http://dx.doi.org/10.17159/2411-9717/2019/v119n3a1

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

by J. Githiria[1] and C. Musingwini[1]

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

1 *University of the Witwatersrand, South Africa.*
 ISSN 2225-6253. Paper received Aug. 2018.

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].

$$\text{Max NPV} = \sum_{i=1}^{N} \frac{Pw_i}{(1+d)^i} \qquad [1]$$

Table I

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

Study	Summary of improvement
Taylor (1972)	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.
Taylor (1985)	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.
Dagdelen (1992, 1993)	Dagdelen developed an analytical method for finding balancing cut-off grades which was more efficient than Lane's graphical.
Whittle and Wharton (1995)	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.
Asad (1997, 2005)	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.
King (2001)	King modified Lane's approach by incorporating variations in throughput for different ore types in order to cater for polymetallic deposits.
Dagdelen and Kawahata (2008)	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 OptiPit®.mining software package
Gholamnejad (2008, 2009)	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.
Osanloo, Rashidinejad, and Rezai (2008)	The study incorporated environmental requirements into Lane's cut-off grade optimization by incorporating waste and tailings disposal costs.
King (2011)	King modified the earlier (King, 2001) version by incorporating operating and administrative costs.
Githiria, Muriuki, and Musingwini (2016)	The study developed a computer-aided application based on Lane's algorithm and improved the efficiency of the cut-off grade calculation process.

Table II

Examples of studies addressing uncertainty in cut-off grade determination

Study	Summary of stochastic or DP approach
Dowd (1976)	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.
Krautkraemer (1988)	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.
Cetin and Dowd (2002, 2013, 2016)	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.
Asad (2007)	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.
Li, Yang, and Lu (2012)	The approach optimized cut-off grade using stochastic programming in open-pit mining but considered commodity price as the only dynamic variable.
Asad and Dimitrakopoulos (2013)	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.
Thompson and Barr (2014)	The approach optimized cut-off grade using stochastic programming in open-pit mining, but considered commodity price as the only dynamic variable.
Myburgh, Deb, and Craig (2014)	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 extraction sequence by managing two lower-level algorithms. The first one is LP-based and determines the optimal flow of material through multiple processing streams and manages stockpiles. The second is a local search technique for deriving the best production schedule.

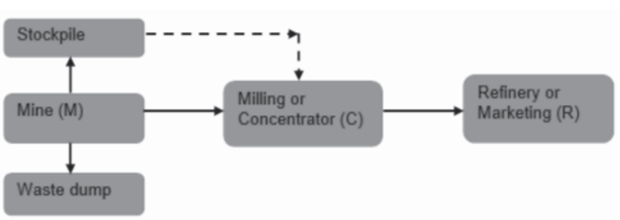

Figure 1 – Schematic layout of material flow in a mining complex (Githiria, 2018)

Table III

Notations used in Lane's algorithm (adapted from Githiria, 2018)

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

where d is the discount rate and Pw_i 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].

$$\begin{aligned} &Cash\ flow, P_{wi} = \left(\sum_{i=1}^{N} s_i - r_i\right) * \\ &Qr_{wi} - c_i * Qc_{wi} - m_i * Qm_{wi} - f_i \\ &\text{subject to: } (Qm_{wi} \leq M \text{ for } i = 1 \ldots N), \\ &(Qc_{wi} \leq C \text{ for } i = 1 \ldots N), \text{ and} \\ &(Qr_{wi} \leq R \text{ for } i = 1 \ldots N) \end{aligned} \quad [2]$$

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 (Qm_w), quantity processed (Qc_w), quantity refined (Qr_w), 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 2013 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 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 *NPVMining* 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 *NPVMining*, 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

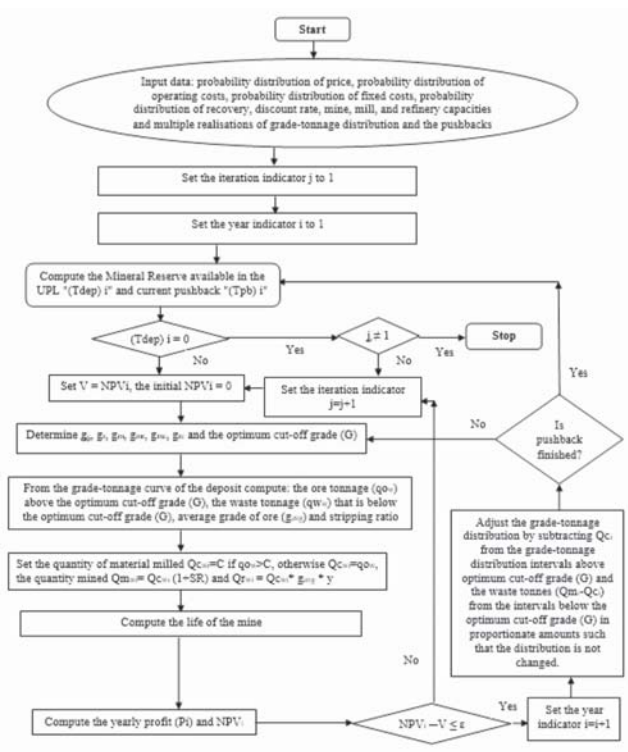

Figure 2—Flow diagram of the modified cut-off grade algorithm (Githiria, 2018)

(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 NPV_i is not known set the NPV_i to zero.
5. Determine the tons of ore (qo_W), tons of waste (qw_W), and average grades of the ore associated w the optimum cut-off grade (g_{avg}). Set: $Qc_{wi} = C$ if qo_W is greater than the milling capacity (C), otherwise $Qc_{wi} = qo_W$. Using Equations [3]–[7], calculate the quantity to be mined (Q_m) and refined (Q_r). Find the limiting capacity and the mine life (N) from the following:

$$Qr_{wi} = Qc_{wi} * g_{avg} * \sum y_i \qquad [3]$$

$$Qm_{wi} = Qc_{wi}(1 + SR) \qquad [4]$$

$$N = \frac{Q_{mwi}}{M} \qquad [5]$$

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

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

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

$$Cash\ Flow, P_{wi} = \left(\sum_{i=1}^{N} s_i - r_i\right) * Qr_{wi} - c_i * Qc_{wi} - m_i * Qm_{wi} - f_i \qquad [8]$$

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.

$$\text{Max } NPV = \sum_{i=1}^{N} \frac{Pw_i}{(1+d)^i} \qquad [9]$$

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 ($Qm_W - Qc_W$) 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.

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.

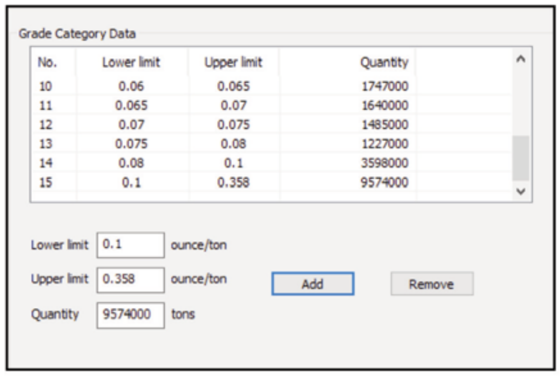

Figure 4—Grade category window (Githiria, 2018)

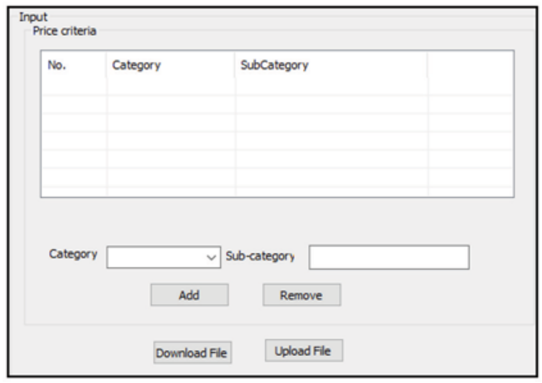

Figure 5—Price criteria window (Githiria, 2018)

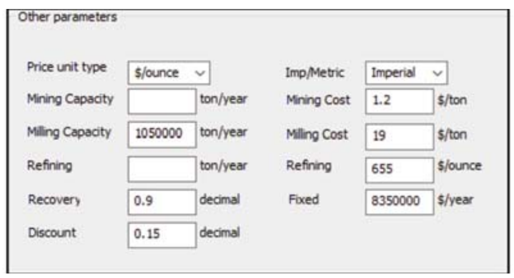

Figure 6—Other parameters (economic and operational parameters) window (Githiria, 2018)

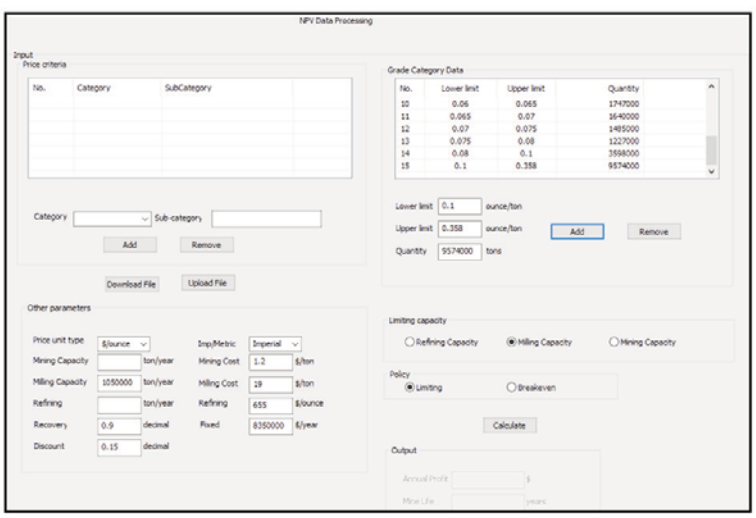

Figure 3—*NPVMining* user interface (Githiria, 2018)

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 an 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 (*Qm*), quantity of ore to be milled (*Qc*), quantity of product refined (*Qr*), 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)

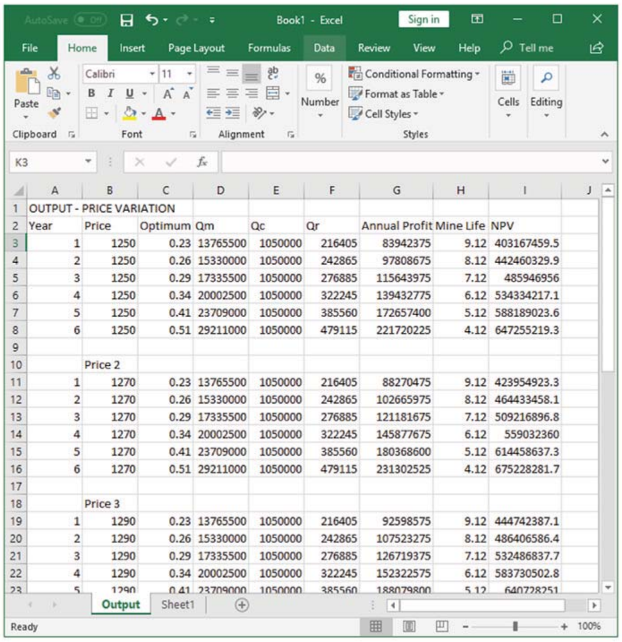

Figure 8—Output exported to a Microsoft Excel® spreadsheet (Githiria, 2018)

Table V

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

Notation	Explanation	Unit	Amount
CC	Capital cost	$ million	105 000 000
m	Mining cost	$/ton	1.20
c	Milling cost	$/ton	19.00
r	Refining/Selling cost	$/oz	655.00
f	Annual fixed costs	$/year	8,350,000
y	Recovery	Decimal	0.90
d	Discount rate	Decimal	0.15
M	Mining capacity	t/a	Unlimited
C	Milling capacity	t/a	1 050 000
R	Refining capacity	t/a	Unlimited

Table IV

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

Grade (oz/t)	Quantity of material in the simulated orebody realization (kt)					
	GTC1	GTC2	GTC3	GTC4	GTC5	GTC6
0.000-0.020	70 000	70 022	70 081	70 003	69 640	70 900
0.020-0.025	7 257	7 263	7 257	7 364	7 890	7 000
0.025-0.030	6 319	6 325	6 332	6 405	6 860	5 700
0.030-0.035	5 591	5 595	5 616	5 632	6 100	5 200
0.035-0.040	4 598	4 603	4 628	4 608	4 950	4 600
0.040-0.045	4 277	4 279	4 306	4 318	4 570	4 300
0.045-0.050	3 465	3 466	3 488	3 502	3 680	3 000
0.050-0.055	2 438	2 439	2 468	2 438	2 630	2 200
0.055-0.060	2 307	2 309	2 335	2 318	2 520	2 100
0.060-0.065	1 747	1 748	1 754	1 740	1 900	1 500
0.065-0.070	1 640	1 643	1 647	1 637	1 770	1 400
0.070-0.075	1 485	1 488	1 494	1 481	1 560	1 200
0.075-0.080	1 227	1 229	1 237	1 246	1 270	800
0.080-0.100	3 598	3 598	3 625	3 636	3 810	3 300
0.100-0.358	9 574	9 578	9 579	9 641	10 330	9 800
Total	125 523	125 585	125 847	125 969	129 480	123 000

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 and the waste material is sent to the waste dump. The operation is assumed to have an unrestricted potential to mine and refine/market the annual gold production, while the processing capacity is set to be at 1.05 Mt/a of ore (Table V). Gold price uncertainty applied in this study was assumed to have a range (2% per period) that is increasing yearly, as shown in Table VI, to be within the generally accepted long-term gold price of around US$1300 per ounce.

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 generated as shown 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.

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 VI

Price variations used in the model (Githiria, 2018)

Price category name	Value ($/oz)
Year 1Range1	1250
Year 2 Range2	1270
Year 3 Range1	1290
Year 4 Range2	1310
Year 5 Range1	1330
Year 6 Range2	1356

Table VII

Net present values generated using *NPVMining* when varying price and grade-tonnage distribution (best–case scenario) (Githiria, 2018)

NPV ($)	Price (oz/ton)	Grade-tonnage (kt)
467 172 060.50	1250	125 523
467 172 060.49	1250	125 585
466 990 510.59	1250	125 847
468 922 686.82	1250	125 969
488 180 335.90	1250	129 480
475 864 529.70	1250	123 000
487 959 524.30	1270	125 523
487 959 524.30	1270	125 585
487 777 974.39	1270	125 847
489 777 241.30	1270	125 969
509 740 604.86	1270	129 480
496 886 282.14	1270	123 000
508 746 988.10	1290	125 523
508 746 988.10	1290	125 585
508 565 438.18	1290	125 847
510 631 795.78	1290	125 969
531 300 873.82	1290	129 480
517 908 034.58	1290	123 000
529 534 451.90	1310	125 523
529 534 451.88	1310	125 585
529 352 901.98	1310	125 847
531 486 350.25	1310	125 969
552 861 142.78	1310	129 480
538 929 787.03	1310	123 000
550 321 915.70	1330	125 523
550 321 915.68	1330	125 585
550 140 365.78	1330	125 847
552 340 904.73	1330	125 969
574 421 411.74	1330	129 480
559 951 539.47	1330	123 000
571 109 379.50	1350	125 523
571 109 379.47	1350	125 585
570 927 829.57	1350	125 847
573 195 459.21	1350	125 969
595 981 680.69	1350	129 480
580 973 291.91	1350	123 000

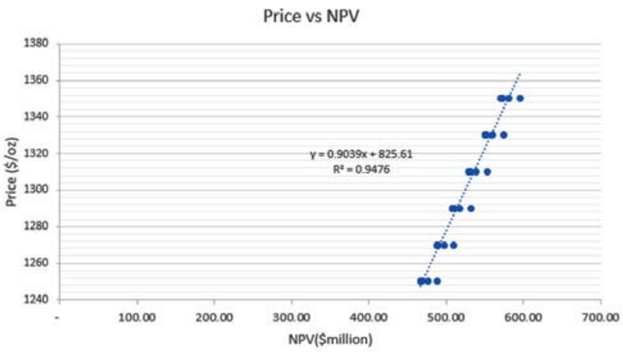

Figure 9—Linear relationship between metal price and NPV (Githiria, 2018)

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:

- 186% better than the break-even cut-off grade model
- 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.

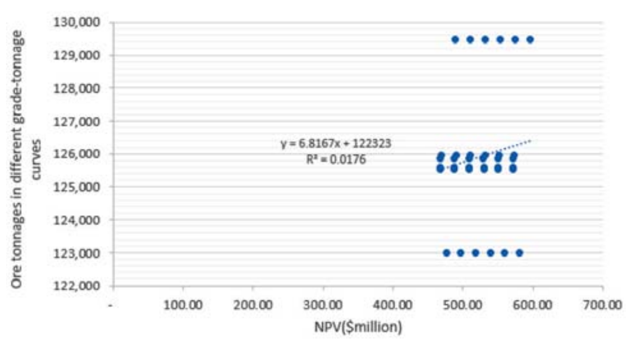

Figure 10—Linear relationship between grade-tonnage distribution and NPV (Githiria, 2018)

Table VIII

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

	Deterministic cut-off grade approaches			Stochastic cut-off grade approaches	
	Break-even cut-off grade model	Cut-off Grade Optimiser	OptiPit®	Maptek Evolution®	*NPVMining*
LOM (years)	35	10	18	10	9.12
Profits/cash flow ($ million)	863	825.44	885.60	760.10	887.09
NPV($ million)	163.42	435.52	347.08	413.84	467.17

References

ASAD, M.W.A. 1997. Multi mineral cut-off grade optimization with option to stockpile, MSc thesis. Colorado School of Mines, Golden, CO.

ASAD, M.W.A. 2002. Development of generalized cut-off grade optimisation algorithm for open pit mining operations. *Journal of Engineering and Applied Sciences*, vol. 21, no. 2. pp. 119–127.

ASAD, M.W.A. 2005. Cut-off grade optimisation algorithm with stockpiling option for open pit mining operations of two economic minerals. *International Journal of Surface Mining, Reclamation and Environment*, vol. 19, no. 3. pp. 176–187.

ASAD, M.W.A. 2007. Optimum cut-off grade policy for open pit mining operations through net present value algorithm considering metal price and cost escalation. *Engineering Computations*, vol. 24, no. 7. pp. 723–736.

ASAD, M.W.A. and DIMITRAKOPOULOS, R. 2013. A heuristic approach to stochastic cut-off grade optimisation for open pit mining complexes with multiple processing streams. *Resources Policy*, vol. 38. pp. 591–597.

CETIN, E. and DOWD, P.A. 2002. The use of genetic algorithms for multiple cut-off grade optimisation. *Proceedings of the 32nd International Symposium on Application of Computers and Operations Research in the Mineral Industry (APCOM)*. Society for Mining, Metallurgy & Exploration, Littleton, CO. pp. 769–779.

CETIN, E. and DOWD, P.A. 2013. Multi-mineral cut-off grade optimization by grid search. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 113, no.8. pp. 659–665.

CETIN, E. AND DOWD, P.A. 2016. Multiple cut-off grade optimization by genetic algorithms and comparison with grid search method and dynamic programming. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 116, no.7. pp. 681–688.

DAGDELEN, K. 1992. Cut-off grade optimisation. *Proceedings of the 23rd International Symposium on Application of Computers and Operations Research in the Minerals Industry (APCOM)*. Society for Mining, Metallurgy & Exploration, Littleton, CO. pp. 157–165.

DAGDELEN, K. 1993. An NPV optimisation algorithm for open pit mine design. *Proceedings of the 24th International Symposium on Application of Computers and Operations Research in the Mineral Industry*, Montreal, Quebec, Canada. Canadian Institute of Mining, Metallurgy and Petroleum, Montreal. pp. 257–263.

DAGDELEN, K. AND KAWAHATA, K. 2008. Value creation through strategic mine planning and cut-off grade optimization, *Mining Engineering*, vol. 60, no. 1. pp. 39–45.

DOWD, P.A. 1976. Application of dynamic and stochastic programming to optimise cut-off grades and production rates. *Transactions of the Institution of Mining and Metallurgy*, *Section A: Mining Technology*, vol. 85, no. 1. pp 22–31.

ESPINOZA, D., GOYCOOLEA, M., MORENO, E., and NEWMAN, A. 2012. MineLib: a library of open pit mining problems. *Annals of Operations Research*, vol. 206. pp. 93–114.

GHOLAMNEJAD, J. 2008. Determination of the optimum cut-off grade considering environmental cost. *Journal of International Environmental Application and Science*, vol. 3, no. 3. pp.186–194.

GHOLAMNEJAD, J. 2009. Incorporation of rehabilitation cost into the optimum cut-off grade determination. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 108, no. 2. pp. 89–94.

GITHIRIA, J.. 2018. A stochastic cut-off grade optimisation algorithm. PhD thesis, University of the Witwatersrand, South Africa.

GITHIRIA, J., MURIUKI, J., and MUSINGWINI, C. 2016. Development of a computer-aided application using Lane's algorithm to optimise cut-off grade. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 116, no. 11. pp. 1027–1035.

GITHIRIA, J. and MUSINGWINI, C. 2018. Comparison of cut-off grade models in mine planning for improved value creation based on NPV. *Proceedings of the 6th Regional Conference of the Society of Mining Professors (SOMP)*, Johannesburg, South Africa. Southern African Institute of Mining and Metallurgy, Johannesburg. pp. 347–362.

HENNING, U. 1963. Calculation of cut-off grade. *Canadian Mining Journal*, vol. 84, no. 3. pp. 54–57.

JOHNSON, T.B. 1969. Optimum open-pit mine production scheduling. *A Decade of Digital Computing in the Mineral Industry*. Weiss A. (ed.). AIME, New York, pp. 539–562.

KING, B. 2001. Optimal mine scheduling policies. PhD thesis, Royal School of Mines, Imperial College, London University, UK.

KING, B. 2011. Optimal mining practice in strategic planning. *Journal of Mining Science*, vol. 47, no. 2. pp. 247–253.

KRAUTKRAEMER, J.A. 1988. The cut-off grade and the theory of extraction. *Canadian Journal of Economics*, vol. 21, no. 1. pp. 146–160. http://doi:10.2307/135216

LANE, K.F. 1964. Choosing the optimum cut-off grade. *Colorado School of Mines Quarterly*, vol. 59, no. 4. pp. 811–829.

LANE, K.F. 1988. The Economic Definition of Ore: Cut-off Grade in Theory and Practice. Mining Journal Books, London.

LI, S., YANG, C., and LU, C. 2012. Cut-off grade optimization using stochastic programming in open-pit mining. *Proceedings of the Sixth International Conference on Internet Computing for Science and Engineering*, Henan, China. IEEE, New York. pp. 66–69. doi:10.1109/ICICSE.2012.24

MORTIMER, G. 1950. Grade control. *Transactions of the Institution of Mining and Metallurgy*, vol. 59. pp. 1–43.

MYBURGH, C.A., DEB, K., and CRAIG, S. 2014. Applying modern heuristics to maximising net present value through cut-off grade optimisation. *Proceedings of Orebody Modelling and Strategic Mine Planning Symposium*, Perth, WA. Australasian Institute of Mining and Metallurgy, Melbourne. pp. 155–164.

OSANLOO, M., RASHIDINEJAD, F., and REZAI, B. 2008. Incorporating environmental issues into optimum cut-off grades modeling at porphyry copper deposits. *Resources Policy*, vol. 33, no. 4. pp. 222–229.

SHAVA, P. and MUSINGWINI, C. 2018. A net smelter return model for a polymetallic deposit. *Proceedings of the 6th Regional Conference of the Society of Mining Professors: Overcoming Challenges in the Mining industry through Sustainable Mining Practices*, Johannesburg, South Africa, 12-13 March 2018. Southern African Institute of Mining and Metallurgy, Johannesburg. pp. 279–291.

TAYLOR, H.K. 1972. General background theory of cut-off grades. *Transactions of the Institution of Mining and Metallurgy, Section A: Mining Technology*, vol. 81. pp. A160–A179.

TAYLOR, H.K. 1985. Cut-off grades – some further reflections. *Transactions of the Institution of Mining and Metallurgy, Section A: Mining Technology*, vol. 96. pp. 204–216.

THOMPSON, M. and BARR, D. 2014. Cut-off grade: a real options analysis. *Resources Policy*, vol. 42. pp. 83–92.

WHITTLE, J. and WHARTON, C. 1995. Optimising cut-offs over time. *Proceedings of the 25th International Symposium on the Application of Computers and Mathematics in the Mineral Industries*, Brisbane, Australia. Australasian Institute of Mining and Metallurgy, Melbourne. pp. 261–265.

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 = 0.0;
    dTotalOre = 0.0;
    while (pos)
    {
        GradeCategory sGradeCat = listGradeCat.GetNext(pos);
        if (sGradeCat.dLowerLimit < dCutoff_grade) dTotalWaste
        += sGradeCat.dQuantity;
        else dTotalOre += sGradeCat.dQuantity;
    }
    dTotalOre -= dMinedOre;
    dSRatio = dTotalWaste / dTotalOre;
    dSRatio = round(dSRatio*100)/100;
    return dSRatio;
}
```

Average grade

```
double CNPVMiningDlg::average_grade(double dCutoff_grade)
{
```

```
    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;
}
```

Optimum cut-off grade

```
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;
    }
}
```

Limiting cut-off grade

```
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_boolMineCap == true) {
        dResult = (cObj->get_millingCapacity() + ((dAFixedCost +
        (cObj->get_discountedRate() * dNPV)) / dMineCapacity))
        / ((dPrice - cObj->get_refiningCost()) * cObj-
        >get_recovery());
    }
    else if (m_boolRefCap == true) {
        dResult = (cObj->get_refiningCost() + ((dAFixedCost +
        (cObj->get_discountedRate() * dNPV)) / dRefCapacity)) /
        ((dPrice - cObj->get_refiningCost()) * cObj-
        >get_recovery());
    }
    dResult = round(dResult * 100) / 100;
    return dResult;
}
```

Break-even cut-off grade

```
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;
}
```

Grade category

```
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;
}
```

Sub category

```
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;
}
```

Creating an array of price values

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

```
{
    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 iCatCmb = 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 < iCatCmb; j++)
    {
        for (int i = 0; i < iCatCmb; i++)
        {
            iCurrVal = iArrayCatSubItemsCmb[iCurrPos];
            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 iCat = 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] == szUsedCatItems[k])
            {
                iArrayCatLastSubItems[k] = t1;
            }
        }
    }
    int iCurrRow = 2, iCurrCol = 1;
    int i = 0;
    while (1)
```

```
    {
        for (int j = 0; j < iCat; j++)
        {
            for (int t = 0; t < iItemCount; 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;
                }
                else {
                        if (t < iArrayCatLastSubItems[j])
                        continue;
                        else {
                          int iTemp = j;
                          bool bTemp = true;
loopj1:
                          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 loopj1;
                            }
                        }
                    }
                        }
                    }
                    else //no number on its right
                    {
                        if (iLastPosArray[j] !=
                        iArrayCatLastSubItems[j]){
                            if (t <= iLastPosArray[j]) continue;
                        }
                    }
                    index1[0] = iCurrRow + i;
                    index1[1] = iCurrCol + j;
                    if (j == 0)
                    {
                        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++;
            }
        }
```

Annual profit

```
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_miningCapacity() *
    cObj->get_miningCost()) - cObj->get_annualFixedCost();
    dResult = round(dResult * 100) / 100;
    return dResult;
}
```

NPV

```
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;
} ◆
```