A stochastic cut-off grade optimization model to incorporate uncertainty for improved project value

by J. Githiria¹ and C. Musingwini¹

Synopsis
Cut-off grade is a decision-making criterion often used for determining the quantities of material (ore and waste) to be mined, ore processed, and saleable product. It therefore directly affects the cash flows from a mining operation and the net present value (NPV) of a mining project. A series of different cut-off grades that are applied over the life of mine (LOM) of an operation defines a cut-off grade policy. Due to the complexity of the calculation process, previous work on cut-off grade calculation has mostly focused on deterministic approaches. However, deterministic approaches fail to capture the uncertainty inherent in input parameters such as commodity price and grade-tonnage distribution. This paper presents a stochastic cut-off grade optimization model that extends Lane’s deterministic theory for calculating optimal cut-off grades over the LOM. The model, code-named ‘NPVMining’, uses realistic grade-tonnage realizations and commodity price distribution to account for uncertainty. NPVMining was applied to a gold mine case study and produced an NPV ranging between 7% and 186% higher than NPVs from deterministic approaches, thus demonstrating improved project value from using stochastic optimization approaches.

Keywords
optimization, cut-off grade policy, deterministic approach, heuristic approach, stochastic approach, grade-tonnage realization, uncertainty.

Background on cut-off grade calculation and optimization
Cut-off grade is a decision-making criterion that is generally used in mining to distinguish ore from waste material. Consequently, it is used to determine the quantities of material (ore and waste) to be mined, ore processed, and saleable product. A series of different cut-off grades that are applied over the life-of-mine (LOM) of a mining operation defines the cut-off grade policy for that particular operation. The cut-off grade therefore directly affects the cash flows to be produced and the net present value (NPV) of a project at the mine planning or feasibility study stage.

Pioneering work on cut-off grade calculation can be attributed to Mortimer’s (1950) work on grade control for gold mines in South Africa. Although Mortimer’s work is given relatively little recognition, it established the fundamental principle that not only must rock at the lowest grade cover its cost of extraction, but that the average grade of the rock must provide a certain minimum profit per ton processed.

Later in the 1960s, work on cut-off grade calculation again appeared, including that published by Henning (1963), Lane (1964), and Johnson (1969). Lane (1988) subsequently published an updated version of his 1964 work as a comprehensive book on the use of cut-off grade to economically define ore using NPV as a proxy for value. To date, it is Lane’s work on value-based cut-off grade optimization that has received the most attention among the mining fraternity. Lane’s work placed more emphasis on optimizing cut-off grade in order to improve the economic viability of mining projects and operations. The cut-off grade algorithm developed by Lane (1964, 1988) was more elaborate than others as it took into account constraints associated with the capacities of the mine, mill, and market, resulting in the derivation of six potential cut-off grades from which an optimal cut-off grade could be selected. Three of the six cut-off grades are described as limiting cut-off grades while the other three are denoted as balancing cut-off grades. Limiting cut-off grades are derived by assuming that each of the three stages (mining, processing, and refining) is an individual and independent constraint on throughput due to production capacity limitations, operating costs, and price attributable to the output product. Balancing cut-off grades are determined by assuming that two out of the three stages are concurrently operating at their capacity limits.

Despite its fairly comprehensive structure, Lane’s cut-off grade optimization algorithm had some shortcomings. For example, it could not be used to determine cut-off grades for polymetallic deposits. This shortcoming is now addressed through the concept of net smelter.
A stochastic cut-off grade optimization model to incorporate uncertainty for improved project value

return (NSR) for evaluating polymetallic deposits, as for example in the work of Shava and Musingwini (2018) who developed an NSR model for a zinc, lead, and silver mine. Shava and Musingwini (2018) then related the NSR values to the applicable cut-off grades for each of the three constituent metals. Table I summarizes some studies that have attempted to address different shortcomings in Lane’s cut-off grade theory.

Despite making improvements to Lane’s original cut-off grade theory, the models in Table I are deterministic and therefore fail to capture the economic, technical, and geological uncertainties that mining operations continually face. The uncertainties include those associated with commodity price and grade-tonnage distribution. Due to this shortcoming, the NPVs generated from these models are sub-optimal. There is, therefore, a need for a stochastic cut-off grade optimization approach that can capture uncertainty in parameters such as commodity price and grade-tonnage distribution. This challenge had been long-recognized in other studies not related to Lane’s framework, but which attempted to use other stochastic and/or dynamic programming (DP) approaches to address the challenge.

Stochastic and dynamic programming approaches to cut-off grade

Several studies that have applied stochastic and/or DP approaches in the calculation of cut-off grades have incorporated the dynamic nature of input parameters. Table II summarizes some studies that have attempted to address uncertainty in parameters for cut-off grade determination.

The studies summarized in Table II, except for the model by Asad and Dimitrakopoulos (2013), were not based on Lane’s framework and tended to consider only one input parameter as being stochastic. The study presented in this paper therefore extended Lane’s algorithm to develop a stochastic cut-off grade model by concurrently considering variability in both commodity price and grade-tonnage distribution. The model that was developed is code-named ‘NPVMining’.

Modifications to Lane’s cut-off grade theory for the stochastic NPVMining model

Lane’s theoretical framework is premised on a schematic material flow as illustrated in Figure 1. Depending on the applicable cut-off grade, material from the mine can be classified as waste and sent to the waste dump; as ore and sent for milling in the processing plant; or as low-grade material and sent to a stockpile for processing later during the LOM. The milling process produces a concentrate which is sent to the refinery to produce the final saleable product, which is then marketed.

The production capacities of the different stages in the mining complex are denoted by M, C, and R, for the mining, milling, and refinery capacities, respectively. Lane’s framework uses the notations in Table III. The input parameters in Table III were then modified to account for variability as explained in the following sections.

Given a set of equally probable grade-tonnage curves (w), the stochastic approach develops a cut-off grade policy by determining the cut-off grade (G) from time periods 1 to N. The NPV of future cash flows is maximized subject to mining, processing, and refining capacity constraints. The objective function for the cut-off grade optimization remains unchanged from Lane’s original formulation, as represented by Equation [1].

\[
\text{Max NPV} = \sum_{t=1}^{N} \frac{P W_t}{(1 + d)^t}
\]

Table I

Examples of studies addressing some shortcomings in Lane’s cut-off grade theory

<table>
<thead>
<tr>
<th>Study</th>
<th>Summary of improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taylor (1972)</td>
<td>Taylor’s approach for balancing cut-off grades used statistical parameters to describe the grade distribution of the orebody, since grade is variable throughout an orebody.</td>
</tr>
<tr>
<td>Taylor (1985)</td>
<td>The approach incorporated a stockpiling stage into Lane’s framework so that instead of dumping low-grade material as waste, it is kept as stockpiles as future ore feed to the mill.</td>
</tr>
<tr>
<td>Dagdelen (1992, 1993)</td>
<td>Dagdelen developed an analytical method for finding balancing cut-off grades which was more efficient than Lane’s graphical.</td>
</tr>
<tr>
<td>Whittle and Wharton (1995)</td>
<td>The approach, which is incorporated in the open-pit mining software Geovia Whittle 4-X®, added linear programming (LP) optimization into Lane’s theory to simultaneously incorporate stockpiling of mined material, ore blending, and account for multiple minerals.</td>
</tr>
<tr>
<td>Asad (1997, 2005)</td>
<td>Asad developed a cut-off grade optimization algorithm based on Lane’s work, but with a stockpiling option for open-pit mining operations with two economic minerals.</td>
</tr>
<tr>
<td>King (2001)</td>
<td>King modified Lane’s approach by incorporating variations in throughput for different ore types in order to cater for polymetallic deposits.</td>
</tr>
<tr>
<td>Dagdelen and Kawahata (2008)</td>
<td>The approach applied mixed-integer linear programming (MILP) to improve the efficiency of the calculation process in Lane’s algorithm and was used develop the OptPit®-mining software package.</td>
</tr>
<tr>
<td>Gholamnejad (2008, 2009)</td>
<td>Gholamnejad modified Lane’s algorithm to cater for the trade-off between an increase in the average mill grade and a concomitant increase in rehabilitation costs. As the cut-off grade is increased, the amount of material mined and dumped as waste also increases, leading to increased rehabilitation costs.</td>
</tr>
<tr>
<td>Osantio, Rashidnejad, and Rezai (2008)</td>
<td>The study incorporated environmental requirements into Lane’s cut-off grade optimization by incorporating waste and tailings disposal costs.</td>
</tr>
<tr>
<td>King (2011)</td>
<td>King modified the earlier (King, 2001) version by incorporating operating and administrative costs.</td>
</tr>
<tr>
<td>Uthina, Munuki, and Musingwini (2016)</td>
<td>The study developed a computer-aided application based on Lane’s algorithm and improved the efficiency of the cut-off grade calculation process.</td>
</tr>
</tbody>
</table>
A stochastic cut-off grade optimization model to incorporate uncertainty for improved project value

Table II
Examples of studies addressing uncertainty in cut-off grade determination

<table>
<thead>
<tr>
<th>Study</th>
<th>Summary of stochastic or DP approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dowd (1976)</td>
<td>The approach was based on DP to optimize cut-off grade. It incorporated the interaction of cut-off grades and fluctuating commodity prices, but ignored variability in costs.</td>
</tr>
<tr>
<td>Krautkraemer (1988)</td>
<td>Krautkraemer analysed the effect of stochastic metal prices on the selection of cut-off grades, depending on the rate of metal price change relative to the discount rate, and concluded that fluctuations in metal prices critically affect the selection of cut-off grades.</td>
</tr>
<tr>
<td>Cetin and Dowd (2002, 2013, 2016)</td>
<td>The approach applied genetic algorithms (GAs) and DP to optimize cut-off grades for polymetallic mines, while assuming a constant grade-tonnage distribution for the entire orebody. Cetin and Dowd (2016) compared GA, the grid search method, and DP when deriving optimal cut-off grades for deposits with up to three constituent minerals and concluded that GAs are more robust in optimizing cut-off grade for multi-mineral deposits under technical and economic constraints only.</td>
</tr>
<tr>
<td>Asad (2007)</td>
<td>Asad optimized cut-off grade for an open-pit mining operation through an NPV-based algorithm that considered metal price and cost escalation but ignored geological uncertainty.</td>
</tr>
<tr>
<td>Asad and Dimitrakopoulos (2013)</td>
<td>Asad and Dimitrakopoulos modified Lane’s approach into a heuristic cut-off grade model to account for geological uncertainty in ore supplied to multiple processing streams. However, the approach was limited to a single mine.</td>
</tr>
<tr>
<td>Thompson and Barr (2014)</td>
<td>The approach optimized cut-off grade using stochastic programming in open-pit mining, but considered commodity price as the only dynamic variable.</td>
</tr>
<tr>
<td>Myburgh, Deb, and Craig (2014)</td>
<td>The approach was a hybrid heuristic approach to maximize NPV through cut-off grade optimization. It is incorporated in the software package, Maptek Evolution®. The approach employs an evolutionary GA to optimize cut-off grade and price as the only dynamic variable.</td>
</tr>
</tbody>
</table>

Table III
Notations used in Lane’s algorithm (adapted from Githiria, 2018)

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>Year</td>
<td>-</td>
</tr>
<tr>
<td>N</td>
<td>Mine life</td>
<td>Years</td>
</tr>
<tr>
<td>CC</td>
<td>Capital cost</td>
<td>$/million</td>
</tr>
<tr>
<td>s</td>
<td>Selling price</td>
<td>$/g</td>
</tr>
<tr>
<td>r</td>
<td>Refining cost</td>
<td>$/oz</td>
</tr>
<tr>
<td>m</td>
<td>Mining cost</td>
<td>$/ton</td>
</tr>
<tr>
<td>f</td>
<td>Milling cost</td>
<td>$/ton</td>
</tr>
<tr>
<td>y</td>
<td>Annual fixed costs</td>
<td>$/year</td>
</tr>
<tr>
<td>y</td>
<td>Recovery</td>
<td>%</td>
</tr>
<tr>
<td>d</td>
<td>Discount rate</td>
<td>%</td>
</tr>
<tr>
<td>M</td>
<td>Mining capacity</td>
<td>t/a</td>
</tr>
<tr>
<td>C</td>
<td>Milling capacity</td>
<td>t/a</td>
</tr>
<tr>
<td>R</td>
<td>Refining capacity</td>
<td>t/a</td>
</tr>
<tr>
<td>Qm</td>
<td>Material mined</td>
<td>t/a</td>
</tr>
<tr>
<td>Qc</td>
<td>Ore processed</td>
<td>t/a</td>
</tr>
<tr>
<td>Qr</td>
<td>Concentrate refined</td>
<td>t/a</td>
</tr>
</tbody>
</table>

where $d$ is the discount rate and $P_{wi}$ is the cash flow generated in period $i$ by extracting the orebody based on a grade-tonnage curve $w$.

However, the cash flow equation changes to incorporate the uncertainty of both the metal price and grade-tonnage distribution in the orebody, as indicated in Equation [2].

$$Cash\ flow, P_{wi} = \sum_{i=1}^{N} (s_i - n_i) * Q_{rwi} - c_{i} * Q_{cwi} - m_{i} * Q_{mwi} - f_{i}$$

subject to: 

$$(Q_{mwi} \leq M \text{ for } i = 1 \ldots N),$$

$$(Q_{cwi} \leq C \text{ for } i = 1 \ldots N),$$

$$(Q_{rwi} \leq R \text{ for } i = 1 \ldots N).$$

Using the optimum cut-off grades obtained in the algorithm for the given grade-tonnage curve $w$, a yearly production schedule that shows the cut-off grade, quantity mined ($Q_{mwi}$), quantity processed ($Q_{cwi}$), quantity refined ($Q_{rwi}$), profit, and NPV is calculated. Figure 2 illustrates the steps involved in the algorithm used in this research study.

The algorithm developed from Figure 2 was coded using the C++ programming language in Microsoft Visual Studio 2017 Integrated Development Environment (IDE) on a standard computer to produce the application code-named ‘NPVMining’. The code for NPVMining is given in Appendix 1. The application is an executable file and will run on computers that run applications with .exe extension. The computer must have Visual Studio 2017 and Microsoft Office 2015 installed on it.

Microsoft Visual Studio 2017 (VC++) supports two versions of the C++ programming language, which are the ISO/ANSI standard C++, and C++/CLI (Common Language Infrastructure). C++/CLI has a highly-developed design capability that enables the assembly of the entire graphical user interface (GUI), and the code that creates it being generated automatically (Githiria, Muriuki, and Musingwini, 2016). The C++ programming language was used in the implementation of the algorithm to enable the execution of
A stochastic cut-off grade optimization model to incorporate uncertainty for improved project value

The application on Windows-based computers with different architectures and/or platforms. This compatibility aspect enables easy portability of the application to make it usable by mine planners working on different computing platforms. The output from the model is easily exported to Microsoft Excel 2013 for comparison and analysis.

The flow diagram in Figure 2 was then modelled as follows:

i. Identify the methods of data entry for the grade-tonnage curves, commodity price, and costs.

ii. Develop software to input the grade-tonnage curves, commodity price, and costs variations and run tests for validation.

iii. Implement the algorithm for calculating cut-off grade and NPV using the inputs provided.

iv. Test and validate the output and provide the output in a format that will display the results appropriately.

Mathematical functions of commodity price against time were applied in the above procedure. A random number generator was developed to generate values within a specified range so that input values are stochastic but within realistic ranges.

General steps in the NPV/Mining stochastic algorithm

The steps to determine a cut-off grade policy as outlined by Lane (1964), but modified as illustrated in Figure 2 to incorporate uncertainty in NPV/Mining, are as follows (Githiria, 2018):

1. Formulate multiple realizations of grade-tonnage distribution for the entire deposit.

2. Input the parameters to be used in the cut-off grade policy, such as the mining capacity (M), milling capacity (C), refining capacity (R), selling price (P), mining cost (m), milling cost (c), refining cost (r), recovery (y), annual fixed costs (f), and discount rate (d).

3. Introduce variability in the input parameters (metal price and grade-tonnage).

4. Determine the optimum cut-off grade to be used in year i using the cut-off grade equations. If the initial NPVi is not known set the NPVi to zero.

5. Determine the tons of ore (qowi), tons of waste (qww), and average grades of the ore associated w the optimum cut-off grade (qavg). Set: Qcwi = C if qowi is greater than the milling capacity (C), otherwise Qcwi = qowi. Using Equations [3]–[7], calculate the quantity to be mined (Qm) and refined (Qr). Find the limiting capacity and the mine life (N) from the following:

\[ Q_{rwi} = Q_{cwi} \times g_{avg} \times \sum g_i \]  
[3]

\[ Q_{mwi} = Q_{cwi}(1 + SR) \]  
[4]

\[ N = \frac{Q_{cwi}}{M} \]  
[5]

\[ N = \frac{Q_{cwi}}{C} \]  
[6]

\[ N = \frac{Q_{rwi}}{R} \]  
[7]

6. Determine the yearly profit using Equation [8].

\[ Cash Flow, P_{wi} = \sum_{i=1}^{N} (S_i - r_i) + Q_{rwi} - c_i \times Q_{rwi} - m_i \times Q_{mwi} - f_i \]  
[8]

7. Compute the NPV using the formula below by discounting the profits at a given discount rate (d) for the time calculated as the LOM.

\[ Max \ NPV = \sum_{i=1}^{N} \frac{P_{wi}}{(1 + d)^i} \]  
[9]

This value of V becomes the second approximation of V (the first was V=0) for use in the formulae to calculate the optimum cut-off grade.

8. Repeat the computation from step 5 until the value, V, converges.

9. Adjust the grade-tonnage distribution by subtracting the ore tons from the grade-tonnage distribution intervals above the optimum cut-off grade (G) and the waste tons (Qmwi – Qcwi) from the intervals below the optimum cut-off grade (G) in proportionate amounts. This is to ensure that the distribution remains unchanged, otherwise it will change to a different grade-tonnage distribution. For each of the multiple realizations of the grade-tonnage distribution the current grade-tonnage curve being used at that specific period will be altered simultaneously. This will cater for the intertemporal dependencies that alter the grade-tonnage distribution with time.

10. If it is the first iteration then, knowing the profits obtained in each year, find the yearly NPV by discounting back those profits and go to step 4. If it is the second iteration, then stop.

Figure 2—Flow diagram of the modified cut-off grade algorithm (Githiria, 2018)
11. Use the net present values obtained in step 10 as the initial NPV for each of the corresponding years for the second iteration.

**Brief description of NPVMining**

NPVMining caters for: (i) economic and operational parameters, (ii) grade-tonnage distribution, and (iii) the cut-off grade policy or production schedule as shown in Figure 3. One of the source files, NPVMiningDlg.cpp, contains the code that executes the algorithm through the user interface. Appendix 1 contains the code for NPVMining. The input parameters (price and costs) used in the calculations are uploaded to the application using Microsoft Excel® spreadsheets. Other parameters are entered into the user interface via the dialog boxes. The metal price variability is obtained from factoring either a fixed or geometric range. After calculating the optimum cut-off grade, a cut-off grade policy is generated showing the three main economic indicators: annual profit, NPV, and LOM. The NPVMining user interface has five dialog boxes: (i) grade category window (Figure 4), (ii) price criteria window (Figure 5), (iii) other parameters window (Figure 6), (iv) limiting capacity selection window (Figure 7), and (v) cut-off grade calculation window (Figure 7).

The grade-tonnage distribution, economic and operational parameters are the input data used in the computer-aided application. The grade-tonnage distribution is entered into the grade category window as shown in Figure 4. The grade category input window has several dialog boxes describing the lower grade limit, upper grade limit, and quantity of ore per increment for the multiple realizations.

The economic and operational parameters are keyed on the user interface using the price criteria and other parameters window as shown in Figures 5 and 6. This data is uploaded using Microsoft Excel® worksheets containing technical data, while other data is entered in the dialog box as required. This process simplifies data entry, which is tiresome if done manually. The two input windows in Figures 5 and 6 have several dialog boxes that represent economic and operational parameters such as metal price, mining cost, milling cost, refining cost, mining capacity, processing capacity, refining capacity, recovery, discount rate, and fixed cost.
A stochastic cut-off grade optimization model to incorporate uncertainty for improved project value

The break-even cut-off grade and Lane’s limiting cut-off grade are selected and calculated as shown in Figure 7, and the output is displayed in the output window as shown in Figure 8. The main economic indicators of the project (annual profit, NPV, and LOM) are displayed on the user interface. After clicking the ‘calculate’ button, the results of the cut-off grade policy calculations are displayed in the output window and generated on a Microsoft Excel® spreadsheet as shown in Figure 8. The spreadsheet consists of seven main columns: optimum cut-off grade, quantity of material to be mined \((Q_m)\), quantity of ore to be milled \((Q_c)\), quantity of product refined \((Q_r)\), mine life, annual profit, and NPV. The resultant cut-off grade policy is exported to Microsoft Excel® for ease of use and interpretation.

The NPVMining application was tested on the data-set for a case study of the McLaughlin gold mine, which is a defunct gold mine in northern California, USA. The data-set was obtained from the mining library compiled by Espinoza et al. (2012) for research purposes. This data-set was selected because it has previously been used in other cut-off grade optimization studies, therefore allowing comparison of the output from NPVMining with those studies.

Description of the McLaughlin gold mine case study data-set

Tables IV and V provide the simulated grade-tonnage distribution and the economic, design, and operational parameters, respectively, for the McLaughlin gold mine case.

![Figure 7—Selection of the limiting capacity and cut-off grade policy used (Githiria, 2018)](image)

![Figure 8—Output exported to a Microsoft Excel® spreadsheet (Githiria, 2018)](image)

### Table IV

**Simulated grade-tonnage distribution of the McLaughlin gold deposit (Githiria, 2018)**

<table>
<thead>
<tr>
<th>Grade (oz/t)</th>
<th>GTC1</th>
<th>GTC2</th>
<th>GTC3</th>
<th>GTC4</th>
<th>GTC5</th>
<th>GTC6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000-0.020</td>
<td>70 000</td>
<td>70 022</td>
<td>70 081</td>
<td>70 003</td>
<td>69 640</td>
<td>70 900</td>
</tr>
<tr>
<td>0.020-0.025</td>
<td>7 257</td>
<td>7 263</td>
<td>7 257</td>
<td>7 364</td>
<td>7 890</td>
<td>7 000</td>
</tr>
<tr>
<td>0.025-0.030</td>
<td>6 319</td>
<td>6 325</td>
<td>6 332</td>
<td>6 405</td>
<td>6 860</td>
<td>5 700</td>
</tr>
<tr>
<td>0.030-0.035</td>
<td>5 591</td>
<td>5 595</td>
<td>5 616</td>
<td>5 632</td>
<td>6 100</td>
<td>5 200</td>
</tr>
<tr>
<td>0.035-0.040</td>
<td>4 598</td>
<td>4 600</td>
<td>4 628</td>
<td>4 608</td>
<td>4 950</td>
<td>4 600</td>
</tr>
<tr>
<td>0.040-0.045</td>
<td>4 277</td>
<td>4 279</td>
<td>4 306</td>
<td>4 318</td>
<td>4 570</td>
<td>4 300</td>
</tr>
<tr>
<td>0.045-0.050</td>
<td>3 465</td>
<td>3 466</td>
<td>3 488</td>
<td>3 502</td>
<td>3 680</td>
<td>3 000</td>
</tr>
<tr>
<td>0.050-0.055</td>
<td>2 438</td>
<td>2 439</td>
<td>2 468</td>
<td>2 438</td>
<td>2 630</td>
<td>2 200</td>
</tr>
<tr>
<td>0.055-0.060</td>
<td>2 307</td>
<td>2 309</td>
<td>2 335</td>
<td>2 318</td>
<td>2 520</td>
<td>2 100</td>
</tr>
<tr>
<td>0.060-0.065</td>
<td>1 747</td>
<td>1 748</td>
<td>1 754</td>
<td>1 740</td>
<td>1 900</td>
<td>1 500</td>
</tr>
<tr>
<td>0.065-0.070</td>
<td>1 640</td>
<td>1 643</td>
<td>1 647</td>
<td>1 637</td>
<td>1 770</td>
<td>1 400</td>
</tr>
<tr>
<td>0.070-0.075</td>
<td>1 485</td>
<td>1 488</td>
<td>1 494</td>
<td>1 481</td>
<td>1 560</td>
<td>1 200</td>
</tr>
<tr>
<td>0.075-0.080</td>
<td>1 227</td>
<td>1 229</td>
<td>1 237</td>
<td>1 246</td>
<td>1 270</td>
<td>800</td>
</tr>
<tr>
<td>0.080-0.100</td>
<td>3 598</td>
<td>3 598</td>
<td>3 625</td>
<td>3 636</td>
<td>3 810</td>
<td>3 300</td>
</tr>
<tr>
<td>0.100-0.150</td>
<td>9 574</td>
<td>9 578</td>
<td>9 579</td>
<td>9 641</td>
<td>10 550</td>
<td>9 600</td>
</tr>
<tr>
<td>Total</td>
<td>125 523</td>
<td>125 585</td>
<td>125 847</td>
<td>125 969</td>
<td>129 480</td>
<td>123 000</td>
</tr>
</tbody>
</table>

### Table V

**Economic, design, and operational parameters used in the model (Githiria, 2018)**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
<th>Unit</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>Capital cost</td>
<td>$ million</td>
<td>105 000</td>
</tr>
<tr>
<td>m</td>
<td>Mining cost</td>
<td>$/ton</td>
<td>1.20</td>
</tr>
<tr>
<td>c</td>
<td>Milling cost</td>
<td>$/ton</td>
<td>19.00</td>
</tr>
<tr>
<td>r</td>
<td>Refining/Selling cost</td>
<td>$/oz</td>
<td>655.00</td>
</tr>
<tr>
<td>f</td>
<td>Annual fixed costs</td>
<td>$/year</td>
<td>8 350 000</td>
</tr>
<tr>
<td>y</td>
<td>Recovery</td>
<td>Decimal</td>
<td>0.90</td>
</tr>
<tr>
<td>d</td>
<td>Discount rate</td>
<td>Decimal</td>
<td>0.15</td>
</tr>
<tr>
<td>M</td>
<td>Mining capacity</td>
<td>t/a</td>
<td>Unlimited</td>
</tr>
<tr>
<td>C</td>
<td>Milling capacity</td>
<td>t/a</td>
<td>Unlimited</td>
</tr>
<tr>
<td>R</td>
<td>Refining capacity</td>
<td>t/a</td>
<td>Unlimited</td>
</tr>
</tbody>
</table>
A stochastic cut-off grade optimization model to incorporate uncertainty for improved project value

study. Table VI provides the price variations used in the calculations.

Six grade-tonnage curves (GTC1 to GTC6) for the McLaughlin deposit, as illustrated in Table IV, were used to determine balancing cut-off grades as described in Lane’s cut-off grade theory (Lane, 1964, 1988). The grade-tonnage curve data was used to calculate the ratios and cumulative values in relation to the different stages in a mining complex. Over-estimation or under-estimation of the grade, volume, or tonnage and other parameters related to a deposit is common in most conventional and deterministic orebody models. This is detrimental to the planning of the mining operation, which consequently leads to loss of profits. There are several statistical methods for measuring uncertainty of the orebody in relation to the geological characteristics. Stochastic approaches are employed to characterize the geological uncertainty by modelling and estimating the orebody more reliably. This study employed the Monte Carlo method to simulate the grade-tonnage distribution of the orebody. It used statistical and graphical techniques, including linear and nonlinear modelling, to simulate the probable distribution of the data. It is evident from the simulated multiple realizations of the orebody that the tonnages decrease with increasing grade ranges.

The McLaughlin mining operation case study incorporates a mine, processing plant, and waste dump, following the three main stages presented in Figure 1, excluding the stockpiling option. The mine produces sulphide and oxides ores mixed with waste material in controlled quantities. The ore goes through several processing stages such as gravity concentration, flotation, and leaching. In the subsequent refining step, the gold is recovered from solution such as gravity concentration, flotation, and leaching. In the following year, the grade–tonnage curves are used to determine the optimum cut-off grade that maximizes NPV in approximately 9 years for the six grade-tonnage curves. The application (NPVMining) also generates possible outcomes in relation to the change in gold price. The six possible outcomes for each price category generate different outcome as shown in Table VII. The resultant cut-off grade policies are generated by varying the grade–tonnage curves for all mineable ore in every grade interval and the metal prices against time. The solution was generated in 14.45 seconds after running the NPVMining code on Microsoft Visual Studio 2017.

The relationship between NPV and the metal price was modelled using a linear regression approach (Figures 9 and 10). The linear regression model shown in Figure 9 is applied to identify the relationship between NPV and the response variables (metal price) when all the other variables in the model are held fixed. The correlation coefficient in the relationship between metal price and NPV is about 0.97, indicating that metal price variation has a high significance for the NPV.

Table VII

<table>
<thead>
<tr>
<th>NPV ($)</th>
<th>Price (oz/ton)</th>
<th>Grade-tonnage (kt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>467 172 060.50</td>
<td>1250</td>
<td>125 523</td>
</tr>
<tr>
<td>467 172 060.49</td>
<td>1250</td>
<td>125 585</td>
</tr>
<tr>
<td>466 990 510.59</td>
<td>1250</td>
<td>125 847</td>
</tr>
<tr>
<td>468 922 666.82</td>
<td>1250</td>
<td>125 969</td>
</tr>
<tr>
<td>486 180 335.90</td>
<td>1250</td>
<td>129 480</td>
</tr>
<tr>
<td>475 864 529.70</td>
<td>1250</td>
<td>132 000</td>
</tr>
<tr>
<td>487 959 524.30</td>
<td>1270</td>
<td>125 523</td>
</tr>
<tr>
<td>487 959 524.30</td>
<td>1270</td>
<td>125 585</td>
</tr>
<tr>
<td>487 777 974.39</td>
<td>1270</td>
<td>125 847</td>
</tr>
<tr>
<td>489 777 241.30</td>
<td>1270</td>
<td>125 969</td>
</tr>
<tr>
<td>509 740 604.86</td>
<td>1270</td>
<td>129 480</td>
</tr>
<tr>
<td>496 886 282.14</td>
<td>1270</td>
<td>132 000</td>
</tr>
<tr>
<td>508 746 988.10</td>
<td>1290</td>
<td>125 523</td>
</tr>
<tr>
<td>508 746 988.10</td>
<td>1290</td>
<td>125 585</td>
</tr>
<tr>
<td>508 565 438.18</td>
<td>1290</td>
<td>125 847</td>
</tr>
<tr>
<td>510 631 795.78</td>
<td>1290</td>
<td>125 969</td>
</tr>
<tr>
<td>531 300 873.82</td>
<td>1290</td>
<td>129 480</td>
</tr>
<tr>
<td>517 908 034.58</td>
<td>1290</td>
<td>123 000</td>
</tr>
<tr>
<td>529 534 451.90</td>
<td>1310</td>
<td>125 523</td>
</tr>
<tr>
<td>529 534 451.88</td>
<td>1310</td>
<td>125 585</td>
</tr>
<tr>
<td>529 532 901.88</td>
<td>1310</td>
<td>125 847</td>
</tr>
<tr>
<td>531 486 350.25</td>
<td>1310</td>
<td>125 969</td>
</tr>
<tr>
<td>552 861 142.78</td>
<td>1310</td>
<td>129 480</td>
</tr>
<tr>
<td>538 929 787.03</td>
<td>1310</td>
<td>123 000</td>
</tr>
<tr>
<td>550 321 915.70</td>
<td>1330</td>
<td>125 523</td>
</tr>
<tr>
<td>550 321 915.68</td>
<td>1330</td>
<td>125 585</td>
</tr>
<tr>
<td>550 140 365.78</td>
<td>1330</td>
<td>125 847</td>
</tr>
<tr>
<td>552 340 904.73</td>
<td>1330</td>
<td>125 969</td>
</tr>
<tr>
<td>574 421 114.74</td>
<td>1330</td>
<td>129 480</td>
</tr>
<tr>
<td>559 951 539.47</td>
<td>1330</td>
<td>123 000</td>
</tr>
<tr>
<td>571 109 379.50</td>
<td>1330</td>
<td>125 523</td>
</tr>
<tr>
<td>571 109 379.47</td>
<td>1330</td>
<td>125 585</td>
</tr>
<tr>
<td>570 927 829.57</td>
<td>1330</td>
<td>125 847</td>
</tr>
<tr>
<td>573 195 459.21</td>
<td>1330</td>
<td>125 969</td>
</tr>
<tr>
<td>595 981 680.69</td>
<td>1330</td>
<td>129 480</td>
</tr>
<tr>
<td>580 973 291.91</td>
<td>1330</td>
<td>123 000</td>
</tr>
</tbody>
</table>

Results from the application of NPVMining to the McLaughlin gold mine case study

The stochastic cut-off grade model (NPVMining) is used to calculate the optimum cut-off grade that maximizes NPV in the shortest mine life possible using the price variations in Table VI and the grade-tonnage curves in Table IV. A summary of the best-case scenario showing the calculated NPV for six equally probable grade-tonnage curves and a range of metal prices is given in Table VII. The resultant cut-off grade policies from the model shows a significant difference between the minimum and maximum NPV generated through the equally probable grade-tonnage curves. The resultant cut-off grade policies generate the optimal NPV in approximately 9 years for the six grade-tonnage curves. The application (NPVMining) also generates possible outcomes in relation to the change in gold price. The six possible outcomes for each price category generate different outcome as shown in Table VII. The resultant cut-off grade policies are generated by varying the grade–tonnage curves for all mineable ore in every grade interval and the metal prices against time. The solution was generated in 14.45 seconds after running the NPVMining code on Microsoft Visual Studio 2017.

Table VI

<table>
<thead>
<tr>
<th>Price category name</th>
<th>Value ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year 1 Range 1</td>
<td>1250</td>
</tr>
<tr>
<td>Year 2 Range 2</td>
<td>1270</td>
</tr>
<tr>
<td>Year 3 Range 1</td>
<td>1290</td>
</tr>
<tr>
<td>Year 4 Range 2</td>
<td>1310</td>
</tr>
<tr>
<td>Year 5 Range 1</td>
<td>1330</td>
</tr>
<tr>
<td>Year 6 Range 2</td>
<td>1356</td>
</tr>
</tbody>
</table>

The relationship between NPV and the metal price was modelled using a linear regression approach (Figures 9 and 10). The linear regression model shown in Figure 9 is applied to identify the relationship between NPV and the response variables (metal price) when all the other variables in the model are held fixed. The correlation coefficient in the relationship between metal price and NPV is about 0.97, indicating that metal price variation has a high significance for the NPV.
A stochastic cut-off grade optimization model to incorporate uncertainty for improved project value

The relationship between NPV and the other parameter (grade-tonnage distribution) when all the other variables in the model are held fixed is shown in Figure 10. The correlation coefficient in the relationship between grade-tonnage distribution and NPV is about 0.13, indicating that grade-tonnage distribution has a low impact on NPV.

### Comparison of NPVMining to other cut-off grade approaches

In deterministic cut-off grade optimization approaches applied in the mining industry, the outcomes are determined through known parameters without any room for random variation. This limits the flexibility of a mine plan in cases where commodity price and operational costs fluctuate through the LOM. However, the self-adaptation in stochastic algorithms contributes greatly to their ability to address successfully most complex real-world problems. This is due to their strategic outlook on mining problems that allows them to be easily applied to complex mining situations.

Table VIII summarizes the results of a comparison conducted between NPVMining and the following cut-off grade approaches:

(i) Break-even cut-off grade model (Githiria and Musingwini, 2018)
(ii) Cut-off Grade Optimiser (Githiria and Musingwini, 2018)
(iii) OptiPit® (Dagdelen and Kawahata, 2008)
(iv) Maptek Evolution® (Myburgh, Deb, and Craig, 2014).

Table VIII shows that NPVMining generated the highest NPV. Compared to the other cut-off grade approaches, NPVMining produced results that were:

- 7% better than the Cut-off Grade Optimiser
- 35% better than OptiPit®
- 13% better than Maptek Evolution®

The superior NPV generated by NPVMining can be attributed to its incorporation of stochasticity of input parameters, which is not incorporated in the other models. This demonstrates that superior results are obtained by incorporating uncertainty into Lane’s cut-off grade theory.

### Conclusion

The stochastic NPVMining model was applied to a gold mine case study data-set to ascertain its benefits in an operational mine. NPVMining was used to generate six cut-off grade policies, indicating that a change in grade-tonnage distribution has an overall effect on NPV. A comparison between NPVMining and other cut-off grade optimization models demonstrated and validated the efficiency of the model. Using an Intel dual-core processor running at 3.00GHz and with 4.00GB RAM, the model generated results for each simulation within 5 seconds. The improvements in NPV generated by NPVMining ranged between 7% and 186%, demonstrating the value of using stochastic approaches to cut-off grade optimization. Ignoring the commodity price and geological uncertainties in daily mining operations may have very serious negative economic implications for a mining project.

### Acknowledgment

The work reported in this paper is part of a PhD research study in the School of Mining Engineering at the University of the Witwatersrand. The financial support obtained from the Julian Baring Scholarship Fund (JBSF) for the PhD study is greatly acknowledged.

![Figure 9 — Linear relationship between metal price and NPV (Githiria, 2018)](image)

![Figure 10 — Linear relationship between grade-tonnage distribution and NPV (Githiria, 2018)](image)

### Table VIII

Comparison of the mine life, cash flow, and NPV from different cut-off grade models (Githiria, 2018)

<table>
<thead>
<tr>
<th>LOM (years)</th>
<th>Break-even cut-off grade model</th>
<th>Cut-off Grade Optimiser</th>
<th>OptiPit®</th>
<th>Maptek Evolution®</th>
<th>NPVMining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profits/cash flow ($ million)</td>
<td>35</td>
<td>863</td>
<td>10</td>
<td>825.44</td>
<td>18</td>
</tr>
<tr>
<td>NPV($ million)</td>
<td>863</td>
<td>163.42</td>
<td>10</td>
<td>347.08</td>
<td>10</td>
</tr>
</tbody>
</table>
A stochastic cut-off grade optimization model to incorporate uncertainty for improved project value

References


Appendix 1: Code for NPVMining (Githiria, 2018)

Stripping ratio
double CNPVMiningDlg::determineStripRatio(double dCutOff_grade, int iPos)
{
    //determine breakeven cutoff grade policy
double dSRatio; int g = 0;
double dCurrentPrice = dValArray[iPos]; //getCurrentPrice();
POSITION pos = listGradeCat.GetHeadPosition();
dTotalWaste = 0; dTotalOre = 0;
while (pos)
{
    GradeCategory sGradeCat = listGradeCat.GetNext(pos);
    if (sGradeCat.dLowerLimit < dCutOff_grade)
    {
        dTotalWaste += sGradeCat.dQuantity;
        dTotalOre = dMinedOre;
        dSRatio = dTotalWaste / dTotalOre;
        dSRatio = round(dSRatio*100)/100;
        return dSRatio;
    }
}

Average grade
double CNPVMiningDlg::average_grade(double dCutOff_grade)
{
A stochastic cut-off grade optimization model to incorporate uncertainty for improved project value

```cpp
double dResult = 0.0;
POSITION pos = listGradeCat.GetHeadPosition();
while (pos)
{
    GradeCategory sGradeCat = listGradeCat.GetNext(pos);
    if (sGradeCat.dLowerLimit >= dCutoff_grade)
    {
        dResult += sGradeCat.dMidpoint * sGradeCat.dQuantity;
        dResult = round(dResult * 1000) / 1000;
    }
}
dResult /= dTotalOre;
dResult = round(dResult * 1000) / 1000;
return dResult;
```

```cpp
void CNPVMiningDlg::determineOptimumGrade()
{
    dTotalWaste = 0.0;
    POSITION pos = listGradeCat.GetHeadPosition();
    while (pos)
    {
        GradeCategory sGradeCat = listGradeCat.GetNext(pos);
        sGradeCat.dOreQty = dTotalQty - dTotalWaste;
        sGradeCat.dWasteQty = dTotalWaste;
        dTotalQty -= sGradeCat.dQuantity;
        dTotalWaste += sGradeCat.dQuantity;
        sGradeCat.dmc = sGradeCat.dOreQty / (sGradeCat.dOreQty + sGradeCat.dWasteQty);
        sGradeCat.dg = determineGSum(pos) / sGradeCat.dOreQty;
        sGradeCat.dmr = (sGradeCat.dOreQty * sGradeCat.dG) / (sGradeCat.dOreQty + sGradeCat.dWasteQty);
        sGradeCat.dcr = sGradeCat.dG;
    }
}
```

```cpp
double CNPVMiningDlg::limitingcutoff_grade(int iPos)
{
    double dResult = 0.0;
    double dPrice = 0.0;
    double dNPV = cObj->get_NPV();
    double dAFixedCost = cObj->get_annualFixedCost();
    double dMillCapacity = cObj->get_millingCapacity();
    double dMineCapacity = cObj->get_miningCapacity();
    double dRefCapacity = cObj->get_refiningCapacity();
    //using the first value in the array
    dPrice = dValArray[iPos];
    if (m_boolMillCap == true)
    {
        dResult = (cObj->get_millingCost() + ((dAFixedCost + (cObj->get_discountedRate() * dNPV)) / dMillCapacity)) / ((dPrice - cObj->get_refiningCost()) * cObj->get_recovery());
    }
    else if (m_boolRefCap == true)
    {
        dResult = round(dResult * 100) / 100;
        return dResult;
    }
}
```

```cpp
double CNPVMiningDlg::breakevencutoff_grade(int iPos)
{
    double dResult = 0.0;
    double dPrice = 0.0;
    //using the first value in the array
    dPrice = dValArray[iPos];
    dResult = cObj->get_millingCost() / ((dPrice - cObj->get_refiningCost()) * cObj->get_recovery());
    dResult = round(dResult * 100) / 100;
    return dResult;
}
```

```cpp
bool CNPVMiningDlg::checkGradeCatDuplicate(double lowerlimit, double upperlimit)
{
    bool bFound = false;
    for (int g = 0; g < m_listGradeCat.GetItemCount(); g++)
    {
        if (_wtof(m_listGradeCat.GetItemText(g, 1)) == lowerlimit || upperlimit == _wtof(m_listGradeCat.GetItemText(g, 2)))
        {
            bFound = true;
            break;
        }
    }
    return bFound;
}
```

```cpp
bool CNPVMiningDlg::checksubCatDuplicate(CString cat, CString subcat)
{
    bool bFound = false;
    for (int g = 0; g < m_listPriceCat.GetItemCount(); g++)
    {
        if (m_listPriceCat.GetItemText(g, 1) == cat && subcat == m_listPriceCat.GetItemText(g, 2))
        {
            bFound = true;
            break;
        }
    }
    return bFound;
}
```

```cpp
int CNPVMiningDlg::factorial(int n)
{
    return (n == 1 || n == 0) ? 1 : factorial(n - 1) * n;
}
```

```cpp
int CNPVMiningDlg::combinationcount(int n, int r)
{
    return (n == 1 || n == 0) ? 1 : factorial(n - 1) * n;
}
```

```cpp
bool CNPVMiningDlg::checksubCatDuplicate(CString cat, CString subcat)
{
    bool bFound = false;
    for (int g = 0; g < m_listPriceCat.GetItemCount(); g++)
    {
        if (m_listPriceCat.GetItemText(g, 1) == cat && subcat == m_listPriceCat.GetItemText(g, 2))
        {
            bFound = true;
            break;
        }
    }
    return bFound;
}
```

```cpp
double CNPVMiningDlg::breakevencutoff_grade(int iPos)
{
    double dResult = 0.0;
    double dPrice = 0.0;
    //using the first value in the array
    dPrice = dValArray[iPos];
    dResult = cObj->get_millingCost() / ((dPrice - cObj->get_refiningCost()) * cObj->get_recovery());
    dResult = round(dResult * 100) / 100;
    return dResult;
}
```
A stochastic cut-off grade optimization model to incorporate uncertainty for improved project value

```cpp
{ return (factorial(n) / (factorial(r)*factorial(n - 1))); }

int CNPVMiningDlg::getusedcategoriesList()
{
    int iTotal = 0, iLast = 0;
    int iCat = m_listPriceCat.GetItemCount();
    for (int v = 0; v < 30; v++)
        szUsedCatItems[v] = L"";
    for (int t = 0; t < iCat; t++)
    {
        CString szTempCat = m_listPriceCat.GetItemText(t, 1);
        for (int k = 0; k < 30; k++)
        {
            if (k == iLast)
            {
                if (szTempCat == szUsedCatItems[iLast - 1])
                    break;
                else {
                    szUsedCatItems[iTotal] = szTempCat;
                    iTotal += 1;
                    iLast += 1;
                    break;
                }
            }
            else if (k < iLast) continue;
        }
    }
    return iTotal;
}

int CNPVMiningDlg::combinationtotals()
{
    CString szArrayVals[100][2];
    CString szArrayCat[20];
    int iArrayCatSubItems[20] = { 0 };
    int iArrayCatSubItemsCmb[20] = { 0 };
    int iArrayCatSubItemsCmbTotals = 0;
    int iCurrVal = 0, iCurrPos = 0;
    //determine how many categories exist
    int iCat = m_cmbPriceCat.GetCount();
    for (int u = 0; u < iCat; u++)
    {
        m_cmbPriceCat.GetLBText(u, szArrayCat[u]);
    }
    //determine the number of combinations expected
    int iTotCat = combinationcount(iCat, 1);
    //loop through all the data and identify the subcategories in each category and store them in an array
    for (int t = 0; t < m_listPriceCat.GetItemCount(); t++)
    {
        szArrayVals[t][0] = m_listPriceCat.GetItemText(t, 1);
        szArrayVals[t][1] = m_listPriceCat.GetItemText(t, 2);
        for (int k = 0; k < iCat; k++)
        {
            if (szArrayVals[t][0] == szArrayCat[k])
                iArrayCatSubItems[k] += 1;
        }
    }
    //determine the number of combinations for each category
    for (int d = 0; d < iCat; d++)
    {
        if (iArrayCatSubItems[d] > 1)
        {
            iArrayCatSubItemsCmb[d] = combinationcount(iArrayCatSubItems[d], 1);
        }
        else if (iArrayCatSubItems[d] == 1)
        {
            iArrayCatSubItemsCmb[d] = 1;
        }
    }
    //get the summation of all the combinations
    for (int j = 0; j < iTotCat; j++)
    {
        for (int i = 0; i < iCatCmb; i++)
        {
            if (i <= iCurrPos) continue;
            else iArrayCatSubItemsCmbTotals += iCurrVal * iArrayCatSubItemsCmb[i];
        }
        iCurrPos += 1;
    }
    return iArrayCatSubItemsCmbTotals;
}

void CNPVMiningDlg::fillcombinationlist(COleSafeArray *m_combinationList)
{
    long index1[2];
    int iArrayCatLastSubItems[20] = { 0 };
    int iLastPosArray[20] = { 0 };
    //get a list of the category combinations
    int iTotCat = iUsedCatNum;
    for (int r = 0; r < 100; r++)
    {
        for (int e = 0; e < 2; e++)
        {
            szArrayVals[r][e] = L"";
        }
    }
    for (int b = 0; b < 20; b++) iLastPosArray[b] = -1;
    int iItemCount = m_listPriceCat.GetItemCount();
    for (int t1 = 0; t1 < iItemCount; t1++)
    {
        CString szCatItem = m_listPriceCat.GetItemText(t1, 1);
        szArrayVals[t1][0] = szCatItem;
        CString szSubCatItem = m_listPriceCat.GetItemText(t1, 2);
        szArrayVals[t1][1] = szSubCatItem;
        for (int k = 0; k < iCat; k++)
        {
            if (szArrayVals[t1][0] == szArrayCat[k])
                iArrayCatLastSubItems[k] = t1;
        }
    }
    int iCurrRow = 2, iCurrCol = 1;
    int i = 0;
    while (1)
    { if (iArrayCatSubItems[d] > 1)
    {
        iArrayCatSubItemsCmb[d] = combinationcount(iArrayCatSubItems[d], 1);
    }
        else if (iArrayCatSubItems[d] == 1)
        {
            iArrayCatSubItemsCmb[d] = 1;
        }
    } //get the summation of all the combinations
    for (int j = 0; j < iTotCat; j++)
    {
        for (int i = 0; i < iCatCmb; i++)
        {
            if (i <= iCurrPos) continue;
            else iArrayCatSubItemsCmbTotals += iCurrVal * iArrayCatSubItemsCmb[i];
        }
        iCurrPos += 1;
    }
    return iArrayCatSubItemsCmbTotals;
} //void CNPVMiningDlg::fillcombinationlist(COleSafeArray *m_combinationList)
```
A stochastic cut-off grade optimization model to incorporate uncertainty for improved project value

```c
for (int j = 0; j < iCat; j++)
{
    for (int t = 0; t < itemCount; t++)
    {
        if (szArrayVals[t][0] != szUsedCatItems[j])
            continue;
        if (j == 0) // has a number on its right
        {
            if (iLastPosArray[j] != iArrayCatLastSubItems[j]) {
                int iTemp = j;
                bool bTemp = true;
                loopj:
                    if (iLastPosArray[iTemp + 1] ==
                        iArrayCatLastSubItems[iTemp + 1])
                        if (iTemp + 1 == iCat - 1)
                            if (t <= iLastPosArray[j] && bTemp == true) {
                                continue;
                            } else {
                                iTemp += 1;
                                if (iLastPosArray[iTemp + 1] <
                                    iArrayCatLastSubItems[iTemp + 1]) bTemp = false;
                                goto loopj;
                            }
                    if (t < iLastPosArray[j]) continue;
                loop1:
                else {
                    if (iLastPosArray[iTemp + 1] ==
                        iArrayCatLastSubItems[iTemp + 1])
                        if (iTemp + 1 == iCat - 1)
                        {
                            if (t <= iLastPosArray[j] && bTemp == true)
                            {
                                // start from the first sub item
                                if (j > 0) t =
                                    iArrayCatLastSubItems[j - 1] + 1;
                                else return;
                            }
                        } else {
                            iTemp += 1;
                            if (iLastPosArray[iTemp + 1] <
                                iArrayCatLastSubItems[iTemp + 1]) bTemp = false;
                            goto loop1;
                        }
                    } else {
                        long numpos[2];
                        numpos[0] = index1[0];
                        numpos[1] = 0;
                        CString szNum; szNum.Format(L"%d",
                            numpos[0] - 1);
                        BSTR bstr = szNum.AllocSysString();
                        m_combinationList->PutElement(numpos, bstr);
                        SysFreeString(bstr);
                        BSTR bstr1 = szArrayVals[t][1].AllocSysString();
                        m_combinationList->PutElement(index1, bstr1);
                        SysFreeString(bstr1);
                        iLastPosArray[j] = t;
                        break;
                    }
                    ++i;
                }
            }
        }
    }
}
```

```c
double CNPVMiningDlg::dgetAnnualProfit(int iPos)
{
    double dResult = 0.0;
    double dPrice = 0.0;
    // using the first value in the array
    dPrice = dValArray[iPos];
    dResult = ((dPrice - cObj->get_refiningCost()) * cObj->
                get_refiningCapacity() - (cObj->get_millingCapacity() *
                cObj->get_millingCost()) - cObj->get_annualFixedCost());
    dResult = round(dResult * 100) / 100;
    return dResult;
}
```

```c
double CNPVMiningDlg::dgetNPV(double dMinelife, double dAnnualProfit)
{
    double dResult = 0.0;
    dResult = (dAnnualProfit * ((round(pow(1 + cObj->
        get_discountedRate(), dMinelife) * 1000) / 1000) - 1)) /
                ((round(pow(1 + cObj->get_discountedRate(), dMinelife) *
                    1000) / 1000) * cObj->get_discountedRate());
    dResult = round(dResult * 100) / 100;
    return dResult;
}
```