STRATEGIC MINE PLANNING, THE BZ ALGORITHM AND BEYOND

Authors: S. Hill, V. Bai, I. Rudge, D. Place and C. Poblete

DASSAULT SYSTEMES

INTRODUCTION

The strategic mine planning problem for open pit mines consists of determining the extraction sequence and utilization of blocks of earth from a mining deposit to maximize an objective. The mining deposit is represented as a 3D block model where attributes are assigned to each block such as grade, tonnes, emissions from removal and profit. Precedence relationships exist between blocks to capture wall slope constraints. The block model is projected over a number of periods to capture time related costs and constraints such as mining capacity per period. The objective of the problem is to maximize the sum of the discounted cash flow that comes from excavating and processing each of the blocks subject to the constraint set.

Strategic mine planning has the greatest impact on the overall profitability of a mining operation [10]. In addition, one of the most effective ways to reduce a mine's environmental and sustainability costs is when optimizing its strategic mine plan.

To solve this complex problem, a mining engineering team will use their skills and experience along with powerful mine planning optimization tools to construct a solution to this problem.

Two key features of such tools that are highly desirable are:

- Solutions are produced that are optimal / provably near optimal
- Quick execution times

Producing good solutions quickly allows a mining engineer to explore many different scenarios and sequences of work, leading to better overall results and reduced risk.

The underlying scheduling problem to be solved is large and complex. Block models can consist of millions of blocks, each with tens or hundreds of precedence relationships to specify wall slope requirements. Other constraints such as mining capacity, minimum equipment utilization and grade constraints per period, are also present.

Traditional approaches to solving these problems have utilized solutions to the Ultimate Pit Limit problem (UPIT), solved using the Lerchs-Grossmann [13] or Pseudoflow algorithm [9]. UPIT problems only consider the precedence relationship between blocks, where the time value of money and other constraints are not directly considered. Other factors are later iteratively added using heuristics. Note that these heuristics do not guarantee an optimal solution is found nor provide an estimate of the quality of the solution.

Better solution methods that consider the problem holistically promise better quality solutions for the mining engineering team to use. Complex constraints such as ore blending specification can be explicitly included, which often are not handled well using approximation techniques.

Life of mine operations can be modelled using Mixed Integer Linear Programming (MILP). The core concept is to assign a [0,1] variable to each block per period and destination then formulate the problem using the time and space properties of the block set.

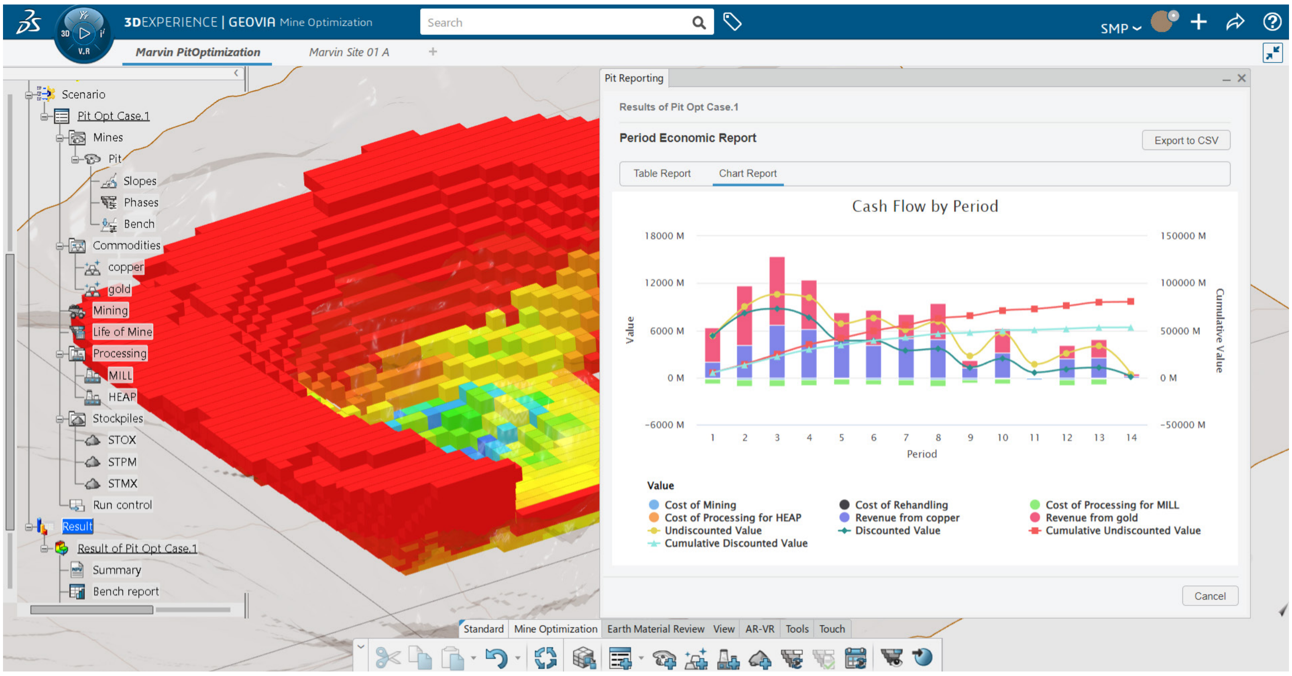
There are many advantages to using this modelling paradigm:

- The modelling is holistic.
- Problems can be solved to optimality / provably near optimality using rigorous mathematical methods.
- MILP is a global optimization technique. Global optimization techniques can increase profit by more than 10% compared to heuristics, as we shall show in the included project study.
- Global optimization techniques often produce a solution to complex problems where simple approximation techniques fail.

Until recently, the size of problems that can be tackled has been limited by the power of commercial MILP solution packages. A recent advance allowing a step change in the size and/or solution time, is the Bienstock-Zuckerberg (BZ) algorithm. The BZ algorithm does not solve the MILP directly, but solves the easier linear programming (LP) problem. LP solutions make an excellent starting point for approximation techniques or for use in solving the overall MILP.

To date, the BZ algorithm has not been easy to understand without reading mathematical optimization papers and presentations. In this paper, we provide an easy to understand description of the algorithm, along with our work in this area.

The GEOVIA Research & Development team has created a new mine planning optimization engine, GEOVIA Mine Maximizer (GMX), that utilizes the BZ algorithm. GMX is the solution engine provided exclusively with the GEOVIA **Strategic Mine Planner** and the GEOVIA **Pit Optimizer** roles available on the **3D**EXPERIENCE platform. Since its introduction in 2010, the BZ algorithm has been significantly improved using a number of speed-up techniques. We also provide easy to understand details of the speed-ups used in GMX, coming from our and other authors' work.



Picture 1: Snapshot of the Mine Optimization App included in the Strategic Mine Planner and the Pit Optimizer roles

One of our main goals of this paper is to demonstrate the benefits of this approach compared to formulating and solving the MILP using a commercial package. These benefits include:

- Problems can be solved in orders of magnitude quicker time.
- Problems solved do not need approximating (such as solving problems period by period or using a sliding time window) but can be tackled holistically.
- Because the solver is more powerful, much larger problems can be tackled.

We provide results for the standard Minelib literature problem set [7] and a comparison with the Prober B engine used in the GEOVIA Simultaneous Optimizer role within Whittle, demonstrating the quality of our work. To conclude the paper, we highlight how this work fits within the GEOVIA mine planning tool set, along with a discussion of the future direction of our work.

In the following sections, a block is considered as a single scheduling unit with precedence relationships existing between blocks to capture geotechnical constraints. Also, mining benches and panels can be pre-designed by engineers, which enforce that one grouping of blocks needs to be extracted completely before the following grouping of blocks can be started. We model both cases as bins to be mined, where bins are part of a grouping referred to as a cluster. Block problems can be considered as cluster problems having one bin, so we do not differentiate between these cases in our description of the work, having created one solution to tackle all problems.

LIMITATIONS OF GENERAL MILP APPROACHES TO STRATEGIC MINE PLANNING

The main difficulty in solving a strategic mine planning problem is the problem's size. The number of variables of the problem is proportional to the number of Bins x the number of Periods x the number of Destinations. Wall slope constraints that must hold per period also number in the millions. Except for smaller sized problems, such formulations are intractable for commercial MILP packages. Often formulations cannot be loaded due to memory restrictions, and solve times become prohibitive [15].

The problem can be reduced so that it can be solved using a commercial package. Blocks may be aggregated, or the problem may be solved on a period by period basis or by using a sliding time window. While taking these steps may mean that solutions can be generated, too much aggregation can lead to a 20-30% drop in Net Present Value (NPV) [19]. Only considering a small set of the periods in an iterative process is myopic where what is scheduled earlier does not take into account later factors. This can lead to additional reductions in NPV or no solution being found for complex problems.

Ideally, we would like to minimize the use of these reduction techniques, but how can these larger problems be solved? Note that commercial MILP packages have no knowledge of the underlying problem. Whether it is nurse roster scheduling, ship scheduling or strategic mine planning, all these problems are approached using a similar solution strategy. The effectiveness of the BZ algorithm comes from recognizing the specific strategic mine planning problem structure and using this in the algorithm's development.

THE BZ ALGORITHM

Originally proposed by Beinstock and Zuckerberg in 2009-2010, the BZ algorithm is a Lagrangian decomposition technique for solving an easier Linear Programming (LP) problem, an essential step for solving the MILP.

The algorithm efficiently takes advantage of the mining problem structure by breaking the formulation's constraint set into two parts:

- The precedence constraint set
- A smaller set of hard constraints

Precedence constraints are those that require one thing to happen before another. They consist of wall slope constraints per period and constraints saying that if something has happened by a certain period, it has also happened by the next time period.

Hard constraints are the more general mining constraints. They include constraints such as the maximum amount of ore processed in a period, the maximum number of tonnes extracted and the minimum grade per period.

The algorithm is summarized in Diagram 1.

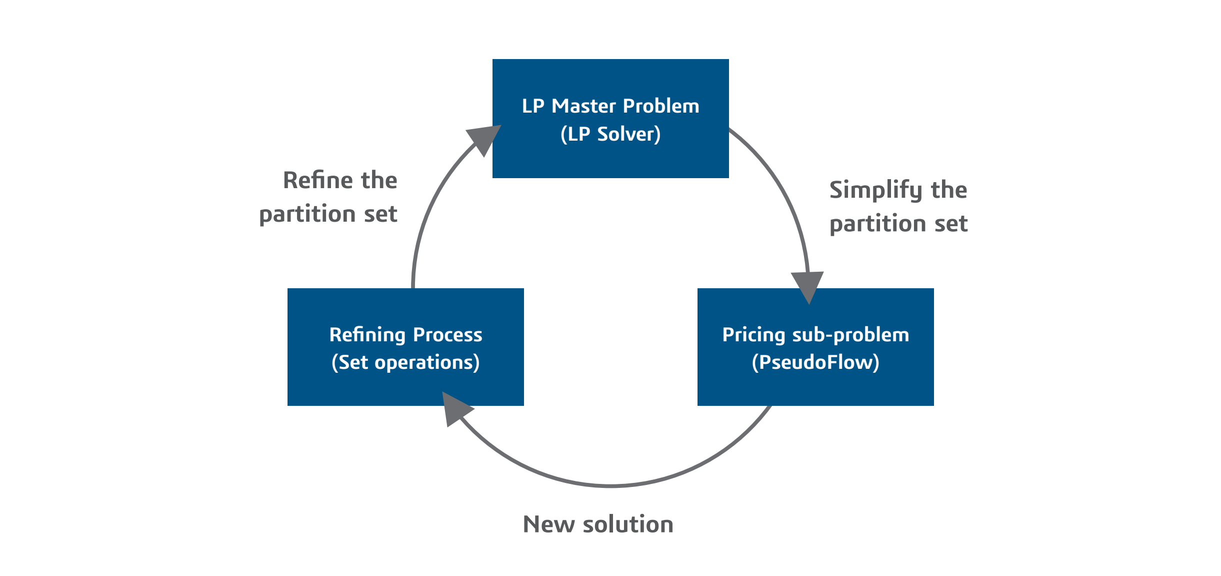


Diagram 1: Algorithm Summary Schema

The following observations are relevant for understanding the algorithm:

Pricing sub-problem (Pseudoflow)

- A problem that only has precedence constraints can be efficiently solved as a network flow problem using the Pseudoflow algorithm [9].
- If the hard constraints are included in the objective using Lagrangian relaxation then this problem (the Pricing sub-problem) yields an upper bound on the solution objective.

LP Master problem

- Problems can be formulated using partitions of variables rather than the individual variables themselves. Variables are only ever allocated to one partition. The allocation must be such that the partitions yield a feasible solution to the problem. Artificial variables are used if an initial feasible allocation is not known
- The solution of a linear program (LP Master problem) in terms of partitions yields the Lagrangian multipliers for use in the next Pricing sub-problem.
- The Partitions of variables in the current iteration contain the partitions of the previous iteration meaning the LP Master Problem objective function value is nondecreasing.
- The LP Master problem yields a lower bound on the solution objective.

Refining Process (Set operations)

- The solution from the Pricing sub-problem is used to create the next partition set, summarized in Diagram 2.
- The partition set is periodically simplified by merging partitions that had the same solution value in the previous iteration. This keeps the number of partitions small.

The algorithm proceeds until the LP Master problem and Pricing sub-problem solutions converge.

We summarized the Refining Process in Diagram 2.

The original partition set is compared to the new Pseudoflow solution from the Pricing Sub-problem to create partition sets that intersect (Intersection), do not intersect (Prior) and a set of variables in the Pseudoflow solution but not in the original partition set (New).

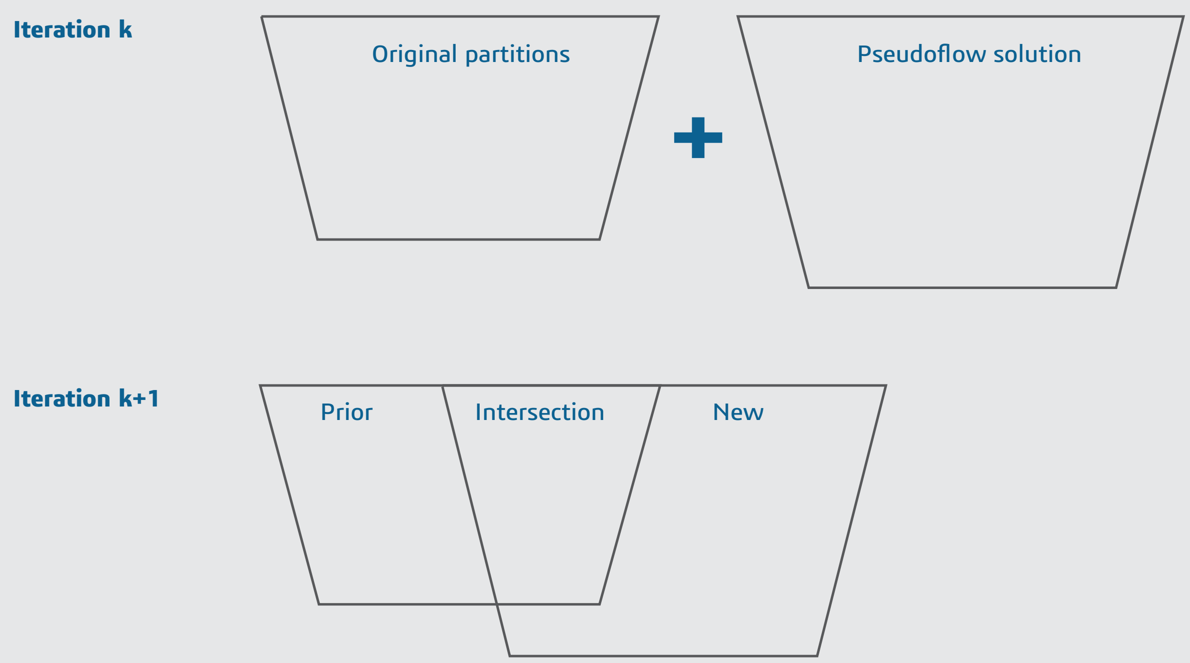


Diagram 2: Refining Process Schema

RUNTIME PROPERTIES

The BZ algorithm efficiently solves the LP formulation of a mine planning problem. To demonstrate this efficiency, we compare results for the literature problem KD, solving the LP using the BZ algorithm and a commercial solver. KD is a copper mine with 14,153 blocks, of which 12,154 appear in the Ultimate Pit to be scheduled over 12 time periods. Each block can go to two destinations, where it is processed as either ore or waste.

A commercial MILP package using the Simplex or Barrier algorithm to solve the LP, solves the LP problem in 4,710 seconds. Using our latest GMX implementation, we solve the LP in 2.6 seconds, an 1,811x speed up!

In the following, we detail methods we used to speed up the original algorithm, including those from the literature along with our own methods.

BZ ALGORITHM SPEED-UPS

For many of the literature speed-up methods, we refer to the paper of Muñoz et al [16]. Our work includes these results along with new work to produce fast results. Overall, we have realized an 87x decrease in total runtime for the nine standard Minelib literature test problems compared to our original implementation of the algorithm from the 2010 paper [2].

Techniques used to speed up the BZ algorithm are summarized below:

Pre-processing

UPIT: We first solve an ultimate pit limit problem, then schedule the solution blocks. This step is proven to not lose the optimality of the solution [3] and often significantly reduces the size of the problem considered.

Waste reduction: It is possible to exclude possible destinations for waste blocks that will never be processed as ore. A block is not processed as ore if no benefit is realized in terms of tonnage and profit achieved.

Early start: The earliest time a cluster can be removed can be determined by considering extraction constraint capacities. We use these early start times to strengthen the problem formulation by removing variables from the LP Master problem and unnecessary nodes from the Pseudoflow problem.

Pricing sub-problem (Pseudoflow)

Most of the work of the BZ algorithm is in solving the Pricing sub-problem which consists of solving a network flow problem using the Pseudoflow algorithm. For test problem KD, solving Pseudoflow problems accounts for 56% of work done in our implementation.

Mineflow: For our purposes, we have created a new fast Pseudoflow solution engine based on the recent work of Deutsch, Dağdelen and Johnson [6]. Mineflow shows speedups over the original Pseudoflow engine by customizing the algorithm specifically for Ultimate Pit Limit problems.

Warm starts: As outlined in Hochbaum [9], the Pseudoflow algorithm can be warm started using a previous solution that is stored as a normalized tree. In our work, we warm start the Pseudoflow algorithm when the solutions from one iteration to the next are expected to be similar. We say that solutions are similar provided the Optimality Gap is below a tolerance.

Path contraction: The Pseudoflow problem to be solved only consists of precedence constraints with hard constraints used to modify the objective from one iteration to the next. For a time period, each destination of a cluster's bin is only dependent on other destinations of the bin, and in fact, they form a chain. Following the work of Muñoz et al [16], for a given set of Langrangian multipliers, these bin destinations per period can be pre-processed to determine the best destination for the bin if it is removed in the period. Then, using the precedence relationships, bin destination nodes can be contracted into a single cluster per period, significantly reducing the size of the Pseudoflow problem to be solved.

LP Master problem

At each iteration of the BZ algorithm, a small LP is created and solved. In our implementation, we use the "By" formulation of the problem, which we have found to be preferable to the "At" one.

LP Creation: We create our LP from the last period to the first and from the last destination of a bin to the first. This reduces work since we know that if a variable is not defined for one instance, then by the chaining property of the precedence constraints, it is not defined for the earlier one. Also, we track constraints that have been added to the LP to avoid duplication.

Warm starts: Due to the way partitions are refined, following the work of Muñoz et al [16], we set initial partition values to be their value in the previous iteration with new partition values set to zero.

Refining Process (set operations)

Initial partitions: A good initial feasible solution can significantly reduce the number of BZ iterations by providing a good initial partition set. We initialize the partition set using a new heuristic, that is included in our scheme to reduce the number of BZ iterations.

Calculation: Each partition is stored as an unsorted list of identifiers. We also store a mapping of identifiers to the partition that it belongs to, which is kept current. Using the Pseudoflow solution along with these data structures allows for efficiently determining partitions from one iteration to the next.

CONVERGENCE

We use a new propriety scheme to significantly reduce the number of BZ iterations. Chart 1 illustrates the effectiveness of this approach via example. For the largest Minelib [7] literature problem, McLaughlin, the number of iterations is 79 using our implementation of the original BZ algorithm, which is reduced to 16 for the new version. Of note is the rate of change of the Optimality Gap from one iteration to the next, highlighting the much faster convergence and so faster runtimes.

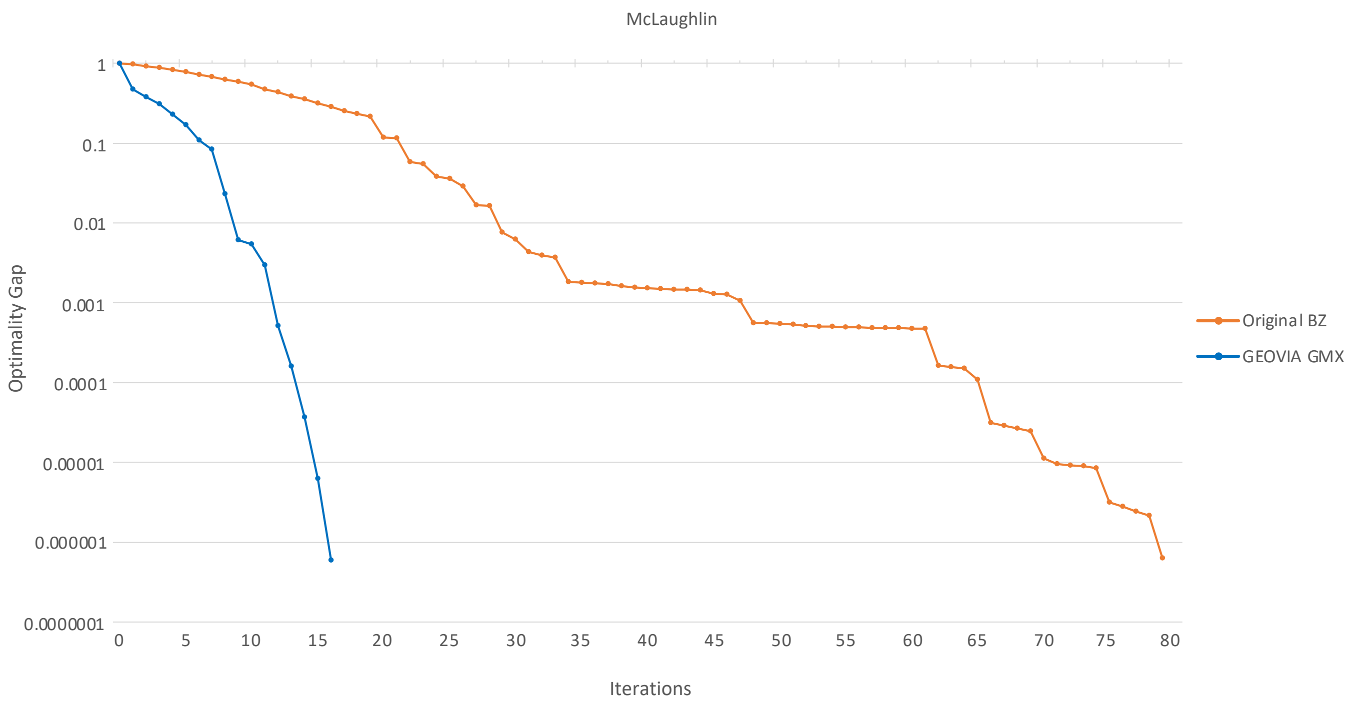


Chart 1: Comparison of convergence of the solution for the McLaughlin Problem

STOCKPILING

Stockpiles have inherent non-linear properties. The grade of ore placed on a stockpile is known, but once a quantity of ore is removed from the stockpile and then other ore is placed on it, the stockpile average grade becomes a non- linear function. While non-linear models can be created to represent stockpiles, these have proven intractable except for small size problems [14].

To model stockpiles using linear constraints, we follow the recent work of Moreno et al [14]. Here, an average grade of ore that is removed per period is specified, with constraints added so that the average grade of ore placed on the stockpile in earlier periods is at least this specified amount. Since it is less profitable to stockpile ore at a higher grade than that removed, in a solution, the two grades often match.

While this modelling approach is often used, we have found the solution procedure of Moreno et al [14] provides excellent results.

COMPUTATIONAL RESULTS

Table 1 reports results for the nine Minelib literature problems [7]. The Minelib problems are Direct Block Scheduling (DBS) problems of various sizes with up to two processing or extraction capacity constraints. All test runs were completed on a standard Lenovo ThinkPad P53 laptop with 9th Gen i7 CPU, 6 cores, running at 2.6 Ghz with 32 GB memory. A lower bound comes from solving the LP Master problem, an upper bound from solving the Pricing sub-problem. If the Optimality Gap (1.0-lower bound/upper bound) is at most 10^{-6} then we say the BZ algorithm has converged. The number of iterations and the time to do this work (in seconds) are reported in the last two columns.

Once the LP has been solved, we run ten variants of the TopoSort heuristic as well as the Optimize-Destinations heuristic [17] to create near optimal solutions. The total runtime for these heuristics is fast, taking less than a minute to complete on average.

Data	Blocks	Periods	Destinations	Precedence	Upper Bound	Feasible Solution	Gap	Literature Gap	BZ Iterations	BZ Runtime (s)
marvin	53,271	20	2	650,631	911,481,083	902,165,755	1.02%	0.79%	17	4.7
kd	14,153	12	2	219,778	410,891,357	409,983,459	0.22%	0.38%	12	2.6
mcLaughlin	2,140,342	20	2	73,143,770	1,512,972,410	1,511,711,920	0.08%	0.07%	16	175.2
mcLaugh-lin_limit	112,687	15	2	3,035,483	1,324,830,265	1,323,162,078	0.13%	0.24%	15	66.7
newman	1,060	6	2	3,922	24,308,812	23,836,969	1.94%	1.27%	8	0.1
sm2	99,014	30	2	96,642	1,652,395,004	1,650,615,684	0.11%	0.09%	17	11.5
zuck_large	96,821	30	2	1,053,105	57,938,839	57,735,892	0.35%	1.04%	12	73.6
zuck_me-dium	29,277	15	2	1,271,207	748,151,214	722,733,896	3.40%	3.00%	15	19.1
zuck_small	9,400	20	2	145,640	905,544,538	894,913,722	1.17%	0.07%	20	6.9

Table 1: GMX results for Minelib problems

Comparing the Optimality Gap (Gap) to the best literature result (Literature Gap), we note that best/near best solutions are produced in each case. Also, the maximum runtime to produce a solution is under 5 and a half minutes. Lastly, our results compare favorable to those of specialized meta-heuristics, which can take many hours to run.

PROJECT STUDY

To provide a fair evaluation of the GMX engine's effectiveness, it is beneficial to compare it with a well-established commercial tool. Many existing studies draw comparisons with GEOVIA Whittle, primarily focusing on the Milawa engine. Milawa is a widely used optimizer specifically designed to optimize the scheduling of panels (pushback/bench-level problems). However, Prober B is a better optimization engine offered within Whittle as part of the Simultaneous Optimizer (SIMO) role. This solver schedules at the panel level with block aggregation by grade bin and offers a broader optimization range than Milawa, tackling cut-off grades, stockpiles, and blending concurrently.

The 2016 paper "Advanced SIMO vs Milawa and SPCO", authored by G. Whittle [18], supports the effectiveness of the Prober B engine in SIMO. The paper compares Prober B to other Whittle scheduling engines. In a study referenced in the paper, Prober B realized an increase of 41.4% in NPV compared to a 28.4% increase using Milawa and SPCO.

We compare the results of the GMX solution engine with Prober B (SIMO) for a project study of a gold and silver mine. The mine consists of two pits with 24,331 bins and is to be mined over 11 years. A three phase case and a seven phase case consisting of 49 and 119 panels respectively, are provided. For the seven phase case, stockpiling is also explored.

GMX can use the original number of bins for these cases whereas Prober B requires further bin aggregation to avoid longer runtimes. Prober B also requires configuring a number of parameters to get the best results.

For these three cases, GMX realizes a 3.1% to 15.7% increase in NPV compared to results that can be gained from Prober B by an experienced user.

GMX is also considerably more efficient than Prober B. In testing these cases and an additional 20 project studies, we have seen 22x faster runtimes on average. This runtime speed allows the consideration of many different scenarios by the engineering team as the strategic mine plan is created.

MULTIDIMENSIONAL SCENARIO DESIGN

GEOVIA has recently implemented several solutions to improve open pit strategic mine planning workflow. GMX is the solution engine provided with our Strategic Mine Planner and Pit Optimizer roles, available on the **3D**EXPERIENCE platform. Besides being a single source of truth, the **3D**EXPERIENCE ecosystem provides a range of powerful tools, including design exploration, parameter analysis and multi-discipline engineering optimizations.

Multidimensional scenario design can be performed using the **3D**Experience toolkit. Experiments that use different data sources and varied parameter values can be created in a single dashboard, allowing the exploration of many different financial and technical parameters.

The following questions can quickly be addressed:

- How is the design effected if the NPV discount rate is increased from 10 to 12%?
- What if trucking capacity is increased in year five?
- What if carbon emissions are reduced by 10% in later years?

Performing this analysis allows the mining engineer to explore many different cases that they otherwise would not have time to do. Multidimensional scenario design reduces project risk, increases efficiencies, and yields higher NPV profits. For project work, we have seen 20-30% increases in NPV profit in some cases.

CONCLUSION

In this paper, we have provided an easy to understand description of the BZ algorithm and ways that it has been sped up. This work is implemented in the new GMX solution engine, exclusively available in the GEOVIA **Pit Optimizer** and the GEOVIA **Strategic Mine Planner** roles on the **3D**EXPERIENCE platform.

We provided a comparison of runtimes for GMX and a commercial LP solution package, showing that GMX is 1811x faster for the literature problem KD. We also demonstrated fast runtimes for the nine literature Minelib problems [7], producing the best/near best solutions in each case. Lastly, we compared the solutions from GMX and Prober B (Whittle), realizing a 3.1 to 15.7% improvement for customer study cases, along with solution runtime benefits. In summary, these results demonstrate the power and benefit of the new GMX solution engine compared to the approach of formulating and solving a MILP using a commercial package, as well as the direct application of heuristics.

The BZ algorithm is an important milestone in solving mine planning problems. However the work does not finish here. Recall that the BZ algorithm does not solve mine planning problems, but rather provides an efficient method to solve the linear programming (LP) formulation of the problem. To further our work, we wish to use the BZ algorithm as an efficient subroutine to solve the more difficult MILP problem.

As shown by our experiments with the KD test problem, using standard Simplex or Barrier methods to solve mine planning programs becomes a significant bottleneck as the problem size increases. Therefore, rather than using the BZ algorithm in a pre-processing step to solve a smaller MILP problem, our future work involves investigating a customized MILP solution method allowing the use of the BZ algorithm at its core. Further, using parallelization will significantly reduce the time to produce optimal/near optimal solutions for large complex problems, something mining engineers want.

REFERENCES

[1] Bienstock D, Zuckerberg M (2009) A new LP algorithm for precedence constrained production scheduling. Working paper, Columbia University, New York.

[2] Bienstock D and Zuckerberg M (2010) Solving LP Relaxations of Large-Scale Precedence Constrained Problems. Integer Programming and Combinatorial Optimization, in Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Jun. 2010, pp. 1–14. doi: 10.1007/978-3-642-13036-6_1.

[3] Caccetta L, Hill SP (2003) An application of branch and cut to open pit mine scheduling. J. Global Optim. 27(2–3):349–365.

[4] Chicoisne R, Espinoza D, M. Goycoolea, E. Moreno, and E. Rubio (2012) A New Algorithm for the Open-Pit Mine Production Scheduling Problem. Oper. Res., vol. 60 (3):517–528.

[5] Dassault Systèmes (2019) GEOVIA Whittle. https://www.3ds.com/products-services/geovia/products/whittle/.

[6] Deutsch M, Dağdelen K, and Johnson T. (2022) An Open-Source Program for Efficiently Computing Ultimate Pit Limits: MineFlow. pp. 1–13, doi: 10.1007/s11053-022-10035-w.

[7] Espinoza D, Goycoolea M, Moreno E, Newman AM (2012) Minelib: A library of open pit mining problems. Ann. Oper. Res. 206(1): 1–22

[8] Gamache M, Brika Z and Dimitrakopoulos R (2018) Multi-product mine scheduling optimization under multi-element geological uncertainty. Business.

[9] Hochbaum DS (2008) The pseudoflow algorithm: A new algorithm for the maximum-flow problem. Oper. Res. 56(4):992–1009

[10] Hustrulid W, Kuchta K, eds. (2006) Open Pit Mine Planning and Design (Taylor and Francis, London).

[11] Jélvez E, Moralesa N, Nancel-Penarda, Cornillierc F (2019) A new hybrid heuristic algorithm for the Precedence Constrained Production Scheduling Problem: A mining application. Omega, Elsevier, vol 94(C).

[12] Kenny, A., Li, X., Ernst, A., and Thiruvady, D. (2017). Towards solving large-scale precedence constrained production scheduling problems in mining. In Proceedings of the Genetic and Evolutionary Computation Conference, pages 1137–1144. ACM

[13] Lerchs H, Grossmann IF (1965) Optimum design of open-pit mines. Trans. Canadian Mining Inst. 58(1):47–54

[14] Moreno E, Rezakhah M, Newman A, Ferreira F (2017) Linear models for stockpiling in open-pit mine production scheduling problems. Eur. J. Oper. Res. 260(1):212–221.

[15] Muñoz G (2018) Mixed-Integer Programming Techniques for Strategic Open-Pit Mine Planning, McGill University. https://cosmo.mcgill.ca/wp-content/uploads/2015/06/Gonzalo-Mu%C3%B1oz-Mixed-Integer-Programming-Techniques-for-Strategic-Open-Pit-Mine-Planning.pdf

[16] Muñoz G, Espinoza D, Goycoolea M, Moreno E, Queyranne M, Rivera O (2018) A study of the Bienstock-Zuckerberg algorithm, Applications in mining and resource constrained project scheduling. Comput. Optim. Appl. 69(2):501–534.

[17] Rivera Letelier O, Espinoza D, Goycoolea M, Moreno E, and Muñoz G (2020) Production Scheduling for Strategic Open Pit Mine Planning: A Mixed-Integer Programming Approach. Operations Research, 68(5): 1425–1444.

[18] Place D, Whittle G and Baxter N (2016): www.3ds.com/fileadmin/PRODUCTS-SERVICES/GEOVIA/PDF/whitepaper/GEOVIA-Whittle-SIMO-WhitePaper.pdf

[19] https://www.linkedin.com/pulse/effect-reblocking-pit-optimization-yaohong-d-jiang/

ABOUT THE AUTHORS

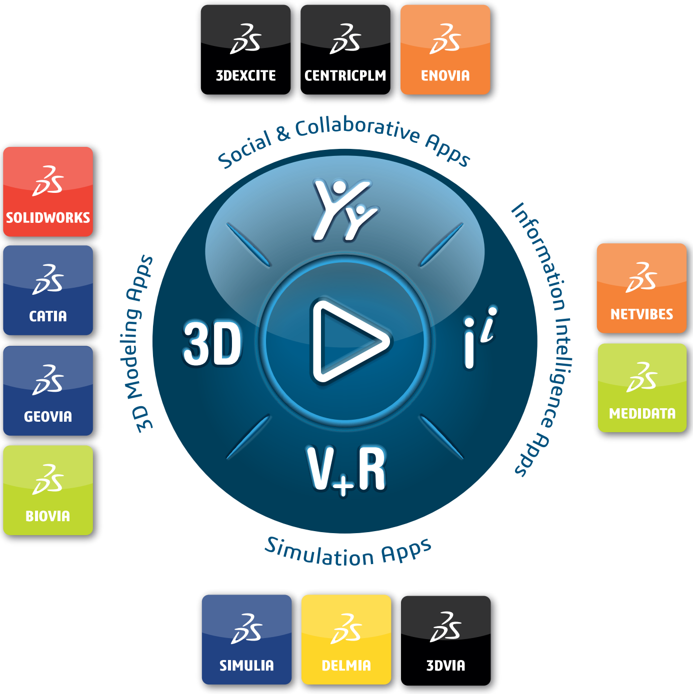
Stephen Hill
Ph.D Operations Research, Mine Planning and Optimization Expert, Dassault Systèmes, GEOVIA R&D, Australia. Experience in Mine Optimization since 1998.

Victor Xiaoyu Bai
Ph.D Geostatistics and Mine Optimization, Software Engineering Senior Manager, Dassault Systèmes, GEOVIA R&D, Australia. Experience in Mine Optimization since 2009.

Iain Rudge
B.Eng, BSc. Mathematics, Software Engineering Specialist,Dassault Systèmes, GEOVIA R&D, Australia. Experience in Mine Optimization since 2022.

David Place
Ph.D Earth Sciences, B. Sc. Mathematics & Physics, Software Engineering Director, Dassault Systèmes, GEOVIA R&D, Australia. Experience in Mining Software since 2002.

Cristian Poblete
Mining Engineer, B. Eng. Sc. Mining & Metallurgy, App Portfolio Senior Manager, Dassault Systèmes, GEOVIA R&D, Australia. Experience in Strategic Mine Planning since 2011.



Our **3D**EXPERIENCE® platform powers our brand applications, serving 12 industries, and provides a rich portfolio of industry solution experiences

Dassault Systèmes, the **3D**EXPERIENCE® Company, is a catalyst for human progress. We provide business and people with collaborative virtual environments to imagine sustainable innovations. By creating virtual twin experiences of the real world with our **3D**EXPERIENCE platform and applications, our customers can redefine the creation, production and life-cycle-management processes of their offer and thus have a meaningful impact to make the world more sustainable. The beauty of the Experience Economy is that it is a human-centered economy for the benefit of all –consumers, patients and citizens.

Dassault Systèmes brings value to more than 300,000 customers of all sizes, in all industries, in more than 150 countries. For more information, **visit www.3ds.com**

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

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

DASSAULT SYSTEMES | The **3D**EXPERIENCE® Company