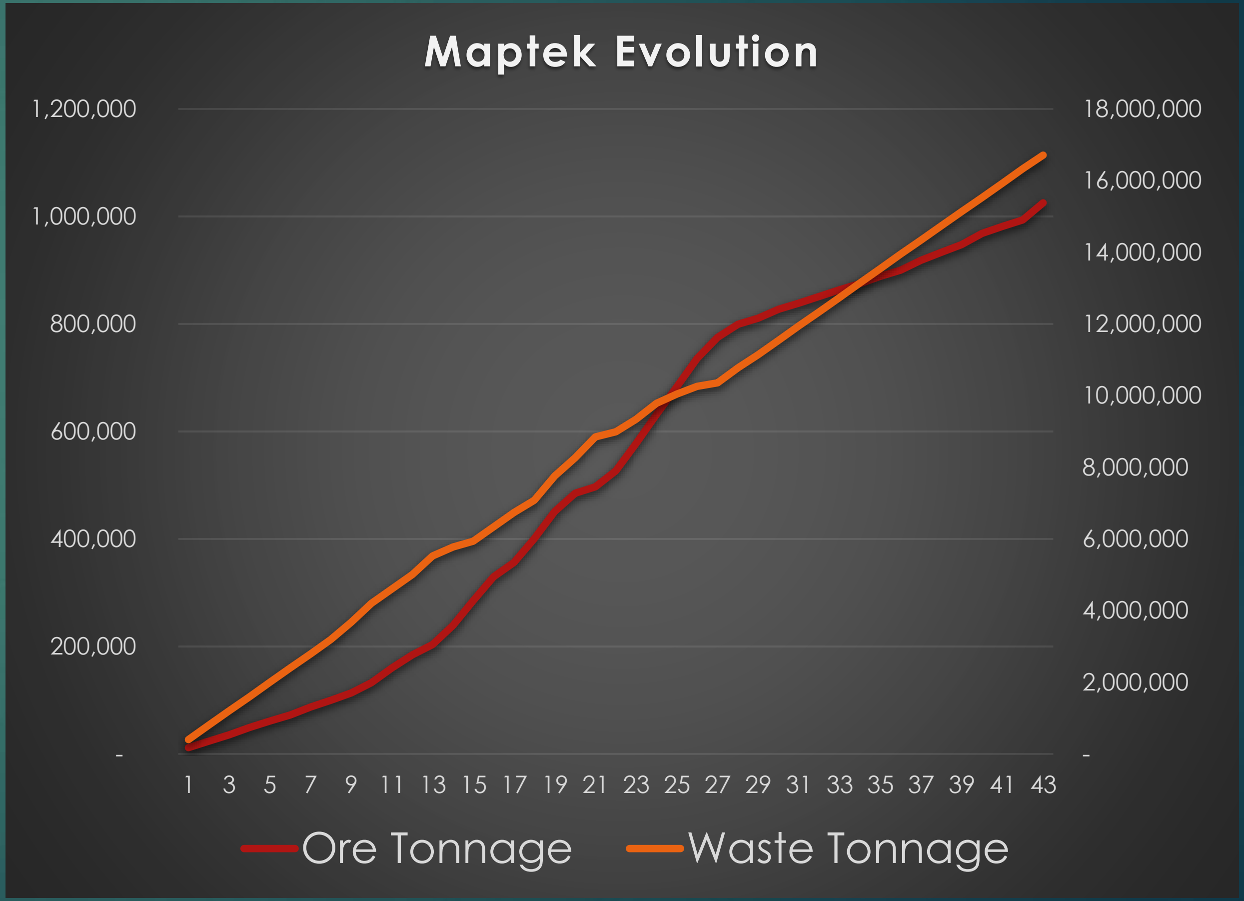



Maptek Workbench - D:\vulcan_data\Research\chronos_rekabetsi
Optimisation Problem Setup
Constraints
Basic Options
Solve Problem
Optimiser Settings
Tolerance Values
Cuts Generation
Algorithmic Strategies
General options
Parallel Optimisation
Automatically determine number of threads
Number of parallel threads to invoke 2
Execution mode
Deterministic
Opportunistic
Process feedback
Quiet mode
Verbose mode
Maximum verbosity
Feasible solution
Store current best feasible solution
OK
Cancel
ptyxiaki.chronos_xls - Excel
Maptek Chronos scheduling environment (running inside Maptek Vulcan and Microsoft Excel)


Maptek Evolution
scheduling environment
(standalone)
final - Maptek Evolution 64bit v4.6.4+1701
Home
Save
New
Open
Close
Settings
Project
Paste
Cut
Copy
Revert
Edit
Block Model
Schedule
Truck
Waste Utility
Strategy Schedule
Import
Setup
Truck
Waste Utility
Strategy Setup
New
Schedule All
Validate
Validate All
Schedule
Scheduling
Go Online
Download Jobs Automatically
Cloud
Project Manager
Block Models
Transforms
Models
rekavetsi_sourtzi
Block Mathematics Formul
Utilities
Equipment
Strategy
Setups
Schedules
Origin
Setups
New Setup
Schedules
Analysis
Block Model Report Formulas
Schedule Report Formulas
rekavetsi_sourtzi
Block
Section Model
Plan
Z
N
E
Job Manager
Go online to view jobs
Project Manager
Haul Network
Phase Design
Output
Job Manager
Haul Profile

Schedule Comparison





Cumulative figures

Criterion	Maptek Evolution	Maptek Chronos – CPLEX
1. Mining Reserves Importing - Representation	Easier and more direct importing as it is a Block Scheduler	More effort and time required as mining blocks have to be constructed using triangulation solids but produced blocks are more representative of original open pit geometry
2. Schedule Setup	Easier to setup as it is a solely open pit scheduler	Harder to setup as nothing is preset – Chronos can be used for both open pit and underground mines. The need for two packages (Vulcan/Chronos and Excel) also adds complication.
3. User Friendliness	Very user friendly even though some commands/options take some time to locate. Help facility is not very well developed.	Chronos has a very old user interface. The use of Excel as a database system helps to some extent. The help facility is more complete than Evolution.
4. Environment Speed	The environment speed of Evolution is very good.	Chronos lacks speed as it depends on the communication of different subsystems with Vulcan and Excel.
5. Mining Constraints	Evolution has ready tools for the definition of mining schedule constraints.	Chronos has special tools for the definition of mining schedule constraints but they require more setup in general.
6. Ability to Change	Any changes require rerunning of optimisation. The result of each run is stored separately which allows for the execution of any number of scenarios without having to rollback the current one.	Any changes require unscheduling everything or storing several unscheduled files before scheduling.

Criterion	Maptek Evolution	Maptek Chronos – CPLEX
7. Algorithm Manipulation Capabilities	There are no particular guidelines for the adaptation of algorithm parameters.	Requires deep knowledge of the particular optimisation algorithm and operation research theory.
8. Algorithm Speed	The advantage of using a HPC service over the cloud to optimise leads to very high speeds.	The speed of executing the optimisation algorithm depends on the capabilities of the local computer.
9. Reporting Capabilities	The available reporting functionality has great flexibility but requires effort.	Chronos has ready options for reporting and being based on Excel, takes advantage of the available options for graphs and pivot tables.
10. Visualisation	It has its own graphical environment that allows static and animated visulisation of the schedule.	Integrates with Vulcan allowing the visualisation of schedule information together with any other geological and mining models and data.
11. Effectiveness	Comparing the produced schedules, it is clear that both packages produce optimum sequences and most differences come from the different representation of the reserves available for scheduling (triangulation solids vs. resource blocks).	

Conclusions

- The comparison focused on criteria such as the value of the produced schedules, time and ease of setup, interactivity, ability to make quick changes and rerun, time required to produce a solution, and schedule reporting and visualisation capabilities.
- The study produced useful conclusions for the effectiveness and robustness of the two schedule optimisation approaches.
- Maptek Evolution proved that it can match the effectiveness of traditional optimisation algorithms.

Thank you for your attention!