GEOVIA WHITTLE SIMULTANEOUS OPTIMIZATION

White Paper

D. Place, GEOVIA R&D, Dassault Systèmes, G. Whittle, Whittle Consulting Pty. Ltd. and N. Baxter, GEOVIA R&D, Dassault Systèmes

SUMMARY

This paper describes the technical foundation of the Advanced Simultaneous Optimization (*Advanced SIMO*) module in GEOVIA Whittle™. *Advanced SIMO* module enables mine planners to create optimal long term schedules for the operation of open pit mines.[1] *Advanced SIMO* uses the ProberB engine that was developed by Whittle Development Pty Ltd. The method to find the optimal schedule is decomposed into three steps: 1) aggregate blocks into pushbacks, panels and blend bins to reduce the problem size; 2) express the scheduling problem into a linear system that can be solved through linear programing and uses an iterative approach to finding an optimal schedule; and 3) iterate semi-randomly through the solution space to ensure the schedule obtained is close to the optimal solution.

INTRODUCTION

Traditional methods used in strategic mine planning tend to optimize one parameter at a time, while Advanced Simultaneous Optimization (*Advanced SIMO*) considers all parameters and alternatives simultaneously, hence providing a near-optimal solution. Previous solutions that used *Milawa* for schedule optimization, *Stockpile & Cut-off Optimization* (SPCO) for cut-off and stockpile optimization, and multiple extractive blend scenarios to optimize blending are all superseded through the use of *Advanced SIMO* which optimizes these parameters all together to maximize profit.

The traditional pit-shell optimization (Lerch-Grossman or Pseudoflow) is still required to create the optimal pit shell. Pit shells are mostly independent of the schedule and stockpiling, and therefore can be done independently of an *Advanced SIMO* optimization. While this statement could be argued, the slope constraints and economics remain by far the main drivers for the shape of pit shell rather than the schedule.

A simultaneous optimization (SIMO) run will optimize one and only one scenario of pushback strategy. If one wants to optimize the pushback selection and strategy, that optimization could be done separately using the *pushback chooser*, or better, with sufficient computing resource by manually evaluating the different strategies, through skin analysis for instance.

From a set of input parameters (block model, economics and mining constraints, etc.), a user wants to calculate a schedule that maximizes profit through Net Present Value (NPV). The NPV is formulated as a function of those input parameters, the schedule and all the other variables. The schedule is the variable that has the most degrees of freedom and has the greatest impact on NPV. The schedule provides a description of which block to mine and when with the detail of its destinations (i.e., if it is to be processed, stockpiled or discarded). Finding the optimal mine plan can be basically reduced to finding the optimal schedule.

The optimal schedule for a mine can be described with a suite of linear equations and inequations with an objective function representing the NPV after the life of mine. The aim of the optimization is to solve all the variables and input parameters that maximize the objective function. If such a system is linear, and small enough, it can be solved using a traditional linear programming (LP) solver and an optimal solution is guaranteed. Unfortunately, in the case of most mine schedules, the system is neither small enough nor linear.

1 *Standard SIMO* module is limited to one pit, 10 pushbacks and 2 processing methods, while the *Advanced SIMO* module can optimise across a maximum of 15 pits, 50 pushbacks and 30 processing methods.

WHY IS SIMULTANEOUS OPTIMIZATION BETTER THAN TRADITIONAL METHODS?

The basic assumption made when optimizing one parameter separately from others is that the parameters are independent of each other. For instance, cut-off optimization assumes that the cut-off grades can be optimized independently of the schedule. However, changing the cut-off often means that the schedule that was once optimal may not be optimal for a different cut-off.

When optimizing in separate steps, a decision made at an early stage (e.g., Schedule in Figure 1) impacts later decisions, hence reducing or masking possibilities of a better solution altogether. The following sections, Steps 1 through 3, will briefly describe the logic in deriving an optimal mining schedule with SIMO.

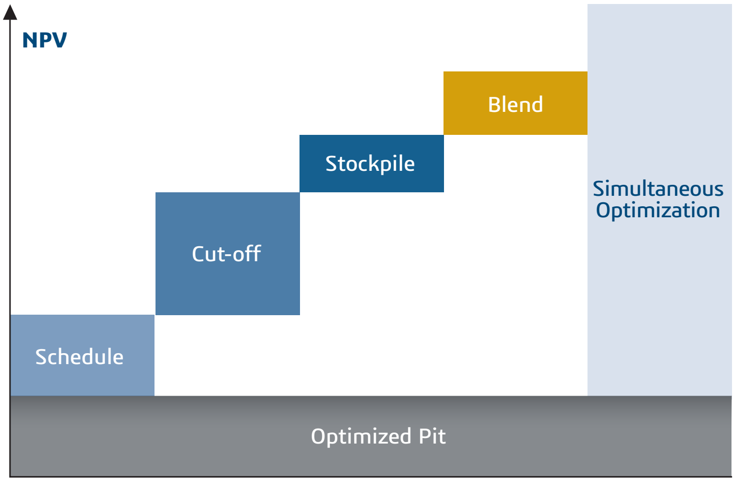

Figure 1
Gain in NPV using incremental method compared to gain in NPV with Simultaneous Optimization. Schedule, Cut-Off, Stockpile and Blend optimization provide incrementally lower NPV.

STEP 1: AGGREGATION TO "BLEND BINS"

The description of this scheduling problem into a system of equations is far too large for any computer to solve. To reduce the system size, GEOVIA Whittle uses "blend bins" as an aggregation methodology. Since blocks having similar grades of material will have a similar outcome, aggregating those blocks together will reduce the size of the problem significantly without impacting the results (at the expense of additional input from the user to specify the blend-bins).

Pits are subdivided into pushbacks, and into benches. A panel designates a specific bench within a pushback (see Figure 2), and the blocks of a specific pushback and bench are assigned to a panel. The definition of blend bins will allow grouping of the blocks of a panel that have similar grade characteristics into a one bin (Figure 3).

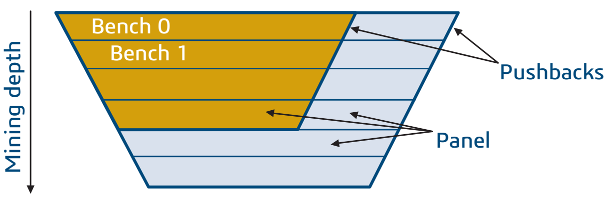

Figure 2

Figure 3

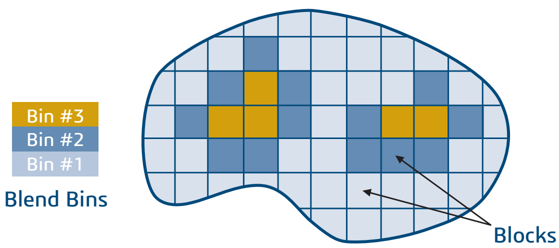

Figure 2
Pushback, Bench and panel decomposition of a Pit. A panel references a specific bench in a pushback.

Figure 3
Top view of a panel composed of three blend bins showing aggregation into blend bins of blocks in a panel.

The user defines blend bins by specifying the grade range for each bin. Note that the bin grade ranges, if incorrectly chosen, will have a significant impact on the optimization. Providing enough blend bins around the sensitive cut-offs is critical to the process. Users generally start off with a higher number of blend bins in order to determine where the ore or waste boundary sits, what groups of bins are processed, stockpiled or reclaimed together, and then consolidate them so that bins become material classes for scheduling.

Once the blocks have been aggregated to pushback, panel and bins, the size of the system is now reduced to an acceptable size. The added benefit in subdividing by pushback and panel is that the structure can now follow implicitly the mining constraints related to slope profile and block precedence.

STEP 2: FINDING AN OPTIMAL SOLUTION

To describe the approach let's have a look first at the case without stockpiles.

Case without stockpile

A schedule can be expressed by a vector $X_{i,j}$ which specifies the total depth mined for pushback *i* and period *j*, so that for every period of a schedule, one can know how much of each pushback is mined. For instance, in a simple case of a pit with one pushback, six benches, and three periods, the vector $X = (0.5, 3.2, 5.8)$ indicates that at end of period 1, half of the first bench (bench 0) is mined; at end of period 2 mining reached 20% of the 4th bench (bench 3); and end of Period 3 only 20% of the last bench is left.

If the schedule problem is expressed using the above principle while adding mining, blending and processing constraints, solving for $X_{i,j}$ will provide the optimal solution for the problem. However, the system cannot be solved as it is. It must be linear to be solvable using traditional LP solvers. If the benches and pushbacks to be mined within each period are fixed, one can make the system linear and find the optimal schedule for this specific case using a LP solver. The LP solver will then be able calculate what is the optimum fraction of the bench to mine. In our previous example, if (0.5, 3.2, 5.8) was an optimum schedule, that solution would have been found when we would fix bench 0 for period 1, bench 3 for period 2 and bench 5 for period 3. In other words, by providing the initial value $X = (0, 3, 5)$ the solver will be able to find the optimum solution (0.5, 3.2, 5.8) for that case. Fixing the benches and pushbacks to mine in each period is a serious constraint, and it is not practical nor useful to request that information from the user. Since the user does not know a priori which benches and pushbacks to mine within each period, one would have to iterate through all the combinations of benches and pushbacks that can be mined in all periods to find the optimal solution. In the previous example with one pushback, five benches and three periods, the software would need to iterate through 56 combinations.[2] That number will grow exponentially as more pushback, benches and periods are added. With just 10 benches, 10 periods and 5 pushbacks the number of combinations are close to 6.7×10^{24}. Hence, for a normal problem size, this yields too many combinations to evaluate. To solve this problem, an iterative approach analogous to Newton's Method (Numerical Recipes. The Art of Scientific Computing, 3rd Edition, 2007) is used to find a local optimal (see Figure 4 in next page).

2 There are $6^3=216$ permutations possible when choosing three benches from 1 to 6. Out of those 216 permutations, 56 are viable permutations when considering bench precedence rules. The number of combination considering *N* pushback, *P* period and *B* benches is: $\left(\frac{(P+B-1)!}{P!\,(B-1)!}\right)^N$

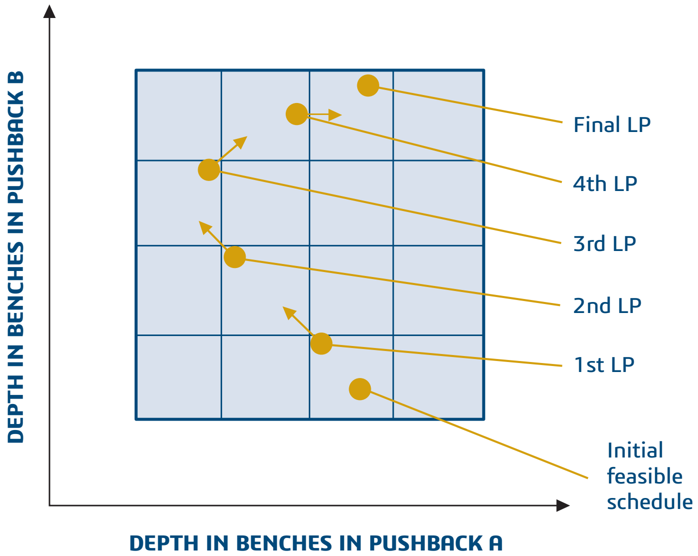

Figure 4
Iteration to find local maximum solution from an initial feasible solution.

As shown in Figure 4, starting from any initial feasible solution (the fixed set of benches and pushbacks to mine for each period), the LP solver calculates the optimal schedule; if the solution is within the bounds of the initial schedule (i.e. it has not reached the limits of benches and pushbacks that were fixed) then the program has found a local maximum; if not, the program restarts the process using this last solution as an initial feasible solution for the next iteration. This process can start from any initial feasible solution, allowing a local maximum to be found in the solution space.

Stockpiles

The use of stockpiles adds a level of complexity to the overall algorithm. Namely, the system of equations described previously is once again non-linear. The added input variables that are needed for the stockpiles will require changing the grade and tonnage constraints, and will need to be expressed as a product of two variables (e.g., $ax_1x_2 + \cdots = k$, where x_1 and x_2 represent the grade of the stockpile and the amount of material going out of the stockpile, respectively). The system is now non-convex quadratic and cannot be solved with a LP solver. Then the approach taken is to separately solve the stockpile grades and the stockpile movements. First, the grades of the stockpile (x_1) are iteratively estimated; then the stockpile grade is fixed to enable the LP solver to solve the schedule.

Note that this approach does not optimize the stockpile grades and the schedule together, so the stockpile grades may not be optimal for the schedule. However, since many blocks contribute to the stockpiles, the stockpile grade does not tend to change significantly with the schedule.

This approach is a significant improvement compared to SPCO which only optimizes the cut-off grades. Here the stockpile material movement and stockpile grades are optimized to deliver the best possible NPV.

STEP 3: FINDING THE OPTIMAL SOLUTION

The iterative approach described previously does not guarantee that the solution found is the optimal solution. Figure 5 illustrates the concept in which starting from different initial feasible schedules can lead to different local maximums, and that approach could return a local maximum instead of the optimal solution.

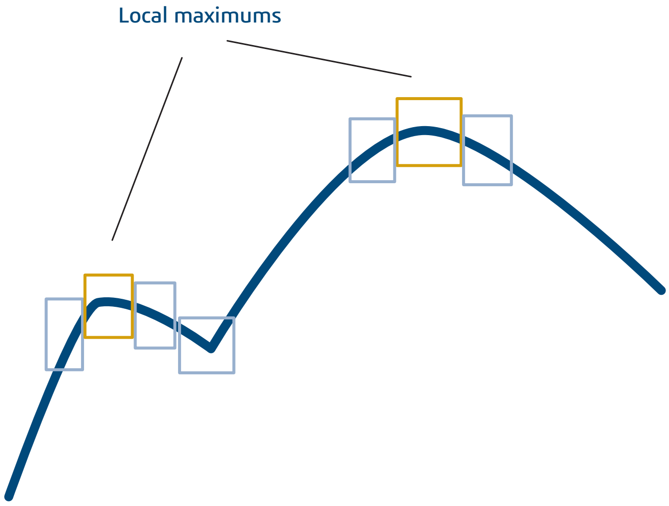

Figure 5
Global maximum vs local maximums.

Using a method analogous to Monte-Carlo method, the procedure is repeated a large number of times starting from different initial feasible solutions. Like a mountain climber reaching the summit of one peak, only to find that they were surrounded by other higher points of the mountain, this method places many mountain climbers randomly through the range and the final solution is obtained from the climber who summits at the highest peak.

Care has been taken to ensure that the full solution space is randomly sampled so that all local maximums are found. The iterations could be left to run for a long time, however the rate at which a better maximum is found decreases quickly, and the algorithm is stopped when the chance of finding a significantly better solution becomes largely improbable. (See Figure 6.)

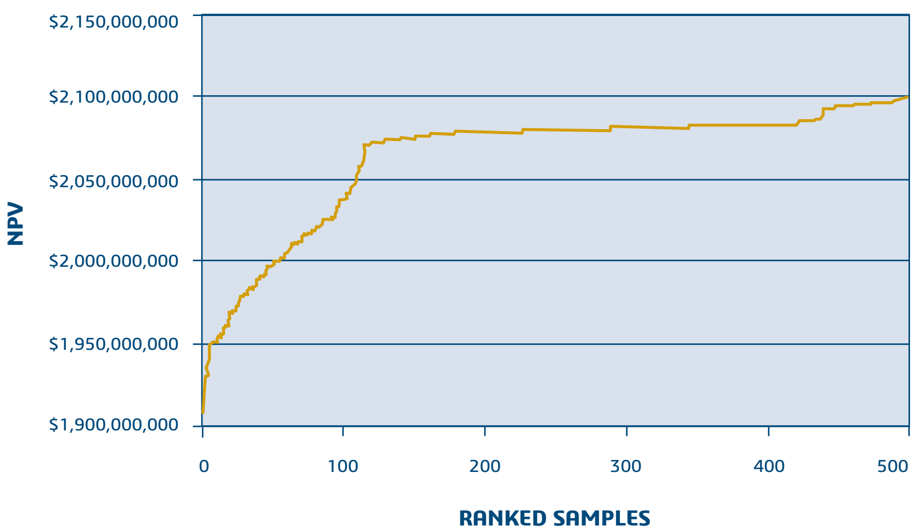

Figure 6
Maximum NPV as a function of iteration number.

CONCLUSION

The GEOVIA Whittle Advanced Simultaneous Optimization approach is able to find better outcomes compared to traditional methods. We have shown that the optimal mine plan for an open pit problem can be reduced to finding the optimal schedule through the simultaneous optimization of blends, stockpiles, and processing strategies within an iterative approach. The algorithm consists of iterating through the space of possible schedules to find the near optimal schedule that obeys the mining and processing constraints.

In a comparison study (Cf. Internal document: Advanced SIMO vs Milawa and SPCO, G. Whittle, 2016), it has been demonstrated that the schedule out of Advanced SIMO optimization was worth a total of $713 million or 41.4 percent additional NPV, whereas the legacy tools Milawa and SPCO showed only $487 million or 28.4 percent. This study was performed on the Marvin Copper & Gold example used in the Money Mining Seminar from Whittle Consulting Pty Ltd. We expect to find greater returns as more and more users apply the Advanced SIMO to their mining problems.

Americas
Dassault Systèmes
175 Wyman Street
Waltham, Massachusetts
02451-1223
USA

Europe/Middle East/Africa
Dassault Systèmes
10, rue Marcel Dassault
CS 40501
78946 Vélizy-Villacoublay Cedex
France

Asia-Pacific
Dassault Systèmes K.K.
ThinkPark Tower
2-1-1 Osaki, Shinagawa-ku,
Tokyo 141-6020
Japan