Flexible Optimization of Stope Boundaries

John Manchuk

Center for Computational Geostatistics
Department of Civil & Environmental Engineering
University of Alberta

Economic improvement of underground mining operations by various optimization techniques is a growing area of research. Sectors of an operation such as designing stopes and scheduling equipment can benefit from computational optimization techniques. For many operations, stopes are designed by hand using ore grade cutoff shells, geology maps and geotechnical constraints; however, these stopes may not be offering their full potential to the economics of the operation. One reason for this is a lack of available software. For surface mining operations, there is an abundance of optimization packages available for defining a pit or scheduling equipment. This scale of software is just not available for underground operations. Introduced in this paper is a program developed for flexible optimization of stope geometry. Practically any designed stope ranging from drift-like for cut and fill to very expanse for sublevel stoping can be optimized by this program.

Introduction

Stope optimization has been in development to aid in the design of stopes in an underground mining context with the purpose of maximizing economic return. Return from an individual stope is maximized while considering uncertainty and adhering to constraints imposed by mining method, geology and geotechnical analysis. Uncertainty is built into an economic or grade model through the use of geostatistics and this model is used in determining the stope yielding maximum return.

An initial piece of software was developed in 2004 to handle very simple stopes and optimize with a quasi simulated annealing algorithm. Only limited constraints were incorporated into the optimization; however, the program did provide a good starting point. A more advanced program is currently in development that has several advantages over the previous version including the ability to handle more complex geometry, more constraints, and more accurate intersection with the underlying economic block model. This new program was tested in its current state as of June 9, 2006 and will be discussed.

Background

Previous work done regarding stope optimization was quite simple and can provide a good summary of the process involved. The initial version was designed for mining of steeply dipping ore bodies using some form of sublevel stoping technique like vertical crater retreat or Alimak mining. With this type of mining, a stope can in most practical cases be defined using eight vertices or corner points, see Figure 1. Constraints included maximum allowable span, minimum required mining width and minimum dip. Since only a stope with eight nodes was considered, these were easy parameters to check during optimization.

The process of optimization used was a quasi simulated annealing (SA) technique in that only changes that resulted in increased economic value made to the stopes' geometry were accepted. Sub-optimal changes were not accepted based on a probability measure as in classical SA. Each step of the following optimization procedure will be discussed, outlining advantages and limitations. Steps required prior to optimization involved building an economic model and designing an initial stope to be optimized.

1. Read in the initial stope and economic block model.
2. Determine which blocks are inside the stope and sum them to acquire the initial economic value.
3. Randomly choose one of the eight vertices and move it a distance *d* in direction **u**, which is a Cartesian vector.
4. Check for violation of constraints and if so, undo the change and return to step 3.
5. Calculate the stopes new value. If an improvement was made, keep it, otherwise undo the change and return to step 3.
6. Return to step 3 until no change improves the value.
7. Decrease *d* by half and return to step 3 until *d* is sufficiently close to zero.

Due to its simplicity, the big advantage of this process was speed. Optimization of one stope took only seconds. Another interesting feature is the method used to determine if blocks were inside the stope. Since only eight vertices were used, the stope could be effectively defined as a simple finite element. Intersection points with a line going through the centre of a block and along one of the major axis directions with the stope solid could be easily calculated. If the block was in-between the two intersection points, it was inside.

Limitations of this program, however, are numerous. Only stopes with eight corner points can be optimized. The direct **u** was limited to one of the major axis directions. These were x or y, but not both simultaneously. Use of the finite element method also meant that the stope vertices must always be lined up with the corner points of the blocks looking in the x or y directions. Otherwise, intersection points would not line up with the centerline described previously. Determining if blocks were inside was based on block centers; no partial blocks were considered. Over the entire surface area of a stope, this could make a significant difference in the outcome.

Other disadvantages involve constraints. The minimum mining width constraint was applied to the top and base of the stope. This was based on the assumption that development for accessing and mining the stope had to be the same width as the stope itself. It is possible that the minimum required mining width be different than the minimum thickness of a stope, see Figure 2. Any constraints imposed by neighboring stopes were not considered. One example may involve limiting deviations along strike from one stope to the next for mining and development equipment, see Figure 3.

Advances in Stope Optimization

Taking into account some of the limitations described in the background section has been achieved with a new stope optimization program. This program is not complete; however some of the problems with the previous version have been solved. Requirements to complete the

program will be discussed in the future work section. Advances made to date include the following:

1. More complex stopes can be read in and optimized. A stope can be any triangulated solid with any number of vertices.
2. An option to use partial blocks when determining which blocks are inside the stope.
3. More degrees of freedom for stope position and shifting of vertices. The user can choose from seven options: move in x, y, or z only; move in the xy, xz, or yz planes only; or move in x, y and z simultaneously. Which vertex to move and its movement vector are chosen randomly.
4. Multiple stopes can be read in. This requires further testing however.
5. An option to perform quasi SA as in the previous version or classical simulated annealing to achieve a solution.
6. Addition of more constraints including minimum and maximum facet size, maximum deviation across faces or around corners and minimum and maximum allowable thickness.
7. Graphical output for immediate viewing in VRML (virtual reality modeling language) format. Results from the previous version had to be imported into some other graphical software to visualize results.

These changes have made for a more dynamic program. Having the ability to read in any triangulated solid representing a stope extends the applicability to numerous mining methods, not just sublevel stoping. Stopes can range from simple eight node solids to those for cut and fill mining where the stope is essentially a drift. Use of partial blocks is important especially with more complex stope shapes to optimize or when the block size is large relative to the stope. Partial blocks are calculated by refining grid cells that are intersecting the stope faces based on a simple collision detection calculation.

Shifting vertices to find the optimal geometry is not very realistic when movement can only take place in one direction. Addition of more degrees of freedom for moving vertices will result in solutions that exist in more than one direction. The reason for giving the option of using classical simulated annealing comes from increasing the degrees of freedom for optimization. With more complex geometry and vertex movement it is more likely that using quasi SA will result in a stope being augmented into a local minimum or suboptimal solution. Classical SA will more often go beyond these local minima.

Some constraints were incorporated, but more are still required. Min and max facet size constraints help prevent a face from becoming too small or large. These also contribute to meeting minimum thickness and maximum span constraints. Deviations across faces and around corners (

Figure 4: **Deviation across a face and around a corner.**

Figure 5: Unacceptable deviation around corners (left) and across faces (right).

) are based on limitations that may be imposed by drilling equipment. If blastholes are drilled down dip, a highly deviated stope in that direction is just not practical. Also, corners with interior angles of 60 degrees for example may be unmineable, see Figure 5. The minimum mining width constraint was simply converted into minimum thickness.

Program and Parameters

The updated program for stope optimization is like any other GSLIB style program. A parameter file is required and the program is executed from a command prompt. General operation involves reading in a block model and initial stope design, optimizing the stope while outputting progress and output of the final stope design along with a VRML file for visualization. Parameters are shown below:

Line	Parameter	Description
1	GENERAL	
2	8node_stope.plg	-file with stope(s) in plg format
3	sgsim.out	-file with block model
4	1	- model type
5	1	- column(s) for value or grade(s)
6	40 0.5 1.0	-grid definition: nx, xmin, xsize
7	40 0.5 1.0	- ny, ymin, ysize
8	40 0.5 1.0	- nz, zmin, zsize
9	69069	-random number seed
10	results.out	-file for optimization progress
11		
12	CONSTRAINTS	
13	1.0	-annealing stopping tempurature, -1.0 = no optimization
14	1	-block/stope intersection accuracy
15	2.0	-minimum mining width
16	40.0	-maximum allowable span
17	1.0 10.0	-min/max facet side-length
18	50.0	-max deviation across faces (0 to max degrees)
19	80.0	-min deviation around corners (min to 90 degrees)
20	0.5 0.1	-initial vertex movement variance, minimum variance
21	0	-input vertex movement type from file, 1 = yes
22	constraints.txt	-file with constraints
23	1 2	- column for vertex id and movement parameter
24	8	-if not from file, number of vertices to consider
25	1 4	- vertex id, movement parameter
26	2 4	
27	3 4	
28	4 4	
29	5 4	
30	6 7	
31	7 7	
32	8 4	
33		
34	IMAGING (use -1.0 to use defaults)	
35	8node_stope.wrl	-File for output VRML
36	1 0 1	-Draw: Initial stope; final stope; blocks: 1 = yes
37	0.0	-vertex radius, 0=not shown
38	1.0 0.0 0.0	- RGB color
39	0.5 0.5 0.0	-RGB line color
40	0.5 0.5 0.0	-RGB face color

41 0.0 0.5 -Face shininess and transparency

Lines 1, 12 and 34 are just parameter section descriptors. The input files containing the stope and block model are on lines 2 and 3. Stope files are in PLG format, which is an older graphics file format, but is very easy to understand and parse. Line 4 offers a model type indicator for economic or grade models; however, only economic models are handled at the moment. An update to the program will be parameters such that if a grade model is read in it can be converted to an economic one. Optimization progress is output to the file on line 10 in GSLIB format with each line showing the iteration, annealing temperature, stope value, which vertex was moved and its movement vector.

Line 13 indicates stopping criteria for simulated annealing. Block model – stope intersection accuracy on line 14 is the number of times a boundary block will be refined for percentage inside calculation (if set to 1, a block is refined into 8, if 2 it is refined into 16 and so on). This is one area of the program that could be improved to reduce runtime. Lines 15 to 19 are the possible constraints for the overall stope. Initial movement variance and minimum variance on line 20 are parameters for choosing random vertex motion vectors. These vectors are scaled off a normal distribution and as progress made in terms of increasing stope value decreases, so is the variance. Allowable movement to individual vertices is described either from a file on line 22 or from manual input into the parameter file on lines 24 to 32 (for the case of 8 vertices). Recall that the movement indicators can range from 0 for fixed position to 7 for full freedom, see Table 1.

Table 1: Vertex movement parameters and description.

Parameter Value	Movement	Parameter Value	Movement
0	None	4	XY Plane
1	X Only	5	XZ Plane
2	Y Only	6	YZ Plane
3	Z Only	7	XYZ Space

The imaging section contains all parameters to output results in VRML format. Parameters provide options to plot initial and/or final stopes along with the clipped block model. Other options are for coloring vertices, lines and facets and characterizing facet material as well.

Examples

Three examples were run with the new program in its current state. Results encourage further development of this software. Example stopes are shown in Figure 6 and include:

1. A simple stope described by eight vertices. Vertices were permitted to move in the xy plane. Maximum allowable facet size was set to 40 units and the minimum to 3. Minimum thickness was set to 3 units and deviations were set at 20 degrees and 70 degrees for across faces and around corners respectively.
2. A more complex stope with 18 vertices was created. This stope was optimized assuming the sides were fixed due to neighboring stopes. Central vertices were permitted to move freely in all three directions. Central upper and lower vertices were shifted in the xy plane.
3. A cut and fill stope or drift was created and optimized. A total of 32 vertices and 60 facets were used. The width of the drift was limited to 2.75 units and the height was

fixed at 2.5. Access to the drift was also kept constant, but the remaining vertices were moved freely in the xy plane.

Table 2 summarizes the initial and final values achieved for all three examples. Initial stopes were designed over synthetic data. The block model was kept small with cube shaped cells, 40 in each direction. Practically no thought went into designing the eight node stope; however, the 18 node stope was designed using three plan views and cross sections. The drift style stope was designed on one level and then just extruded to a height of 2.5 units. Optimization results show the effect of time put into initial designs.

Table 2: Optimization results for three examples

Example	Partial Blocks	Initial Value	Final Value	% Gain
1	No	199,827	566,032	183.0
2	No	271,898	314,002	15.5
	Yes	267,718	306,340	14.4
3	No	38,104	52,288	37.2
	Yes	31,648	41,963	32.6

For the first example, optimization was done with full blocks only. Based on results, the optimal stope went close to limits in facet size along strike and reduced in thickness. The resulting stope would not have been possible with the first program for stope optimization as was the purpose of this first example, see Figure 7. Optimization of the second stope was carried out using full blocks and partial blocks for comparison. It seems that using partial blocks resulted in a lower value. This can be explained by the fact that more costly blocks are being discounted when up to 50% of the block could actually be inside the stope, see Figure 8. Using partial blocks gives a better solution in terms of what will actual be mined. It should be noted that run-time for partial blocks was substantially longer than for full blocks.

Optimization of the drift-like stope was done using both partial and full blocks as well. Because of the drifts size relative to the blocks, using partial blocks is important. Mining using drifts is also a much more selective method so the actual mined drift can be very close to the design. Using too large of blocks becomes impractical with this level of selectivity. Results improved the stope by a substantial amount with very little change in the design, see Figure 9. A limitation of the program that became apparent with this example is the need to move vertices in tandem to overcome constraints. Over 1,800 iterations were carried out during optimization, 45 of which resulted in improvements. Many of the others were rejected because a constraint was violated. In this case, the minimum and maximum thickness constraints prevented the drift from deviating very far from its initial position. If sets of vertices forming a cross section through the drift could be moved in tandem, the drift could have deviated much further, see Figure 10.

Conclusions and Future Work

Using up-to-date computational geometry algorithms and simulated annealing permits optimization of very simple to very complex underground mining stopes. Being optimized over an expected profit model also allows uncertainty in the model to be quantified by the stope. That is, optimized stopes are near “best” in terms of how much we know about a particular region underground. As with any program, constraints and parameters could be added indefinitely to cover numerous possibilities regarding underground mining methods and procedures; however, what was presented offers enough flexibility for a wide variety of cases.

Without making too many additions to the program, some that would be interesting and beneficial to incorporate will be discusses. To be sufficient, more constraints and options require implementation especially in the area of multiple or joint stope optimization. By incorporating these and possibly other parameters to the stope optimization program, it will be able to handle significantly more optimization problems than in its current state.

- Minimum permitted dip to prevent ore hang-ups. This is currently controlled by the angle tolerance around corners, which are differentiated in the program by initial angle values. Top-to-face and bottom-to-face are considered corners and in most cases, the dip could be as low as 50 degrees. Differentiation of corners along sides from corners along top and base should be incorporated.
- Stability constraints regarding overall stope dimensions, perhaps similar to the Matthew Stability Graph method for analyzing underground openings.
- More efficient calculation of partial block intersections with the stope solid. The examples indicated that run-time was much worse for partial blocks.
- An option to link together sets of vertices that are to be moved in tandem. This could be useful if the cross section of a stope is to remain constant (when the stope is actually a drift as in some cut and fill operations).
- Expansion of the vertex movement options to consider movement in a specified direction only, perhaps perpendicular to strike or mining direction (these would have to be specified as parameters).
- Use of other block models such as geology or rock quality models for constraints that may be rock-type dependent.
- Parameters for reading in multiple stopes and accepting various constraints specific to joint stope optimization.
- More economic parameters. Currently, an economic model is read in and used as the only control for determining the optimal stope. There are other costs that depend on the size and configuration of a stope. These costs cannot be incorporated into the block model, but should be available for optimization. An example of one type of cost is rock support, which is dependent on the surface area where support may be required.
- Create output that is compliant with some major pieces of modeling software including Vulcan.

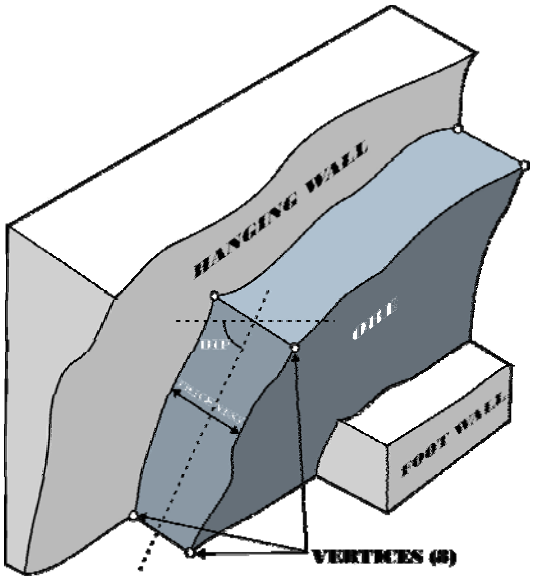

Figure 1: Stope described by eight vertices.

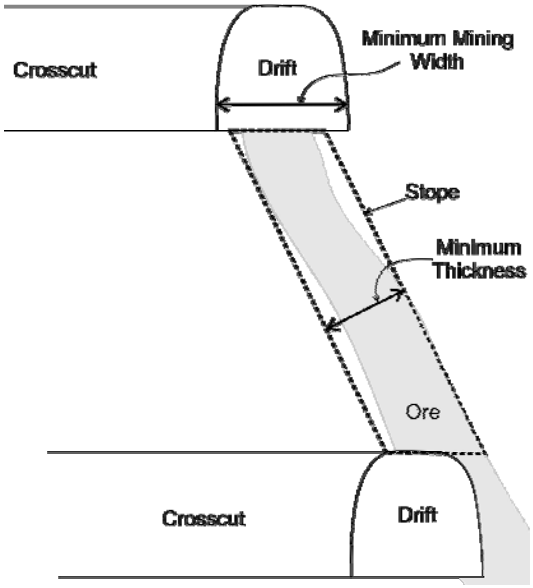

Figure 2: Difference between minimum required mining width and minimum stope thickness.

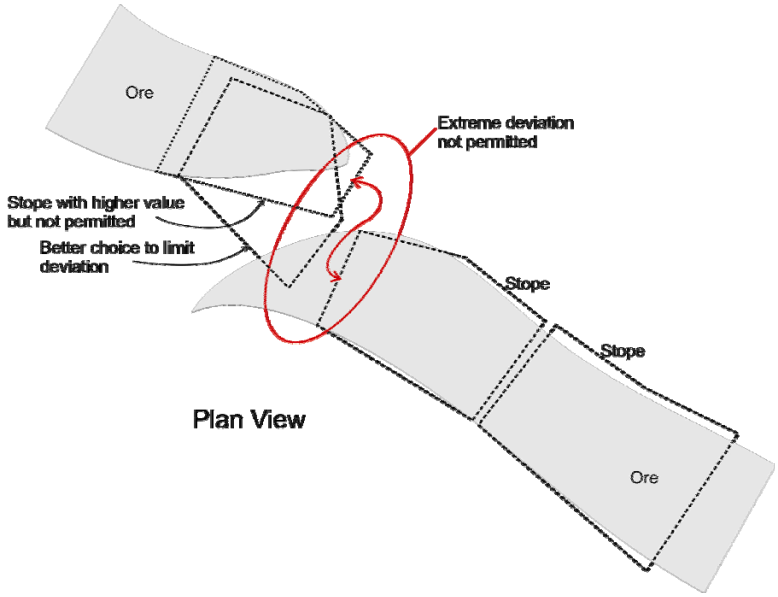

Figure 3: Unacceptable deviation from one stope to the next along strike.

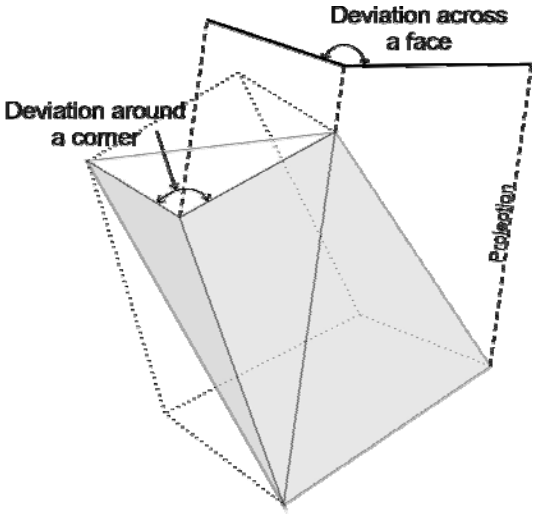

Figure 4: Deviation across a face and around a corner.

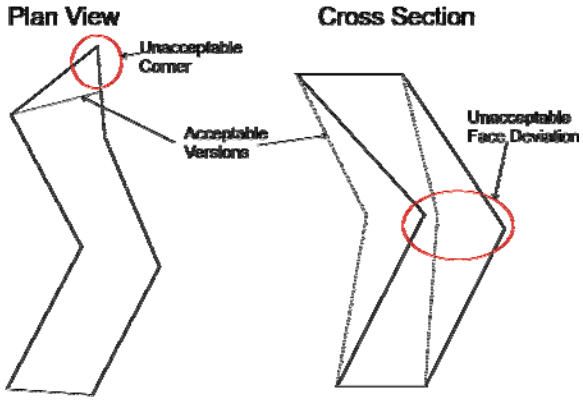

Figure 5: Unacceptable deviation around corners (left) and across faces (right).

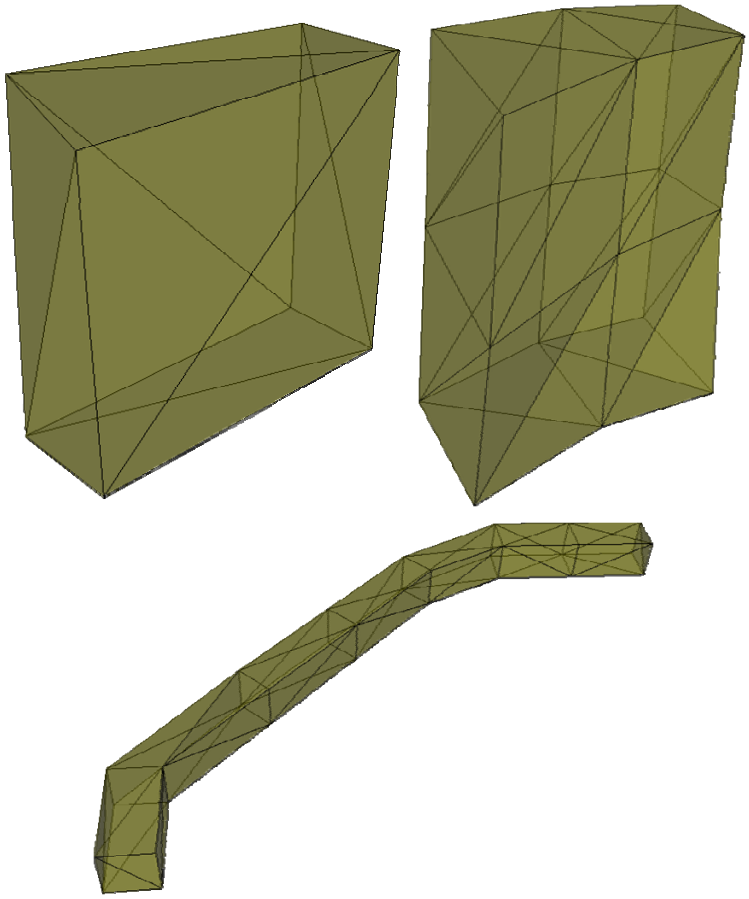

Figure 6: Eight-vertex stope (upper left), 18-vertex stope (upper right) and a drift style stope (bottom).

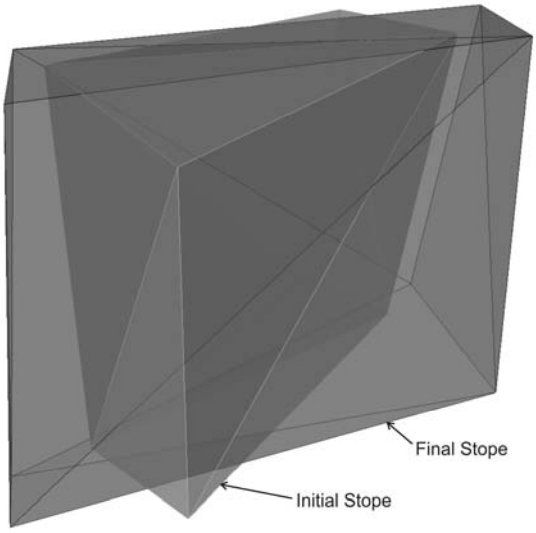

Figure 7: Initial and final stopes for the eight vertex stope optimization.

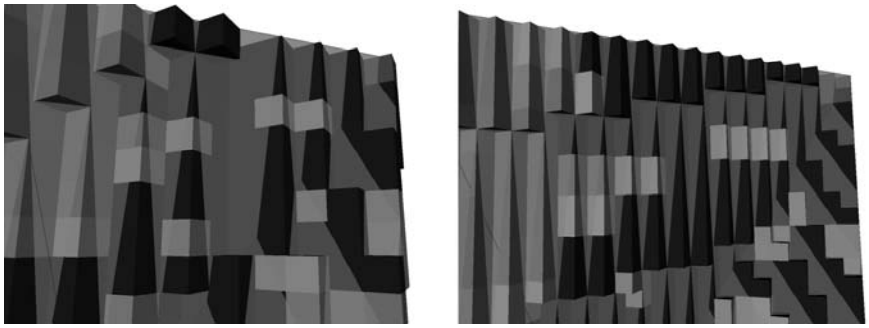

Figure 8: Full block (left) and partial blocks (right) intersected with a stope.

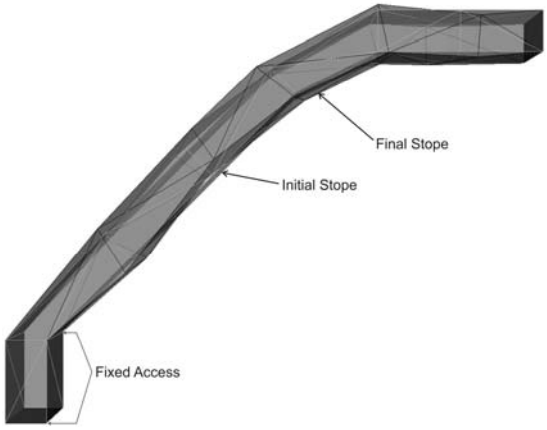

Figure 9: Initial and optimized drift style stope.

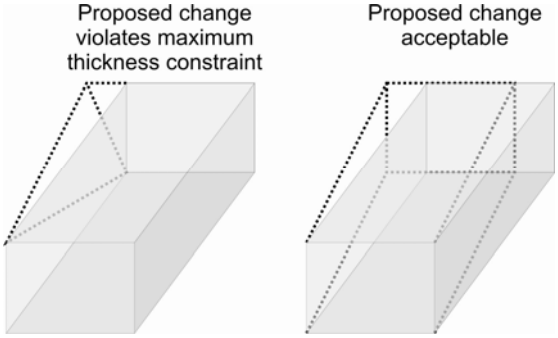

Figure 10: Moving points individually (left) versus moving them in tandem (right).