Bed blending design incorporating multiple regression modelling and genetic algorithms

by M. Kumral*

Synopsis

The efficiency of an ore-processing unit depends on the consistency of the characteristics of raw material entering the plant. When the mined ore is highly variable in quality, the only way to ensure consistency is to homogenize the ore prior to feeding to the processing plant. The homogenization can generally be achieved by the bed blending operation. Given that the stockpiling and reclamation processes are very expensive, it is necessary to design the process in such a way as to minimize variabilities of specified properties of raw material. In this paper, for alternative stacking types, optimal stockpile geometry is found in three stages: First, stockpile input is simulated by sequential Gaussian simulation, and then the variance reduction ratios (*VRR*) as a criterion of stockpile efficiency are calculated for various stockpile geometry scenarios by a stockpile simulator written in FORTRAN. Second, multiple regression analysis is performed to model the *VRR* by the use of stockpile length, the number of layers and stacker speed as the independent variables. Finally, the model is an optimization problem. Decision variables are the stockpile length, the number of layers, stacker speed and stacking type. The genetic algorithms (GA) are used to minimize the *VRR*. The approach was demonstrated on data from an iron orebody. The problem was to reduce fluctuations of iron, silica, alumina and lime contents in the stockpile output. The results showed that the approach could be used for the bed blending design efficiently.

Keywords: bed blending design, multiple regression analysis, genetic algorithms, iron ore, content fluctuation

Introduction

Many industries (e.g., cement, iron-steel or coal fired power generation) that consume mined products require homogenization of the raw material prior to consumption so as to reduce product variability. Stockpiling is the most widely used method of homogenization. A stacking machine usually constructs stockpiles. Reclamation is achieved by slicing across the pile orthogonal to the direction of layering. The efficiency of a stockpile system depends upon three factors:

- Stockpiling method is defined by the type of stacking used for stockpiling such as chevron or windrow
- Stockpiling parameters are stockpile length, width, the number of layers, the size and shape of the slice and layer, the equipment properties of the stacker and the reclaimer, raw material characteristics such as bulk density, particle size and oxidation
- Variability of stockpile input is a function of the stockpile efficiency.

The objective is to design a stockpile that will minimize the grade variation of stockpile output. One way to measure for the stockpiling efficiency is *VRR*:

$$VRR = \frac{\sigma^2_{out}}{\sigma^2_{in}} \qquad [1]$$

where σ^2_{in} and σ^2_{out} are the variances of the grades of stockpile input and output, respectively.

Bed blending design has been a well-known problem in mineral industries and many researches were devoted to designing optimal stockpile. The theory of bed blending was developed by Gy's pioneering work[1]. The mathematical modelling based on geostatistical approach or time series analysis was used to predict the output characteristics of a particular stockpile[2-9]. Geostatistical modelling was based on the quantification of variability through the semi-variogram and this quantification was used for the prediction of output variability.

This research can be categorized into three stages:

- The best way to understand a stockpile is to imagine it as a set of discrete blocks created by the intersection of slices and layers. Note that the term 'block' used throughout the paper refers to weight. It may alternatively be related to volume or time. Geostatistical simulation is used to generate block realizations of variables on any specified scale. A stockpile simulator has been, on the basis of the

* *Inonu University, Engineering Faculty, Mining Engineering Department, Malatya, Turkey.*

© *The South African Institute of Mining and Metallurgy, 2006.* *SA ISSN 0038–223X/3.00 + 0.00. Paper received Aug. 2005; revised paper received Jan. 2006.*

simulated block grades, developed to calculate input and output variances for various stockpiling geometries and types.

- Multiple regression analysis has been used to model each variable with respect to *VRR*s obtained by the stockpile simulator. The number of blocks in each layer and slices, stacker speed and stacking type are the independent variables (x). The *VRR* is dependent variable (y). The modelling has been performed for iron, silica, alumina and lime.
- This problem has a multi-objective nature. Four individual models have been transformed into single objective using a weighting scheme. The GA has been used to find optimal stockpile geometry for alternative stacking types.

Development of the stockpile simulator

Geostatistical simulation can be used to assess uncertainty in the stockpiling design. It is impossible to completely know the contents of input blocks because of sparse samples. Geostatistical simulation does not create data; it simply provides one possibility (among an infinite number) of what may actually be present at non-sampled locations. The ore stream is simulated by weight increments by using the sequential Gaussian simulation method[10,11].

The stockpile capacity is an important parameter and is constrained by mining and processing capacity. In stockpiling operation the *VRR* should usually be minimized for multiple variables, and hence the number of variables is also a significant parameter.

The stockpile simulation can be implemented for chevron or windrow methods. In the chevron method, the raw material is stacked by coming and going. This is the most common method in which stacking is done in both directions. Chevron stockpiling tends to particle segregation. Coarser particles gravitate towards the base of the pile in chevron. The windrow method stacks the ore in triangular rows, using multiple peaks. It entails more intricate stacking since the stockpile is formed in several small parallel rows. The method involves the traversing of the stacker much more frequently, and fluctuations in size distribution are most effectively minimized.

In both methods, the stockpile length, the number of layers and stacker speed are given in terms of the number of blocks. The number of blocks at each stacking layer (x_1), the number of layers (x_2) and the number of blocks formed by the layer pattern of a reclaiming slice (x_3) are initiated as decision variables. The stockpile capacity is given as:

$$cap = x_1 * x_2 * x_3 \qquad [2]$$

Input and output variances used to calculate the *VRR* should refer to identical supports i.e. identical weight or volume of material, otherwise the ratio is meaningless. Since the output variance is calculated on the basis of the dispersion variance of reclaimed slices, the input variance must be calculated on the basis the dispersion variance of the averages of the grades of input blocks equal to the number of blocks formed a reclaiming slice. The ratio of these two variances yields *VRR* or the homogenization effect.

A new scenario of decision variables (x_1, x_2 and x_3) is generated for each corresponding stockpiling type. A new *VRR* is yielded. The procedure is repeated for each variable (iron, silica, alumina or lime). Chevron and windrow methods are illustrated in Figure 1. After generating a sufficient number of scenarios and obtaining a sufficient number of *VRR*s, these scenarios and ratios are used to model the stockpiling operation by multiple regression analysis.

Modelling with multiple regression analysis

Given a stockpile geometry and stacker speed, *VRR*s can be predicted by multiple regression modelling that is a statistical methodology that is used to relate variables. A variable called dependent or response variable is related to one or more so-called predictor or independent variable(s). The objective is to construct a regression model relating dependant variable, y, to independent variable(s), $x_1, x_2,\ldots, x_p$.

The multiple regression model relating y to $x_1, x_2,\ldots, x_p$ is

$$y_i = \mu_i + \varepsilon_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \varepsilon_i \qquad [3]$$

where:

μ_i is the mean value of the dependent variable when the values of the independent variable when $x_1, x_2,\ldots, x_p$ are $x_{i1}, x_{i2},\ldots, x_{ip}$.

$\beta_0,\beta_1,\ldots,\beta_p$ are unknown parameters relating μ_i to $x_{i1},x_{i2},\ldots,x_{ip}$

ε_i is an error term that determines the effect on y_i of all factors other than the values $x_{i1}, x_{i2},\ldots, x_{ip}$ of the independent variables $x_1, x_2,\ldots, x_p$.

The point prediction of [3] is

$$\hat{y}_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + \ldots + b_p x_{ip} \left(\varepsilon_i = 0\right) \qquad [4]$$

Next, the residuals is defined for i =1,2,…,n:

$$e_i = y_i - \hat{y}_i \qquad [5]$$

and the sum of squared residuals to be

$$SSE = \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2 \qquad [6]$$

Intuitively, if any values of b_0, b_1,….b_p are good point estimates, the predicted value $\hat{y}_i$ will be close to the observed value y_i and thus will make *SSE* fairly small. We define the least squares estimates to be values of b_0, b_1,….b_p that minimize *SSE*. The least squares estimates b_0, b_1,….bp are calculated by the formula:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_p \end{bmatrix} = \mathbf{b} = \left(\mathbf{X}\mathbf{X}^{\mathbf{T}}\right)^{-1} \mathbf{x}^{\mathbf{T}} \mathbf{y} \qquad [7]$$

where:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \text{ and } \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1p} \\ 1 & x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{np} \end{bmatrix} \qquad [8]$$

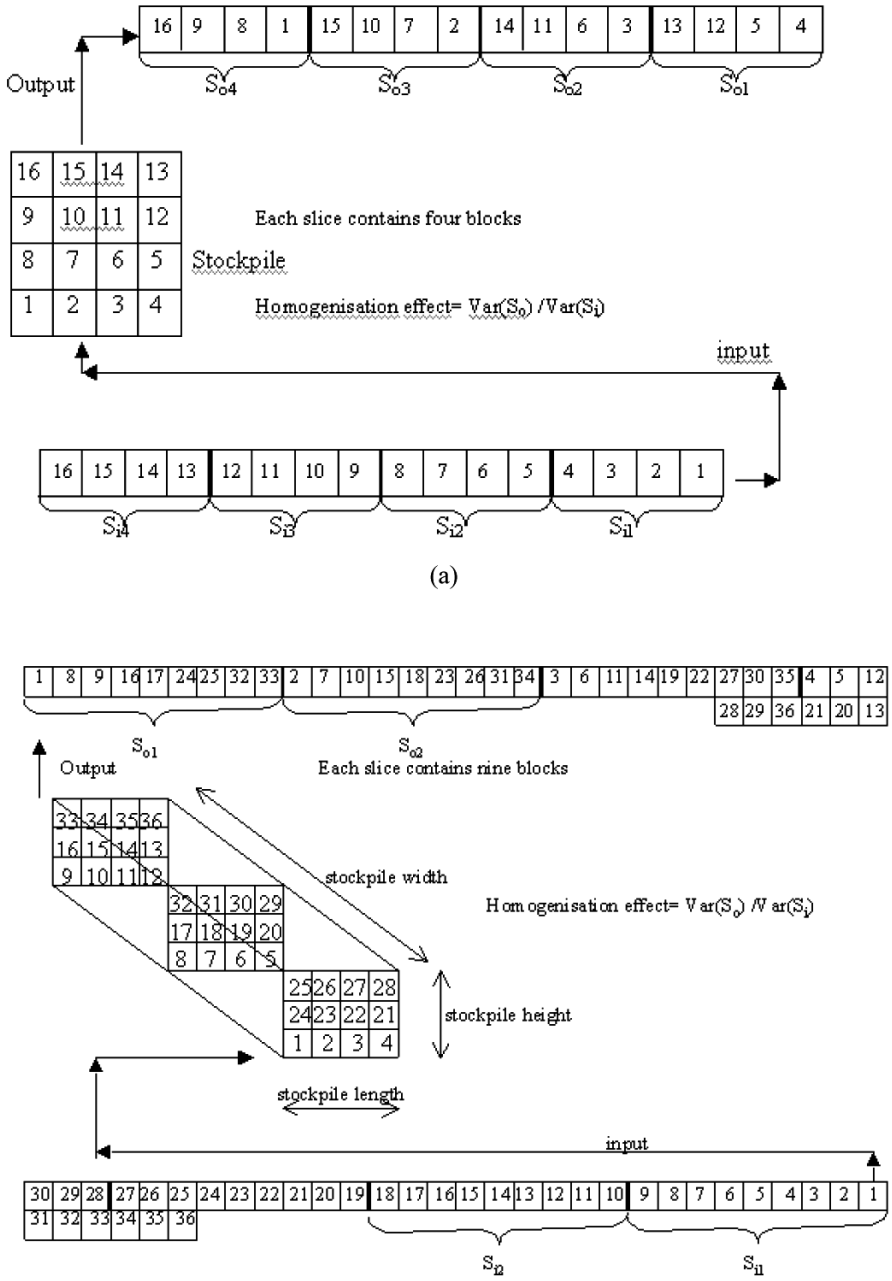

Figure 1—Illustrations of chevron (a) and windrow (b) methods

Optimization procudure

The question arising in this point is to determine optimal stockpile parameters by minimizing the model obtained by the multiple regression analysis so as to ensure minimum *VRR*. The problem is solved by the GA, which is a stochastic search algorithm that mimics the process of natural selection and genetics[12–16]. The GA has exhibited considerable achievement in yielding good solution to many complex optimization problems. When the objective functions are multi-model or the search space is irregular, highly robust algorithms are required so as to avoid trapping at local optima. The GA can reach the global optimum fairly. Furthermore, the GA does not require the specific mathematical analysis of the optimization problem. The GA is an iterative algorithm that yields a pool of solutions at each iteration. Firstly, the pool of initial solutions is generated at random. A new pool of solutions is created by the genetic operators at each new iteration. Each solution is evaluated with an objective function. This process is repeated until the convergence is reached.

A solution is called a chromosome or string. The GA with an initial set of randomly generated chromosomes is called a population. The number of individuals in the population is called the population size. The objective function is known as the evaluation or fitness function. A new population is created by the selection process using some sampling mechanism. An iteration of the GA is called a generation. All chromosomes are updated by the reproduction, crossover and mutation operators in each new generation. The revised chromosomes are also called the offspring.

The simple algorithm of the GA consists of the following steps:

1. Generate an initial population of strings
2. Evaluate the string according to the fitness function
3. Apply a set of genetic operators to generate a new population of strings
4. Go to Step 2 until a solution converges.

Problem description in GA

A binary or floating vector can be used as the representation structure in the GA. In this research a floating vector represents a real value of a decision variable as a chromosome because binary coding has received substantial criticism[17]. When the values of the decision variables are continuous, it is necessary to represent them by a floating vector. Furthermore, real-valued GA can ensure the values of decision variables to the full machine precision. The real valued GA also has the advantage of requiring less storage than the binary valued GA. As the number of bits in binary coding representation increases, the storage becomes important. The representation of the fitness function in real valued GA is also more accurate as a result.

The length of the vector of the floating number is the same as that of the solution vector. The chromosome $V=(x_1, x_2,\ldots,x_n)$ represents a solution $x = (x_1, x_2, \ldots, x_n)$ of the problem where n is the dimension. In order to solve the problem by the GA, each solution is coded by a chromosome $V(x_1, x_2,\ldots,x_n)$. A predefined integer population-size, which is the number of chromosomes, is initiated at random:

do i=1, population-size
chromosome$_i = x_j = (r_{ij} * (xupp_j - xlow_j)) + xlow_j$
(j = 1, jvar)
enddo

where:

r	is a random number between [0, 1]
x	decision variables
$xlow_j$ and $xupp_j$	are lower and upper limits in variable j
jvar	is the number of decision variables attributed to stockpile design

Until the predetermined population size is reached, the feasible solutions are accepted as chromosomes in the population. Then the fitness value of each chromosome is calculated. The chromosomes are rearranged in ascending order on the basis of the fitness values.

Now the parameter, a, is initiated in the genetic system. The rank-based evaluation function is defined as follows:

$$E(V_i) = a(1-a)^{i-1} \qquad i = 1, 2, \ldots, population\text{-}size \qquad [9]$$

When $i = 1$ represents the best individual, $i = population\text{-}size$ is the worst individual. The reproduction operator used herein is a biased roulette wheel, which is spun population-size times. A single chromosome is selected in each spinning for a new population. The roulette wheel is a fitness-proportional selection. The selection process is as follows:

1. The cumulative probability q_i is calculated for each chromosome:

$$q = \begin{cases} q_0 = 0 & if\ i = 0 \\ q_i = \sum_{j=1}^{i} E(V_i) & if\ i = 1, \ldots, population\text{-}size \end{cases}$$

2. A random number r is drawn in $(0, q_{population\text{-}size})$
3. The chromosome V_i is selected such that $q_{i-1} < r \le q_i$
4. The second and third steps are repeated *population-size* times.

This population is updated by the crossover and mutation operators. First of all, the crossover probability, P_c, is defined. $P_c * population\text{-}size$ gives the expected value of the number of chromosomes undergoing the crossover process. In order to carry out this process, random numbers, r_i, are generated from the interval [0, 1] in $i = 1, population\text{-}size$. If r_i is smaller than P_c, V_i is selected as a parent. The selected chromosomes are randomly grouped as pairs. If the number of selected chromosomes is odd, one of them is removed from the system. The crossover procedure is performed on each pair. Let the pair (V_1, V_2) be subjected to the crossover operation. Firstly, a random number, r, is generated from the interval (0, 1). Then the crossover operator will yield two children X and Y as follows:

$$X = r * V1 + (1-r) * V2 \text{ and} \qquad Y = (1-r) * V1 + r * V2 \qquad [10]$$

The feasibility of each child is checked. If feasible, the child is accepted.

The mutation operator is implemented on new version of population. Similar to the crossover operation, a mutation probability, P_m, is defined. $P_m * population\text{-}size$ gives the expected value of number of chromosomes undergoing the mutation operation. In this procedure a random number, r_i, is generated i=1 to *population-size* from the interval [0, 1]. If r_i is smaller than P_m, V_i is selected as a parent for the mutation. A random direction, d, is generated in $\Re^3$. The selected parent will be mutated by $V + M*d$ in this direction. A proper large number, M, is also initiated in this section.

The feasibility of each new chromosome is ensured as follows:

generate a random number, r_a
sum=1.
do j=1, *jvar*-1
$x_j = (r_a*(xupp_j - xlow_j))+xlow_j$
$sum=sum*x_j$
enddo
sp=(cap / sum) **(1.0 / (*nvari*-1))
do m=1, *jvar*-1
$x_m=x_m*s_p$
if $(x_m>xlow_m$ *.and.* $x_m<xupp_m)$ *feasible*
otherwise unfeasible and go to generate a new random number, r_a
enddo
continue until population-size is reached

where:

xlow and *xupp*	are lower and upper search limits in x and y coordinates
cap	is the stockpile capacity

If $V + M*d$ is not feasible to the constraints, M is set as a random number from interval [0, M] until it is feasible. If this procedure does not manage to find a feasible solution in a predetermined number of iterations, M is set to zero.

Thus one generation is completed. The whole procedure is implemented up to the predetermined number of iterations. After running the program, the best solution is reported as the results yielding the minimum *VRR*.

Case study

The stockpile input was simulated at 30 000 locations, each of which was one ton, by sequential Gaussian simulation that was implemented for iron, silica, alumina and lime variables. In order to show the effect of dependency on *VRR*, auto-correlations values for various lags were calculated and are given in Table I. As seen from Table I, lime and silica show stronger correlation than iron and alumina. Therefore, it is expected that the stockpiling procedure has more influence on lime and silica. This expectation was supported by the results (Table IV). As the weights of lime and silica increase, *VRRs* are minimized. For each variable, 15 stockpile scenarios were generated in both chevron and windrow methods and *VRRs* were calculated (Table II). Each variable was modelled by multiple regression analysis. The model summaries were as depicted in Table III for iron, silica, alumina and lime, respectively. For each variable, the best model was found to be:

$$VRR_{iron} = 0.507 - 6.96 * 10^{-6} * x_1 * x_2 + 1.612 * 10^{-4} * x_2 * x_3 - 0.167 * x_4 \quad [11]$$

$$VRR_{silica} = 1.44 * 10^{-4} * x_1 + 7.552 * 10^{-4} * x_2 + 1.288 * 10^{-2} * x_3 + 1.656 * 10^{-4} * x_1 * x_3 - 0.163 * x_4 \quad [12]$$

$$VRR_{alumina} = 0.435 - 3.96 * 10^{-4} * x_1 + 1.85 * 10^{-4} * x_1 * x_3 - 0.356 * x_4 \quad [13]$$

$$VRR_{lime} = 2.033 * 10^{-2} * x_3 + 3.145 * 10^{-5} * x_1 * x_3 - 2.1 * 10^{-2} * x_3 * x_4 \quad [14]$$

The objective is to minimize these functions. Four objectives were incorporated into the objective function as:

$$\text{Minimize} \sum_{i=1}^{m\,var} \lambda_i * VRR_i \quad [15]$$

where λ_i is the weight of variable *i*. The problem was now a multiple-objective decision-making problem. The problem could be transformed into a single objective problem using the weighting method[18]. Weights express the decision-maker' preferences. In this section the objective function was derived for setting 0.25 to each objective as weight and the objective function was transformed when the Equations [11–14] were substituted in Equation [16]. The objective function is expressed as:

Table I

Auto-correlations of simulated values

Lag	Iron	Silica	Alumina	Lime
1	0.38	0.67	0.24	0.85
2	0.30	0.57	0.17	0.79
4	0.19	0.43	0.09	0.71
8	0.09	0.26	-0.01	0.54
16	0.00	0.11	-0.05	0.32
32	-0.10	-0.01	-0.12	0.08

Table II

(a) VRR_{iron}, (b) VRR_{silica}, (c) $VRR_{alumina}$ and (d) VRR_{lime} values for various scenarios

(a)

x_1	x_2	x_3	x_4	VRR_{iron}
50.00	50.00	12.00	.00	.602
75.00	50.00	8.00	.00	.495
150.00	40.00	5.00	.00	.580
200.00	30.00	5.00	.00	.546
100.00	100.00	3.00	.00	.519
240.00	25.00	5.00	.00	.486
300.00	50.00	2.00	.00	.358
80.00	75.00	5.00	.00	.559
500.00	60.00	1.00	.00	.292
180.00	83.00	2.00	.00	.437
120.00	50.00	5.00	.00	.458
225.00	44.00	3.00	.00	.460
400.00	15.00	5.00	.00	.462
375.00	40.00	2.00	.00	.407
280.00	53.00	2.00	.00	.385
50.00	119.00	5.00	1.00	.425
79.00	126.00	3.00	1.00	.330
157.00	95.00	2.00	1.00	.227
220.00	136.00	1.00	1.00	.125
100.00	100.00	3.00	1.00	.289
250.00	120.00	1.00	1.00	.180
329.00	91.00	1.00	1.00	.171
85.00	117.00	3.00	1.00	.280
545.00	55.00	1.00	1.00	.270
195.00	51.00	3.00	1.00	.340
129.00	231.00	1.00	1.00	.150
247.00	60.00	2.00	1.00	.225
454.00	66.00	1.00	1.00	.160
384.00	78.00	1.00	1.00	.120
282.00	105.00	1.00	1.00	.096

(b)

x_1	x_2	x_3	x_4	VRR_{silica}
50.00	50.00	12.00	.00	.337
75.00	50.00	8.00	.00	.191
150.00	40.00	5.00	.00	.276
200.00	30.00	5.00	.00	.379
100.00	100.00	3.00	.00	.240
240.00	25.00	5.00	.00	.216
300.00	50.00	2.00	.00	.173
80.00	75.00	5.00	.00	.175
500.00	60.00	1.00	.00	.204
180.00	83.00	2.00	.00	.161
120.00	50.00	5.00	.00	.190
225.00	44.00	3.00	.00	.207
400.00	15.00	5.00	.00	.481
375.00	40.00	2.00	.00	.264
280.00	53.00	2.00	.00	.199
50.00	119.00	5.00	1.00	.021
79.00	126.00	3.00	1.00	.019
157.00	95.00	2.00	1.00	.016
220.00	136.00	1.00	1.00	.022
100.00	100.00	3.00	1.00	.021
250.00	120.00	1.00	1.00	.039
329.00	91.00	1.00	1.00	.022
85.00	117.00	3.00	1.00	.016
545.00	55.00	1.00	1.00	.041
195.00	51.00	3.00	1.00	.036
129.00	231.00	1.00	1.00	.047
247.00	60.00	2.00	1.00	.022
454.00	66.00	1.00	1.00	.034
384.00	78.00	1.00	1.00	.054
282.00	105.00	1.00	1.00	.025

Table II

(a)VRR_{iron}, (b) VRR_{silica}, (c) $VRR_{alumina}$ and (d) VRR_{lime} values for various scenarios (continued)

(c)

x_1	x_2	x_3	x_4	$VRR_{alumina}$
50.00	50.00	12.00	.00	.550
75.00	50.00	8.00	.00	.479
150.00	40.00	5.00	.00	.610
200.00	30.00	5.00	.00	.617
100.00	100.00	3.00	.00	.558
240.00	25.00	5.00	.00	.580
300.00	50.00	2.00	.00	.282
80.00	75.00	5.00	.00	.575
500.00	60.00	1.00	.00	.193
180.00	83.00	2.00	.00	.459
120.00	50.00	5.00	.00	.478
225.00	44.00	3.00	.00	.451
400.00	15.00	5.00	.00	.613
375.00	40.00	2.00	.00	.415
280.00	53.00	2.00	.00	.376
50.00	119.00	5.00	1.00	.035
79.00	126.00	3.00	1.00	.059
157.00	95.00	2.00	1.00	.049
220.00	136.00	1.00	1.00	.044
100.00	100.00	3.00	1.00	.027
250.00	120.00	1.00	1.00	.040
329.00	91.00	1.00	1.00	.042
85.00	117.00	3.00	1.00	.051
545.00	55.00	1.00	1.00	.060
195.00	51.00	3.00	1.00	.088
129.00	231.00	1.00	1.00	.042
247.00	60.00	2.00	1.00	.051
454.00	66.00	1.00	1.00	.057
384.00	78.00	1.00	1.00	.039
282.00	105.00	1.00	1.00	.029

(d)

x_1	x_2	x_3	x_4	VRR_{lime}
50.00	50.00	12.00	.00	.27600
75.00	50.00	8.00	.00	.16800
150.00	40.00	5.00	.00	.11300
200.00	30.00	5.00	.00	.17000
100.00	100.00	3.00	.00	.07170
240.00	25.00	5.00	.00	.12600
300.00	50.00	2.00	.00	.05730
80.00	75.00	5.00	.00	.09860
500.00	60.00	1.00	.00	.05150
180.00	83.00	2.00	.00	.06900
120.00	50.00	5.00	.00	.12200
225.00	44.00	3.00	.00	.05350
400.00	15.00	5.00	.00	.17100
375.00	40.00	2.00	.00	.06130
280.00	53.00	2.00	.00	.04550
50.00	119.00	5.00	1.00	.00640
79.00	126.00	3.00	1.00	.00370
157.00	95.00	2.00	1.00	.01470
220.00	136.00	1.00	1.00	.00537
100.00	100.00	3.00	1.00	.00666
250.00	120.00	1.00	1.00	.00510
329.00	91.00	1.00	1.00	.00822
85.00	117.00	3.00	1.00	.00370
545.00	55.00	1.00	1.00	.01460
195.00	51.00	3.00	1.00	.02400
129.00	231.00	1.00	1.00	.00340
247.00	60.00	2.00	1.00	.00590
454.00	66.00	1.00	1.00	.01100
384.00	78.00	1.00	1.00	.00730
282.00	105.00	1.00	1.00	.01000

Minimize

$$VRR_m = 0.2355 - 0.000063 * x_1 + 0.00018875 * x_2 + 0.0083025 * x_3 - 0.1715 * x_4 - 0.00000174 * x_1 * x_2 + 0.0000955125 * x_1 * x_3 + 0.0000403 * x_2 * x_3 - 0.00525 * x_3 * x_4 \quad [16]$$

Subject to

$$xlow_j \le x_j \le xupp_j \ (j = 1, jvar) \quad [17]$$

$$x_1 * x_2 * x_3 = cap \quad [18]$$

$$x_4 = \begin{cases} 0 \text{ if } chevron \\ 1 \text{ if } windrow \end{cases} \quad [19]$$

$$\sum_{i=1}^{j\,\mathrm{var}} \lambda_i = 1 \quad [20]$$

$$VRR_m \ge 0 \ \ \forall_m \quad [21]$$

In order to solve the problem with the GA, a computer program was written. The input parameter file of the program is given in Figure 2. There is no clear rule for the selection of control parameters (the population size, the crossover and mutation probability). Therefore, the parameters were determined by the experimentation. The fitness values versus the values of parameters (crossover, mutation and population size) are given in Figure 3. It was observed that small population size led the GA to quickly converge at a local optimum. On the other hand, large population size is prohibitively time-consuming. High crossover and mutation probability converted the GA into a random search. Low crossover and mutation probability caused trapping at local optima. Figure 4 shows how the fitness value changes with the number of iterations. The procedure was repeated 1750 times in approximately five minutes. The best solutions are given in Table IV for various weighting sets.

Conclusions

The approach incorporating multiple regression modelling and the GA can be used to design mineral stockpiles. This approach yields optimum stockpile parameters and provides a means to compare chevron and windrow layering types. The level of auto-correlation of input variable also has an influence on the *VRR*. As weights of lime and silica exhibiting strong auto-correlation increase, *VRR* yields better results. As observed from Table III, beta values indicated that stacker speed was relatively more significant than other parameters. The level of auto-correlation of the input variable also has an influence on the *VRR*. It is observed that as the degree of auto-correlation increases, the *VRR* will be minimized. One can clearly observe that the windrow method is more efficient than the chevron method. However, bear in mind that the windrow is more expensive than the chevron method. Note that geostatistical simulation provides a response distribution by generating multiple images, i.e. a series of optimal stockpile designs based on possible realizations of the input grades. Geostatistical simulation

Table III

Multiple regression results (a) for iron, (b) for silica, (c) for alumina and (d) for lime

(a)

Variable	Estimated Coefficient	t-statistics	Variance Inflation Factor	Beta
Constant	0.507	17.039		
x_1x_2	$-6.96*10^{-6}$	-5.154	2.585	-0.463
x_2x_3	$1.612*10^{-4}$	2.078	1.825	0.157
x_4	-0.167	-7.597	1.710	-0.556
Adjusted R^2	0.909			
F-statistics	97.887			

(b)

Variable	Estimated Coefficient	t-statistics	Variance Inflation Factor	Beta
x_1	$1.440*10^{-4}$	2.323	5.519	0.204
x_2	$7.552*10^{-4}$	4.340	4.897	0.359
x_3	$1.288*10^{-2}$	3.670	4.006	0.274
x_4	-0.163	-8.461	3.858	-0.621
x_1x_3	$1.656*10^{-4}$	5.711	7.174	0.571
Adjusted R^2	0.958			
F-statistics	138.380			

(c)

Variable	Estimated Coefficient	t-statistics	Variance Inflation Factor	Beta
Constant	0.435	12.362		
x_1	$-3.96*10^{-4}$	-4.032	1.183	-0.228
x_1x_3	$1.85*10^{-4}$	4.226	1.675	0.284
x_4	-0.356	-11.831	1.534	-0.762
Adjusted R^2	0.922			
F-statistics	114.731			

(d)

Variable	Estimated Coefficient	t-statistics	Variance Inflation Factor	Beta
x_3	$2.033*10^{-2}$	21.622	2.748	0.896
x_1x_3	$3.145*10^{-5}$	5.631	2.545	0.225
x_3x_4	$-2.10*10^{-2}$	-13.684	1.196	-0.374
Adjusted R^2	0.981			
F-statistics	524.435			

```
30                                                  \number of chromosome
 1750                                               \number of iterations
 0                                                  \stockpile type (0=chevron,1=windrow)
 4                                                  \number of variables (length, height, speed and type)
 50.00 400.00                                       \lower and upper limits for stockpile length
 50.00 300.00                                       \lower and upper limits for stockpile height
  1.00   20.00                                      \lower and upper limits for stacker speed
  0.00    1.00                                      \lower and upper limits for stockpile type
 30000                                              \capacity
 0.05                                               \parameter (a-(1-a)i-1)
 0.20                                               \crossover probability
 10.0  10.0  1.0  1.0                               \a large positive number (M)
 0.15                                               \mutation probability
 0.25  0.25  0.25  0.25                             \weights used in multiobjective case
 16                                                 \the number of parameters
 0.5070000000   0.0000000000   0.4350000000   0.00000000000      \model coefficients 0
 0.0000000000   0.0001440000  -0.0003960000   0.00000000000      \model coefficients 1
 0.0000000000   0.0007552000   0.0000000000   0.00000000000      \model coefficients 2
 0.0000000000   0.0128800000   0.0000000000   0.02033000000      \model coefficients 3
-0.1670000000  -0.1630000000  -0.3560000000   0.00000000000      \model coefficients 4
-0.0000069600   0.0000000000   0.0000000000   0.00000000000      \model coefficients 12
 0.0000000000   0.0001656000   0.0001850000   0.00003145000      \model coefficients 13
 0.0000000000   0.0000000000   0.0000000000   0.00000000000      \model coefficients 14
 0.0001612000   0.0000000000   0.0000000000   0.00000000000      \model coefficients 23
 0.0000000000   0.0000000000   0.0000000000   0.00000000000      \model coefficients 24
 0.0000000000   0.0000000000   0.0000000000  -0.02100000000      \model coefficients 34
 0.0000000000   0.0000000000   0.0000000000   0.00000000000      \model coefficients 123
 0.0000000000   0.0000000000   0.0000000000   0.00000000000      \model coefficients 124
 0.0000000000   0.0000000000   0.0000000000   0.00000000000      \model coefficients 234
 0.0000000000   0.0000000000   0.0000000000   0.00000000000      \model coefficients 134
 0.0000000000   0.0000000000   0.0000000000   0.00000000000      \model coefficients 1234
```

Figure 2—Input parameter file

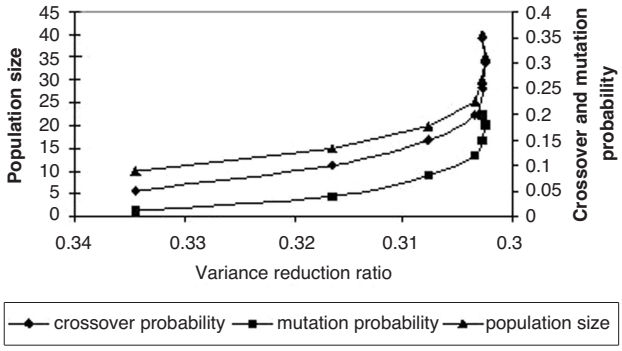

Figure 3—Determination of optimal GA parameters

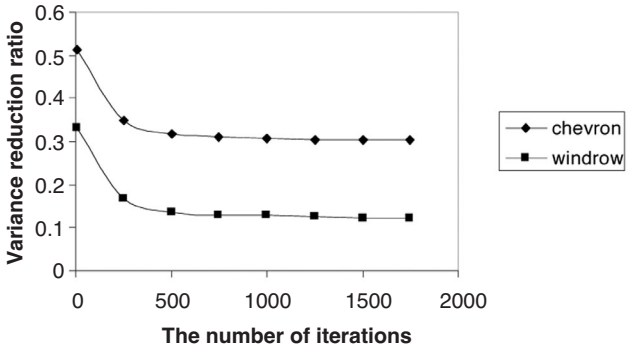

Figure 4—Variance reduction versus the number of iterations in chevron and windrow stockpiling methods

Table IV

Optimal results for various weighting sets in both chevron (a) and windrow (b)

(a)

λ_1	λ_2	λ_3	λ_4	x_1	x_2	x_3	*VRR* chevron
0.25	0.25	0.25	0.25	339.0692	88.1943	1.00321	0.2231409
0.15	0.35	0.15	0.35	275.8111	108.4383	1.00306	0.1773022
0.40	0.30	0.20	0.10	259.3114	114.7819	1.00792	0.2604867
0.10	0.20	0.30	0.40	332.8980	89.81108	1.00341	0.1899968
0.35	0.15	0.35	0.15	253.8817	117.7304	1.00369	0.2762195
1.00	0.00	0.00	0.00	203.6018	146.9666	1.00258	0.3224906

(b)

λ_1	λ_2	λ_3	λ_4	x_1	x_2	x_3	*VRR* window
0.25	0.25	0.25	0.25	280.5316	106.9015	1.000358	0.048495
0.15	0.35	0.15	0.35	285.3380	104.9892	1.001422	0.034165
0.40	0.30	0.20	0.10	259.5320	114.6551	1.008178	0.071475
0.10	0.20	0.30	0.40	332.8980	89.81108	1.003414	0.025468
0.35	0.15	0.35	0.15	294.9280	101.5937	1.001241	0.061633
1.00	0.00	0.00	0.00	203.6018	146.9666	1.002585	0.015549

does not create data; it builds alternative and equally probable realizations of variables at respective locations. Many alternate models can be generated and processed to construct a distribution of possible *VRR* for specified attributes. This distribution is used to assess the risk associated with the uncertainty at unsampled locations. In future research, cross-variograms can be used to provide spatial correlations of multiple variables. The stockpiling operation can reduce only variation among reclamation slices. It has no effect on variation among stockpiles. Therefore, the effect of extraction on homogeneity of the mineral characteristics should also be determined. This can be achieved by simulation of the mining (extraction) process.

Acknowledgement

The author would like to acknowledge the support by Inonu University Research Grant Scheme for this work (Project No: 2002 / 04).

References

1. Gy, P.M. *The Sampling of Heterogeneous and Dynamic Material Systems*, Elsevier, Amsterdam, 1992.
2. Serra, J., Huijbregets, C.H., and Ivanier, L. Laws of linear homogenization in ore stockyards, 11. *International Mineral Processing Congress*, Cagliari, 1975, pp. 263–291.
3. Gy, P.M. A New Theory of Bed-Blending Derived From The Theory of Sampling—Development and Full—Scale Experimental Check, *International Journal of Mineral Processing*, 1981, vol. 8, pp. 201–238.
4. Dowd, P.A. The design of a rock homogenising stockpile, *Mineral Processing in the UK*, IMM, London, 1989, pp. 63–82.
5. Pavloudakis, F.F., and Agioutantis, Z. Simulation of Bulk Solids Blending in Longitudinal Stockpiles, *International Journal of Surface Mining, Reclamation and Environment,* 2003, vol. 17, no. 2, p. 98–112.
6. Gerstell, A.W. and Werner, J.W. Computer simulation program for blending piles, *Bulk Solids Handling*, 1996, vol. 16, no. 1, pp. 49–58.
7. Kumral, M. Quadratic programming for the multivariable pre-homogenization and blending problem, *Journal of South African Institute of Mining and Metallurgy*, 2005, vol. 105, no. 5, pp. 317–322.
8. Robinson, G.K. How much would a blending stockpile reduce variation?, *Chemometrics and Intelligent Laboratory Systems,* 2004, vol. 74, no. 1, pp. 121–133.
9. Petersen, I.F. Blending in circular and longitudinal mixing piles, *Chemometrics and Intelligent Laboratory Systems*, 2004, vol. 74, no. 1, pp. 135–141.
10. Journel, A.G. and Alabert, F. Non-Gaussian data expansion in the earth science, *Terra Nova*, 1989 vol. 1, pp. 123–134.
11. Deutsch, C.V. and Journel, A.G. *GSLIB: Geostatistical Software Library and User's Guide*, Oxford Univ. Press, 1998.
12. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Pub. Co., 1989.
13. Reeves, K.R. Genetic algorithms, *Modern Heuristic Techniques for Combinatorial Problems*, C. Reeves (ed.), 1993, pp. 151–196.
14. Davis, L. *Handbook of Genetic Algorithms*, New York: V.N. Reinhold, 1991.
15. Ansari, N. and Hou, N. *Computational Intelligence for Optimization*, Kluwer Academic Pub. 1997.
16. Haupt, R.L. and Haupt, S.E. *Practical Genetic Algorithms*. John Wiley & Sons, 1998.
17. Liu, B. *Uncertain Programming*. John Wiley and Sons Inc., 1999
18. Cohon, J.L. *Multiobjective Programming and Planning*. Academic Press, 1978. ◆

Comment on the paper 'Bed blending design incorporating multiple regression modelling and genetic algorithms'

by M. Kumral

in the Journal of SAIMM, vol. 106, no. 3, pp.229–236

by G.K. Robinson*

The conclusion from this paper which I consider to be of greatest practical importance is that windrow stacking leads to a much greater variance reduction than chevron stacking. For instance, in his conclusion, Kumral writes: 'One can clearly observe that the windrow method is more efficient than the chevron method.'

I do not believe that this conclusion is correct. I believe that windrow stacking has an advantage over chevron stacking only when segregation is a problem. The conclusion is not consistent with the modelling reported in Robinson (2004), but a comparison between the modelling methodologies is not straightforward because that paper emphasized the end effects, which cause grade variation as the ends of stockpiles are reclaimed. (It should also be noted that all the modelling discussed in both papers ignores segregation.)

The point that I wish to make here is that Kumral's conclusion is not consistent with his own model.

Consider the building of a 36 kiloton stockpile in 9 stacker passes using Kumral's model, with the pile to be reclaimed in four reclaim slices. For chevron stacking, this is modelled as 36 blocks of ore, as in Figure 1, which is based on Kumral's illustration of chevron stacking in his Figure 1(a) but with 9 stacker passes rather than 4.

33	34	35	36
32	31	30	29
25	26	27	28
24	23	22	21
17	18	19	20
16	15	14	13
9	10	11	12
8	7	6	5
1	2	3	4

Figure 1—Diagram of chevron stacking of 36 blocks of ore to be reclaimed in four reclaim slices. Blocks are numbered in the order in which they are stacked

Each block is 1000 tons and is stacked over a quarter of the length of the stockpile. Its distribution over the width of the stockpile is not considered in the model.

The reclaim slices each consist of 9 blocks. They correspond to the columns in Figure 1.

Kumral's Figure 1(b) illustrates windrow stacking of the same 36 kilotons of material using a windrow stacker in 9 stacker passes, with the intention that the pile will be reclaimed in four reclaim slices.

These two models, one for chevron stacking and one for windrow stacking, have the same block sizes and the same sizes for reclaim slices. The sets of blocks, which get combined to form the four reclaim slices, are precisely the same for the two models. The first reclaim slice consists of the blocks 1, 8, 9, 16, 17, 24, 25, 32 and 33. The second reclaim slice consists of the blocks 2, 7, 10, 15, 18, 23, 26, 31 and 34. The third reclaim slice consists of the blocks 3, 6, 11, 14, 19, 22, 27, 30 and 35. The fourth reclaim slice consists of the blocks 4, 5, 12, 13, 20, 21, 28, 29 and 36. Hence the models give precisely the same predictions for blending performance.

This equivalence holds whenever the number of stacker passes is the same for the chevron and windrow models. It is not consistent with Kumral's conclusion that windrow stacking leads to a much greater variance reduction than chevron stacking.

* *CSIRO Mathematical and Information Sciences, Clayton South Victoria, Australia.*

© *The South African Institute of Mining and Metallurgy, 2006.* *SA ISSN 0038–223X/3.00 + 0.00. Originally published in March 2006. Comments received March 2006.*