

https://dspace.org/

Development of a computer-aided application using Lane's algorithm to optimize cut-off grade.

Githiria, J.; Musingwini, C.; Muriuki, J.

2016-11

The Southern African Institute of Mining and Metallurgy.

https://hdl.handle.net/10539/25479

Githiria, J., Musingwini, C. and Muriuki, J. 2016. Development of a computer-aided application using Lane's algorithm to optimize cut-off grade. Journal of the Southern African Institute of Mining and Metallurgy 116(11), pp. 1027-1035. http://dx.doi.org/10.17159/2411-9717/2016/v116n11a4

Downloaded from a DSpace repository. https://dspace.org/

http://dx.doi.org/10.17159/2411-9717/2016/v116n11a4

Development of a computer-aided application using Lane's algorithm to optimize cut-off grade

by J. Githiria*, C. Musingwini*, and J. Muriuki†

Synopsis

The maximization of net present value (NPV) is a primary objective in open pit mine planning processes. In an attempt to meet this objective, cut-off grades are considered in all the stages of mining. There are three stages involved in resource extraction, namely mining, processing, and refining/marketing, which are all defined in Lane's approach. Using Lane's approach, the economics involved in each stage are identified independently and interact to provide an optimum cut-off grade.

A hypothetical block model is used to illustrate how a computer application (Cut-off Grade Optimiser) that was developed in this study is used in the optimization of cut-off grades using Lane's algorithm. This paper analyses the application of Lane's approach for a single element in cut-off grade optimization as applied in the calculation of the optimum cut-off grade through a computer-aided application.

Although Lane's approach is not complex it is not widely applied to maximize the NPV of mining operations as the iterative calculations can be lengthy. The computer application developed in this study shows how Lane's algorithm combined with a linear programming function can optimize cut-off grade with regard to the complex situations faced by mining operations. The application saves users from performing otherwise lengthy, iterative calculations.

Keywords

cut-off grade optimization, Lane's algorithm, computer-aided application, C++ programming, Cut-off Grade Optimiser.

Background and previous research

An open pit mining complex constitutes multiple pits, ore processing streams, and metal refining options. An open pit mine produces ore (valuable) and waste material. Ore is transported to an appropriate processing stream for producing concentrate. A refinery processes the concentrate to produce marketable metal product (Lane, 1964, 1988).

Cut-off grade optimization is based on Lane's theory, where a mining system consists of three major stages such as mining, milling, and refining. In this approach the cut-off grade selection is done between six cut-off grades that involve all stages of a mining operation. Selection of the optimum cut-off grade influences the net present value (NPV) by maximizing the net cash flows from the operation.

The cut-off grades include three limiting cut-off grades, which are applied where one stage of the mining system is limiting the operation. These cut-off grades depend directly on price and costs, but indirectly on the grade distribution of the deposit. The other three are balancing cut-off grades, which balance the capacities of each pair of the stages. They depend on the grade distribution of the deposit only. The economic parameters do not affect the change in the balancing cut-off grade. Using these six cut-off grades the optimum cut-off grade is selected using rules that are discussed later in this paper. The optimum cut-off grade is then used in the cut-off grade calculations to obtain the yearly profits and net present value.

The optimal cut-off grade approach (Lane, 1964, 1988) has been modified in several studies by incorporating different mine scenarios. Dagdelen (1992) and Asad (2002) discussed the concept of optimum cut-off grade policy as outlined by Lane and formulated an algorithm used to maximize the NPV of a project. Dagdelen (1993) introduced an analytical method to find the balancing cut-off grades instead of using graphical methods by modifying Lane's approach, which simplified the problem. King (2001) modified the approach by incorporating variations in ore type throughput for multi-element mineralization. Cetin and Dowd (2002) applied genetic algorithms to optimize cut-off grade for a multi-mineral mine. Asad (2005) developed a cut-off grade optimization algorithm with a stockpiling option for open pit mining operations exploiting two economic minerals. Asad (2007) optimized cut-off grade for an open pit mining operation using an NPV-based

* *School of Mining Engineering, University of the Witwatersrand, Johannesburg, South Africa.*

† *Department of Mining and Mineral Processing Engineering, Jomo Kenyatta University of Agriculture and Technology, Kenya.*

© *The Southern African Institute of Mining and Metallurgy, 2016.* *ISSN 2225-6253. Paper received Apr. 2016; revised paper received Aug. 2016.*

algorithm that considered metal price and cost escalation. Osanloo, Rashidinejad, and Rezai (2008) incorporated environmental issues into cut-off grade optimization, while King (2011) presented a variation of cut-off grade policies considering operating and administrative costs. Asad and Dimitrakopoulos (2013) developed a heuristic approach that determines the optimal cut-off grade policy based on a stochastic framework that accounts for uncertainty in supply of ore to multiple ore processing streams.

Model definition

Most mining operations involve three common stages, namely mining, concentrating, and refining, each stage with its own associated costs and a limiting capacity as shown in Figure 1. The basic equations used in cut-off grade optimization can be developed using the definitions in Table I.

The profit, P from the next Q_m of material mined can be calculated as shown in Table I. The profit expression is formulated by considering the next unit of material to be mined, which is Q_m. Therefore, the cost of mining the material will be mQ_m. Some of the mined material will be sent to the concentrator, depending on the capacity of the concentrator which is denoted as Q_c. The cost incurred in processing this material is cQ_c. The refining costs are incurred on the final product, Q_r, thus having a cost of rQ_r. Total fixed costs, fT, are incurred over a time period T.

Steps of the algorithm for determining optimum cut-off grades

Table II shows the notations of parameters in Lane's algorithm. These are the input and output parameters for the computer application.

The following steps, as outlined by Lane (1964) and modified by Dagdelen (1993), are the general steps used in determining Lane's cut-off grade policy for a single-mineral deposit:

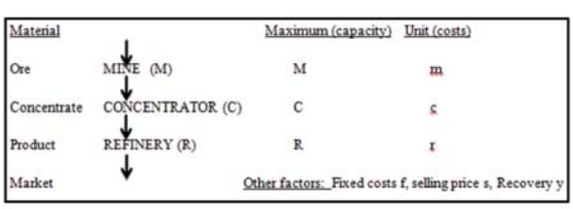


Figure 1—The model described by Lane's algorithm (Lane, 1964)

Table I

Basic equations

Definition	Equations
The total costs T_C, are	$Tc = mQ_m + cQ_c + rQ_r + fT$
Since the revenue, R is	$R = sQ_r$
The profit, P is given by	$P = R\text{-}T_c = sQ_r - (mQ_m + cQ_c + rQ_r + fT)$
Combining terms yields: (basic profit expression)	$P = (s\text{-}r)\ Q_r - cQ_c - mQ_m - fT$

i. Read the input files:
   a. Grade-tonnage distribution in each pushback for the whole deposit. [{$g(1), g(2)$}, {$g(2)$, g(3)}, ---------- , {$g(K\text{-}1), g(K)$}]
   b. Economic parameters to be used in the cut-off grade policy: mining capacity (M), milling capacity (C), refining capacity (R), selling price (s), refining cost (r), recovery (y), milling cost (c), mining cost (m), annual fixed costs (f), discount rate (d).
ii. Set the iteration indicator j to 1.
iii. Set the year indicator i to 1.
iv. Determine $g_c, g_r, g_m, g_{mr}, g_{mc}, g_{rci}$, and the optimum cut-off grade, G, using Equations [1], [2], [3], [4], [5], and [6], respectively. The initial V value NPV_i is set to zero.

$$g_c = \frac{c_i + \frac{f_i + Vd}{C}}{(s_i - r_i)y} \quad [1]$$

$$g_m = \frac{c_i}{(s_i - r_i)y} \quad [2]$$

$$g_r = \frac{c_i}{(s_i - r_i - \frac{f + Vd}{R})y} \quad [3]$$

$$g_{mc} = \left(\frac{\frac{C}{M} - mc(k^*)}{\frac{mc(k^*+1) - mc(k^*)}{g(k^*+1) - g(k^*)}} \right) + g(k^*) \quad [4]$$

$$g_{mr} = \left(\frac{\frac{R}{M} - mr(k^*)}{\frac{mr(k^*+1) - mr(k^*)}{g(k^*+1) - g(k^*)}} \right) + g(k^*) \quad [5]$$

Table II

Notations of Lane's algorithm

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

$$g_{rc} = \left(\frac{\frac{R}{C} - rc(k^*)}{\frac{rc(k^*+1) - rc(k^*)}{g(k^*+1) - g(k^*)}} \right) + g(k^*) \quad [6]$$

The optimum cut-off grade, G, is given by:

$$G_{mc} = \begin{cases} g_m & \text{if } g_{mc} \le g_m \\ g_c & \text{if } g_{mc} \ge g_c \\ g_{mc} & \text{otherwise} \end{cases} \quad [7]$$

$$G_{rc} = \begin{cases} g_r & \text{if } g_{rc} \le g_r \\ g_c & \text{if } g_{rc} \ge g_c \\ g_{rc} & \text{otherwise} \end{cases} \quad [8]$$

$$G_{mr} = \begin{cases} g_m & \text{if } g_{mr} \le g_m \\ g_r & \text{if } g_{mr} \ge g_r \\ g_{mr} & \text{otherwise} \end{cases} \quad [9]$$

$$\mathrm{G} = \text{middle value } (G_{rc},\ G_{mr},\ G_{mc}) \quad [10]$$

v. From the grade-tonnage curve of the deposit compute:
   a. The ore tonnage (*To*) above the optimum cut-off grade (*G*) obtained in step (iv)

$$To(G) = \sum_{k=k^*}^{k} T_k \quad [11]$$

   b. The waste tonnage (*Tw*) that is below the optimum cut-off grade (*G*) is calculated as:

$$Tw(G) = \sum_{k=1}^{k^*} T_k \quad [12]$$

   c. Average grade of ore, represented as g_{avg} is calculated as:

$$\mathrm{g}_{\mathrm{avg}}(k^*) = \frac{\sum_{k=k^*}^{K} \left(\mathrm{T}_{\mathrm{k}} \left[\frac{g(k) + g(k+1)}{2} \right] \right)}{\sum_{k=k^*}^{K} \mathrm{T}_{\mathrm{k}}} \quad [13]$$

   d. The stripping ratio *SR* is given by:

$$SR = \frac{Tw}{To} \quad [14]$$

vi. Set:
   a. $Qci = C$, if *To* is greater than the milling capacity (*C*), otherwise, $Qci = To$ (G)
   b. The *Qmi* quantity mined $Qmi = Qci\ (1+SR)$and $Qri = Qci \times g_{avg} \times Y$

vii. If the iteration indicator *j* is equal to 1, compute:
   a. The life of deposit *N* based on the limiting capacity among mine, mill, and refinery.
   b. The annual profit for the life of mine is given by:

$$P = \left\{ [(s-r)g_{avg}y] - c \right\} \mathrm{Qc}_{\mathrm{i}} - mQm_i - fT \quad [15]$$

where:
$T = 1$ if $N \ge 1$
$T = N$ if $N < 1$

viii. Adjust the grade-tonnage distribution by subtracting the ore tons Q_c from the grade distribution intervals above optimum cut-off grade (*G*) and the waste tonnes (Q_m-Q_c) from the intervals below the optimum cut-off grade (*G*) in proportionate amount such that the distribution is not changed.

ix. Check if Q_c is less than the milling capacity *C*, then set mine life $N = i$ and go to step (x); otherwise set the year indicator $i = i + 1$ and go to step (iv).

x. Compute the accumulated future NPVs based on the profits *P* calculated in step (vii) for each year from *i* to *N* by using the following formula for discounting the profits at given interest rate *d*:

$$NPV = \sum_{i=1}^{n} \frac{P_i}{(1+d)^i} \quad [16]$$

For each year $i = 1$, *N* is the total mine life in years.

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

xii. If it is the first iteration then, knowing the profits obtained in each year, find the NPV year-by-year by discounting back those profits

xiii. Use the NPVs obtained in step (xi) as initial NPVs for each corresponding year for the second iteration

xiv. Repeat the computation from step (iv) until the value *V* converges

xv. If the computed NPV converges, the application stops.

Using the optimum cut-off grades obtained in the algorithm, a yearly production schedule that shows the cut-off grade, quantity mined (Q_m), quantity processed (Qc), quantity refined (Q_r), profit, and NPV is calculated.

Development of the computer application for cut-off grade optimization

The application for cut-off grade optimization was developed using the C++ programming language in Visual Studio 2008 and is referred to as 'Cut-off Grade Optimiser'. This application is an executable file on computers that run applications with .exe extension. To run, the computer must have Visual Studio 2008 or a higher version or VC++ runtime libraries installed. Cut-off Grade Optimiser was developed using C++/CLI.

Components of the cut-off grade optimization application

The application comprises of a header, resource, and source files in the project folder. The header files are also known as 'include files' and these have a .h extension. Header files hold declarations for other files to use, in this case to hold the grade-tonnage distribution and economic and operational parameters. Header files comprise Cmining.h, Form.h, resource.h, and stdafx.h.

Source files contain a collection of statements or declarations in C++/CLI and are used to specify the actions to be performed by the computer. Source files have a .cpp

extension and each file is compiled at a time. The source files comprise of AssemblyInfo.cpp, Cmining.cpp, Cut-off Grade Optimiser.cpp, and stdafx.cpp. These are shown in Figure 2.

Resource files are interface elements that provide information to the user. Bitmaps, icons, toolbars, and cursors are all resources. Some resources can be manipulated to perform an action such as selecting from a menu or entering data in a dialog box. The resource files comprise app.ico and app.rc.

The cut-off grade algorithm is based on Lane's approach as described previously and shown in Figure 3 in Appendix 1.

Input

The input data for the application consists of the grade-tonnage distribution and economic and operational parameters; these are given in the form of the data in Tables III and IV, respectively. The grade-tonnage distribution is fed into the grade-tonnage input window while the economic and operational parameters are fed into the economic and operational parameters input window.

The grade-tonnage distribution input window bears three columns, namely the lower grade limit, upper grade limit, and quantity per increment. The economic and operational parameters input window queries the capacities associated with mine, mill, and refinery, metal price, refinery or marketing cost, milling cost, mining cost, recovery, discount rate, and the fixed cost.

Output

The result (production schedule) is tabulated in the output window shown in Figure 4 and automatically displayed in Microsoft Excel where it can be saved for ease of use. The output is displayed on the output window in Figure 5, which consists of eight columns. The output is the result of the cut-off grade policy calculations after the user clicks the 'calculate' button. The year is shown in the first column, optimum cut-off grade in the second column, quantities to be mined (Q_m) in the third column, quantities to be milled (Q_c) in the fourth column, quantities to be refined (Q_r) in the fifth column, the profit in sixth column, NPV in the seventh column, and number of iterations in each period in the eighth column. This output is automatically exported to Microsoft Excel and presented in the form of a Microsoft Excel spreadsheet for ease of use. The application is then closed using the 'close' command as shown in Figure 6 (Appendix 2).

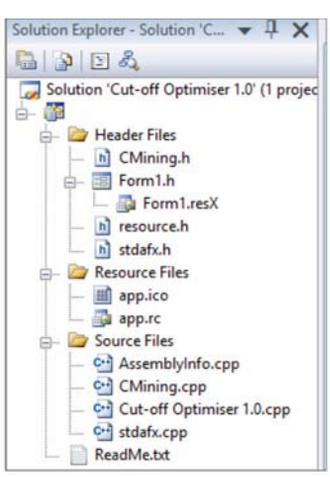


Figure 2—Component files of Cut-off Grade Optimiser on Solution Explorer in VC++

Results and discussion

Testing of the application

Cut-off Grade Optimiser was tested using various sets of data to authenticate its functionality. The data in Tables III and IV were used as the input values. The Microsoft Excel spreadsheet displays the inputs (grade-tonnage distribution and economic and operational parameters) and output (production schedule). The input parameters are displayed to help the user confirm the data inputs and detect any discrepancies in the results.

Table III provides the grade-tonnage distribution of the material within the optimal pit of this deposit, while Table IV gives the economic and operational parameters.

Table III

Grade-tonnage distribution data from the operational mine

Lower grade category (g/t)	Upper grade category (g/t)	Ore tons	Waste tons
0.00000	1.06250	1 819 351	76 105 532
1.06250	2.12500	1 788 756	77 924 883
2.12500	3.18750	909 145	79 713 639
3.18750	4.25000	316 053	80 622 784
4.25000	5.31250	242 488	80 938 837
5.31250	6.37500	171 553	81 181 325
6.37500	7.43750	190 734	81 352 878
7.43750	8.50000	135 038	81 543 612
8.50000	9.56250	113 672	81 678 650
9.56250	10.6250	63 922	81 792 322
10.6250	11.6875	49 007	81 856 244
11.6875	12.7500	54 133	81 905 251
12.7500	13.8125	23 927	81 959 384
13.8125	14.8750	10 525	81 983 311
14.8750	15.9395	6 243	81 993 836
15.9395	17.0000	4 812	82 000 079
17.0000	18.0000	116	82 004 891

Table IV

Economic parameters and operational capacities from the operational mine

Parameters	Symbol	Values
Price	*S*	$38.58 per gram
Sales	*R*	$0.16 per gram
Processing cost	*C*	$29.43 per ton
Mining cost	*M*	$1.20 per ton
Fixed cost	*F*	10 950 000 $ per year
Mining capacity	*M*	29 000 000 t
Milling capacity	*C*	1 050 000 t
Refining capacity	*R*	5 400 000 g
Capital costs	*CC*	$ 90 000 000
Processing recovery	*Y*	90%
Discount rate	D	10%
Mining recovery	1-*k*	95%
Mining dilution	*K*	5%
Time costs	*Tc*	$10 per ton

On running the data using Cut-off Grade Optimiser the yearly cut-off grades are as indicated in Table V. The cut-off grades change along the mine life in that average ore grades are higher in the earlier years, and lower towards the end of the mine life.

The cut-off grades are dynamic, being higher in earlier years and reducing gradually over the life of the mine.

The working of Cut-off Grade Optimiser is as follows:

i. The application is run and the display screen is as shown in Figure 4
ii. Input the grade-tonnage distribution (Table III) in the grade-tonnage distribution input window
iii. Input the economic and operational parameters (Table IV) in the economic and operational parameters input window
iv. Press the calculate button
v. The application runs the calculations in the background and displays the output
vi. The application prompts MS Excel to launch
vii. The output is displayed in MS Excel
viii. If results are satisfactory, then the user can close the application.

The user should confirm that the input values processed by the application are correct; if they are not, the values should be entered again and re-calculated. If the values are satisfactory the user can save or print the results in the spreadsheet. The results obtained are shown in Table V.

Conclusions and recommendations

Cut-off Grade Optimiser is an executable computer application based on Lane's approach, developed using the C++/CLI programming language in VC++ IDE. The application is user-friendly and easily obtainable and can be used to determine the optimum mining cut-off grade and the production schedule over the life of a mine. It saves users from performing otherwise lengthy, iterative calculations.

The main features of Cut-off Grade Optimiser are as follows.

- Cut-off Grade Optimiser is an executable computer application written in C++/CLI programming language in Visual Studio 2008 (VC++) Integrated Development Environment
- Cut-off Grade Optimiser is based on Lane's cut-off grade model (Lane, 1964), which considers the time value of money and maximizes the NPV of a mining project based on the grade-tonnage distribution of the deposit and the economic and operational parameters
- The input parameters are lower grade limits, upper grade limits and quantity in each increment, prevailing economic conditions, as well as the operational parameters of the mine project
- The outputs from the iterative calculations based on Lane's model are the production schedule (yearly quantities of material mined, processed, and refined), yearly profits and NPV, as well as the number of iterations per year
- To run on a computer the user needs to have installed VC++ 2008 or a higher version, or have VC++ runtime libraries installed.

Cut-off Grade Optimiser uses Lane's algorithm in an iterative approach, which involves factors such as the grade-tonnage distribution, prices, costs, recoveries, and capacities. In comparison to mining software such as Whittle 4X there is need for further modification to Cut-off Grade Optimiser. Whittle 4X involves more factors in its calculation and uses a proprietary search algorithm combined with powerful linear programming. Some of these factors are multiple mines, multiple mineral elements, multiple processing paths, alternative products, blending, changing prices, costs, recoveries, and capacities.

Cut-off Grade Optimiser can be adopted for use in determining the optimum cut-off grade for single-mineral deposit in open pit mines where complex mining scenarios such as blending, mine rehabilitation, are multiple economic minerals and not being considered.

Further modifications to Cut-off Grade Optimiser are essential so as to incorporate factors such as mining dilution, stockpiling or blending, multi-mineral orebodies, and mine rehabilitation, to mention but a few, so as to further optimize the NPV values.

Acknowledgement

The work reported in this paper is part of a PhD research study at the University of the Witwatersrand.

References

ASAD, M.W.A. 1997. Multi mineral cut-off grade optimisation 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.

Table V

Results of cut-off grade policy as reported from Cut-off Grade Optimiser

Year	Cut-off grade (g/t)	Q_m (t)	Q_c (t)	Q_r (g)	NPV ($ million)
1	1.54	26 281 763	1 050 000	3 363 255	138 832 595
2	1.31	22 180 888	1 050 000	2 362 304	55 434 814
3	0.9	14 674 405	1 050 000	1 821 960	18 290 682
4	0.85	16 013 837	1 050 000	1 738 800	9 885 399
5	0.85	1 817 411	242 281	572 390	9 230 324
		Total NPV			231 673 816

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.

BASCETIN, A. and NIETO, A. 2007. Determination of optimal cut-off grade policy to optimise NPV using a new approach with optimisation factor. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 107, no. 2. pp. 87–94.

CETIN, E. and DOWD, P.A. 2002. The use of genetic algorithms for multiple cut-off grade optimisation. *Proceedings of the 30th International Symposium on Application of Computers and Operations Research in the Mineral Industry*. Bandopadhyay, S. (ed.). Society for Mining, Metallurgy and Exploration, Littleton, CO. pp. 769–779.

DAGDELEN, K. 1992. Cut-off grade optimisation. *Proceedings of the 23rd International Symposium on Application of Computers and Operations Research in the Minerals Industry*, Tuczon, Arizona. Kim, Y.C.(ed.). Society for Mining, Metallurgy and 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. CIM, Montreal. pp. 257–263.

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

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

LANE, K.F. 1964. Choosing the optimum cut-off grade. *Colorado School of Mines Quarterly*, vol. 59. pp. 485–492.

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

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.

Appendix 1

Lane's cut-off grade policy flow diagram

Cut-off grade optimization using Lane's approach works in three major segments, namely the cut-off grade optimization segment, net present value (NPV) convergence segment, and the grade-tonnage curve adjustment segment. The quantity of material mined (Q_m), quantity of material processed (Q_c), quantity of ore processed (Q_r), profit, and NPV are all functions of the optimum cut-off grade. Figure 3 shows the flow diagram of Lane's algorithm, illustrating the three segments and the steps involved in each.

Appendix 2

Graphical user interface of the application

A program interface takes advantage of the computer's graphics capabilities to make the program easier to use. Graphical user interfaces (GUIs) free the user from learning complex command languages. In Cut-off Grade Optimiser, the main features in the GUI are; (i) an I-beam pointer shaped like a capital 'I' which appears where the input values are to be entered, and (ii) command buttons which execute the 'calculate' or 'close' actions. The display screen of the application is shown in Figure 4. The output is shown in Figure 5, and the 'close' command in Figure 6.

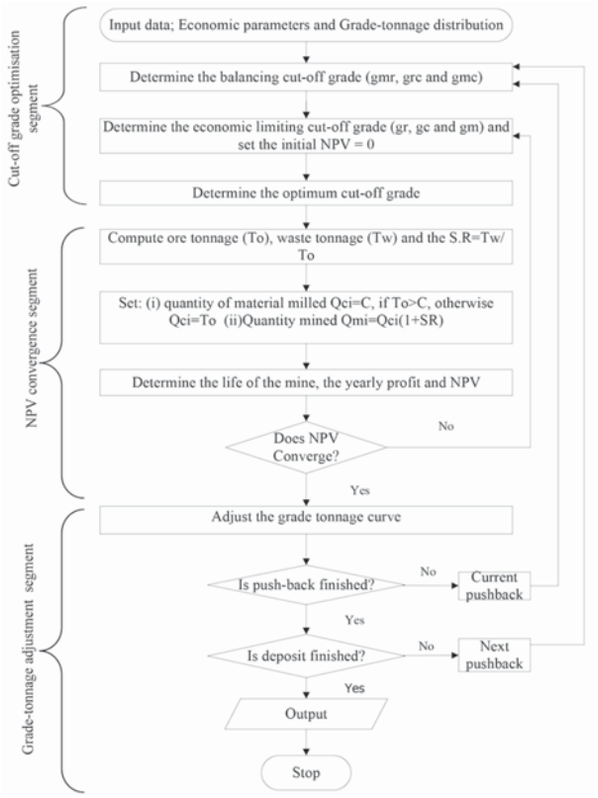


Figure 3—Flow diagram of Lane's approach (Asad, 1997; Bascetin and Nieto, 2007)

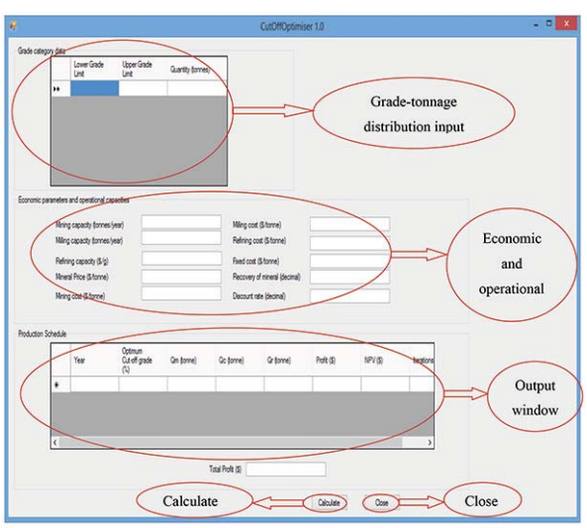


Figure 4—Graphical user interface of Cut-off Grade Optimiser

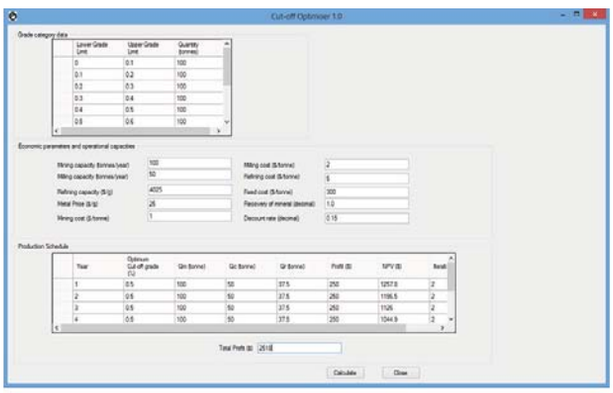


Figure 5 – Output showing the production schedule and total profit

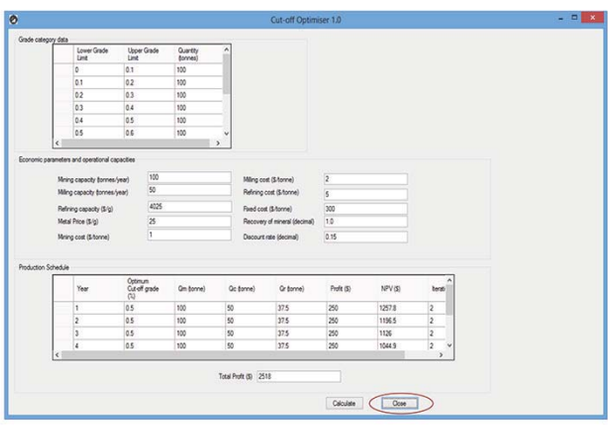


Figure 6—Close command

Appendix 3

Cmining.h

```
public class CMining
{
public:
    CMining();
    double get_miningCapacity();
    void set_miningCapacity(double value);
    double get_millingCapacity();
    void set_millingCapacity(double value);
    double get_refiningCapacity();
    void set_refiningCapacity(double value);
    double get_sellingPrice();
    void set_sellingPrice(double value);
    double get_refiningCost();
    void set_refiningCost(double value);
    double get_recovery();
    void set_recovery(double value);
    double get_millingCost();
    void set_millingCost(double value);
    double get_miningCost();
    void set_miningCost(double value);
    double get_annualFixedCost();
    void set_annualFixedCost(double value);
    double get_discountedRate();
    void set_discountedRate(double value);
    double get_NPV();
    void set_NPV(double value);
    double get_Gc();
    double get_Gmc();
    double get_Gr();
    double get_Gmr();
    double get_Gm();
    double get_Grc();
    void calc_Gc();
    void calc_Gmc(double mc_k, double mc_k1, double g_k,
    double g_k1);
    void calc_Gr();
    void calc_Gmr(double mr_k, double mr_k1, double g_k,
    double g_k1);
    void calc_Gm();
    void calc_Grc(double g_k, double g_k1, double rc_k,
    double rc_k1);
    double get_Gopt();
    double get_limiting_N(double N_r, double N_c, double
    N_m);
private:
    double miningCapacity;
    double millingCapacity;
    double refiningCapacity;
    double sellingPrice;
    double refiningCost;
    double recovery;
    double millingCost;
    double miningCost;
    double annualFixedCost;
    double discountedRate;
    double g_c;
    double g_mc;
    double g_r;
    double g_mr;
    double g_m;
    double g_rc;
    int i;
    double npValue;
    double Gmc;
    double Gmr;
    double Grc;
    double G_opt;
};
```

Appendix 4

Cmining.cpp

```
#include "stdafx.h"
#include "CMining.h"
CMining::CMining() {
    i = 1;
```

```
    npValue = 0;
}
double CMining::get_annualFixedCost()
{
    return annualFixedCost;
}
void CMining::set_annualFixedCost(double value)
{
    annualFixedCost = value;
}
double CMining::get_discountedRate()
{
    return discountedRate;
}
void CMining::set_discountedRate(double value)
{
    discountedRate = value;
}
double CMining::get_millingCapacity()
{
    return millingCapacity;
}
void CMining::set_millingCapacity(double value)
{
    millingCapacity = value;
}
double CMining::get_millingCost()
{
    return millingCost;
}
void CMining::set_millingCost(double value)
{
    millingCost = value;
}
double CMining::get_miningCapacity()
{
    return miningCapacity;
}
void CMining::set_miningCapacity(double value)
{
    miningCapacity = value;
}
double CMining::get_miningCost()
{
    return miningCost;
}
void CMining::set_miningCost(double value)
{
    miningCost = value;
}
double CMining::get_recovery()
{
    return recovery;
}
void CMining::set_recovery(double value)
{
    recovery = value;
}
double CMining::get_refiningCapacity()
{
    return refiningCapacity;
}
void CMining::set_refiningCapacity(double value)
{
    refiningCapacity = value;
}
double CMining::get_refiningCost()
{
    return refiningCost;
}
void CMining::set_refiningCost(double value)
{
    refiningCost = value;
}
double CMining::get_sellingPrice()
{
    return sellingPrice;
}
void CMining::set_sellingPrice(double value)
{
    sellingPrice = value;
}
double CMining::get_NPV()
{
    return npValue;
}
void CMining::set_NPV(double value)
{
    npValue = value;
}
double CMining::get_Gc()
{
    return g_c;
}
double CMining::get_Gmc()
{
    if (g_mc <= g_m){
        Gmc = g_m;
    }else if (g_mc >= g_c){
        Gmc = g_c;
    }else {
        Gmc = g_mc;
    }
    return Gmc;
```

```
}
double CMining::get_Gr()
{
    return g_r;
}
double CMining::get_Gmr()
{
    if (g_mr <= g_m){
        Gmr = g_m;
    }else if (g_mr >= g_r){
        Gmr = g_r;
    }else {
        Gmr = g_mr;
}
    return Gmr;
}
double CMining::get_Gm()
{
    return g_m;
}
double CMining::get_Grc()
{
    if (g_rc <= g_r){
        Grc = g_r;
    }else if (g_rc >= g_c){
        Grc = g_c;
    }else {
        Grc = g_rc;
}
    return Grc;
}
void CMining::calc_Gc()
{
    g_c = (millingCost + (annualFixedCost +
    npValue*discountedRate)/millingCapacity)/((sellingPrice -
    refiningCost)*recovery);
}
void CMining::calc_Gm()
{
    g_m = millingCost/((sellingPrice -
    refiningCost)*recovery);
}
void CMining::calc_Gr()
{
    g_r = millingCost/((sellingPrice - refiningCost -
    (annualFixedCost+npValue*discountedRate)/refiningCapa
    city)*recovery);
}
    void CMining::calc_Gmc(double mc_k, double mc_k1,
    double g_k, double g_k1)
{
    g_mc = (((millingCapacity/miningCapacity)-
    mc_k)/((mc_k1 - mc_k)/(g_k1 - g_k)))+g_k;
}
    void CMining::calc_Gmr(double mr_k, double mr_k1,
    double g_k, double g_k1)
{
    g_mr = (((refiningCapacity/miningCapacity)-
    mr_k)/((mr_k1 - mr_k)/(g_k1 - g_k)))+g_k;
}
    void CMining::calc_Grc(double g_k, double g_k1, double
    rc_k, double rc_k1)
{
    g_rc = (((refiningCapacity/millingCapacity)-rc_k)/((rc_k1
    - rc_k)/(g_k1 - g_k)))+g_k;
}
double CMining::get_Gopt()
{
    get_Grc();
    get_Gmr();
    get_Gmc();
    if (Gmc >= Gmr){
        if (Gmc >= Grc){
        if (Gmr >= Grc){
            return Gmr;
        }else {
            return Grc;
            } else {
            return Gmc;
        }else {
        if (Gmr >= Grc){
        if (Gmc >= Grc){
            return Gmc;
        }else {
            return Grc;
        } else {
            return Gmr;
    }
    }
    }
double CMining::get_limiting_N(double N_r, double N_c,
double N_m)
{
        double temp_val = 0.0;
        if (N_r >= N_c){
            temp_val = N_r;
        } else {
            temp_val = N_c;
        if (temp_val >= N_m){
            return temp_val;
        } else {
            return N_m;
    }
    }
```