

# 战略矿业规划， BZ 算法及其他

作者：S. Hill、V. Bai、I. Rudge、D. Place 以及 C. Poblete



# 介绍

露天矿的战略矿业规划问题包括确定开采顺序以及对矿床中的岩块加以利用，以实现目标最大化。矿床采用 3D 区块模型进行表达，每个区块都有自己的属性，如品位、吨数、开采排放和利润等。区块之间存在优先级关系，以捕获边坡约束。区块模型被投影到多个时段，以捕获与时间相关的成本和约束，如单位时段的采矿能力等。问题的目标是在约束集内，最大限度地挖掘和处理每个区块产生的折现现金流。

战略矿业规划对采矿作业的整体盈利能力影响最大[10]。此外，优化战略矿业规划是降低采矿环境和可持续发展成本最有效的方法之一。

为了解决这一复杂问题，采矿工程团队将利用他们的技能和经验，以及强大的矿业规划优化工具来构建该问题的解决方案。

这类工具具备两大重要功能：

- 生成最优/可证明接近最优的解决方案
- 快速执行时间

快速生成优质解决方案可允许采矿工程师探索多个不同的场景和工作顺序，从而获得更出色的整体效果并降低风险。

有待解决的基本调度问题是个庞大且复杂的问题。区块模型可能由数百万个区块组成，每个区块都有数十或数百个优先级关系，用于指定边坡要求。此外，还存在其他约束，如采矿能力、最低设备利用率和单位时段的品味约束等。

解决这些问题的传统方法是利用 Lerchs-Grossmann 算法[13]或 Pseudoflow 算法[9]解决最终矿井极限问题（UPIT）。UPIT 问题只考虑区块之间的优先级关系，不直接考虑资金的时间价值和其他约束。其他因素随后通过启发式算法以迭代方式进行添加。请注意，这些启发式算法既无法确保找到最优解决方案，也无法提供解决方案质量的估计值。

从整体上考虑问题的更出色解决方案有望为采矿工程团队提供更高质量的解决方案。可以明确包含配矿规范等的复杂约束，而使用近似方法往往无法非常妥善地处理这些约束。

可以使用混合整数线性规划（MILP）对矿山运营寿命进行建模。其核心理念是为单位时段的每个区块分配一个[0,1]变量，然后利用区块集的时间和空间属性来表述问题。

使用这种建模范式有很多优势：

- 建模是整体性的。
- 可以使用严格的数学方法来求解问题，以达到最优或可证明接近最优解。

- MILP 是一种全局优化方法。与启发式算法相比，全局优化方法可以实现超过 10% 的利润增长，所包含的项目研究也证明了这一点。

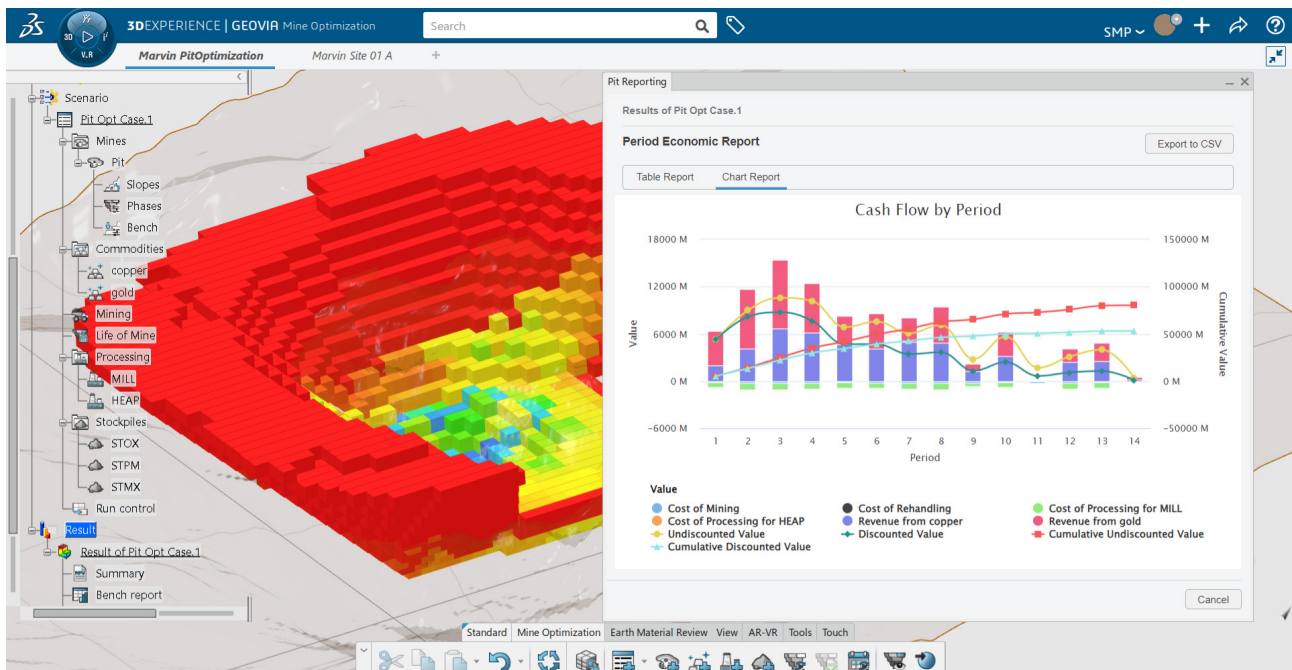
- 全局优化方法往往能解决简单近似方法无法解决的复杂问题。

直到最近，由于商业 MILP 解决方案软件包的功能有限，因此可以解决的问题的规模也是有限的。Bienstock-Zuckerberg (BZ) 算法是近年来取得的新进展，其使问题的规模和/或求解时间有了质的飞跃。BZ 算法并不直接求解 MILP，而是求解更简单的线性规划（LP）问题。LP 解是近似方法或求解整个 MILP 的绝佳起点。

直至今日，如果不参阅数学优化论文和报告，就无法轻松地理解 BZ 算法。在这篇论文中，我们不仅会提供关于该算法的通俗易懂的描述，还会介绍我们在该领域的研究成果。

GEOVIA 研发团队利用 BZ 算法创建了一个新的矿业规划优化引擎，即 GEOVIA Mine Maximizer (GMX)。GMX 是由 3DEXPERIENCE 平台上的 GEOVIA Strategic Mine Planner 和 GEOVIA Pit Optimizer 角色提供独家支持的解决方案引擎。自 2010 年推出以来，得益于一系列的加速方法，BZ 算法得到显著改进。此外，我们还提供了关于 GMX 中使用的加速方法的易于理解的详细信息，这些信息来自我们作者和其他作者的研究成果。





图片 1: Strategic Mine Planner 和 Pit Optimizer 角色中包含的矿业优化应用程序快照

这篇论文的主要目标之一是展示与使用商业软件包建立和求解 MILP 相比，这种方法所具备的优势。这些优势包括：

- 解决问题的速度快了几个数量级。
- 求解的问题无需近似（如逐期求解或使用滑动时间窗口等），而是可以整体解决。
- 由于求解器功能更强大，因此可以解决规模更大的问题。

我们可提供标准 Minelib 文献问题集[7]的结果，将其与 Whittle 中的 GEOVIA Simultaneous Optimizer 角色中使用的 Prober B 引擎进行比较，从而展示我们的工作质量。对本篇论文进行一下总结，我们主要介绍了这项工作如何与 GEOVIA 矿业规划工具集进行整合，并讨论了我們工作的未来发展方向。

在接下来的章节中，我们将区块视为一个单一的调度单元，区块之间存在优先级关系，以捕获岩土约束。此外，工程师还可以预先设计采矿工作台和采矿面板，规定在开始开采下一组矿块之前，必须完全开采完一组矿块。我们将这两种情况都建模为待开采的矿仓，矿仓是群集分组的一部分。矿块问题可视为只包含一个矿仓的集群问题，因此我们在工作说明中没有区分这些情况，而是创建了一个解决方案来解决所有问题。

## 战略矿业规划的一般 MILP 方法的局限性

解决战略矿业规划问题的主要难点在于问题的规模。问题的变量的数量与矿仓数  $\times$  时段数  $\times$  目的地数成正比。此外，必须满足的单位时段边坡约束也数以百万计。除了规模较小的问题外，这种公式对于商业 MILP 软件包来说是难以处理的。由于内存限制，通常无法加载公式，求解时间也长得惊人[15]。

可以缩小问题规模，以便使用商业软件包进行求解。可以汇总各区块，或逐期求解，或使用滑动时间窗口进行求解。尽管采取这些步骤可能意味着可以生成解决方案，但过多的汇总可能会导致净现值 (NPV) 下降 20%-30%[19]。在迭代过程中只考虑一小部分时段是一种目光短浅的行为，即前期规划未考虑后续因素。这可能会导致净现值进一步下降，或者无法找到复杂问题的解决方案。

理想情况下，我们希望最大限度地减少使用这些缩小规模的方法，但如何才能解决这些规模较大的问题呢？请注意，商业 MILP 软件包对基本问题并不是很了解。无论是护士轮班调度、船舶调度还是战略矿业规划，所有这些问题都采用类似的解决策略。BZ 算法的有效性来自于对特定战略矿业规划问题结构有一定的了解，并在算法开发过程中加以利用。

## BZ 算法

BZ 算法最初由 Beinstock 和 Zuckerberg 于 2009-2010 年提出，是一种用于解决较简单的线性规划（LP）问题的拉格朗日分解方法，是解决 MILP 的重要步骤。

该算法通过将公式的约束集分解为两部分，可高效地利用采矿问题结构的优势：

- 优先级约束集
- 一小部分硬约束集

优先级约束是指要求一件事在另一件事之前发生的约束。它们包括每个时段的边坡约束，以及如果某件事情在某一时段之前发生，那么它也会在下一个时段之前发生的约束。

硬约束是更常用的采矿约束。硬约束包括单位时段处理的最大矿石量、最大开采吨数以及单位时段最低品位等的约束。

图1对该算法进行了总结。

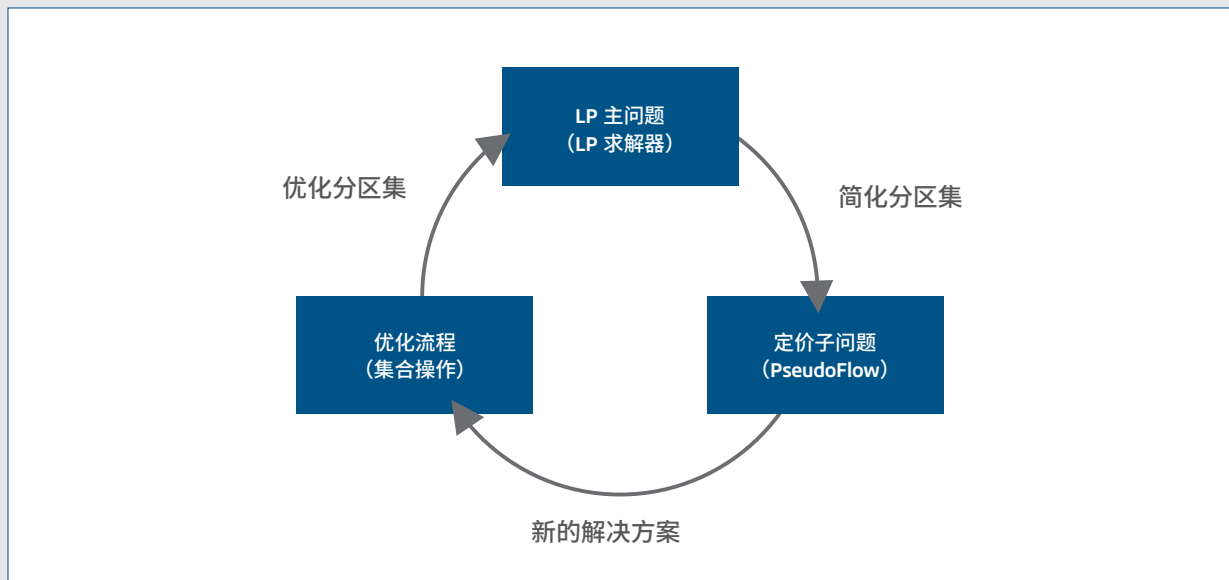


图 1: 算法概要模式

以下观测结果有助于理解算法：

### 定价子问题（Pseudoflow）

- 使用 Pseudoflow 算法 [9]，可以将仅有优先级约束的问题作为网络流问题进行高效解决。
- 如果使用拉格朗日松弛法将硬约束纳入相关目标，那么这个问题（定价子问题）就会产生求解目标的上限。

### LP 主问题

- 可以使用变量分区而不是单个变量本身来表述相关问题。仅将变量分配给一个分区。分区的分配必须使问题有可行解。如果没有掌握初始可行分配，则可以使用人工变量。
- 借助分区，线性规划（LP 主问题）的解可以得到下一个定价子问题所需的拉格朗日乘数。
- 当前迭代中的变量分区包含上一次迭代的分区，这意味着 LP 主问题的目标函数值是不递减的。
- LP 主问题产生了求解目标的下限。

## 优化流程（集合操作）

- 定价子问题的解用于创建下一个分区集，具体总结如图 2 所示。
- 通过合并在上一次迭代中具有相同解值的分区，可以定期简化分区集。这样可以保持较少的分区数量。

该算法一直进行到 LP 主问题和定价子问题的解收敛为止。

我们在图 2 中对优化流程进行了总结。

将原始分区集与定价子问题中新的 Pseudoflow 解决方案进行比较，以创建相交（Intersection）、不相交（Prior）的分区集，以及在 Pseudoflow 解决方案中但不在原始分区集中的变量集（New）。

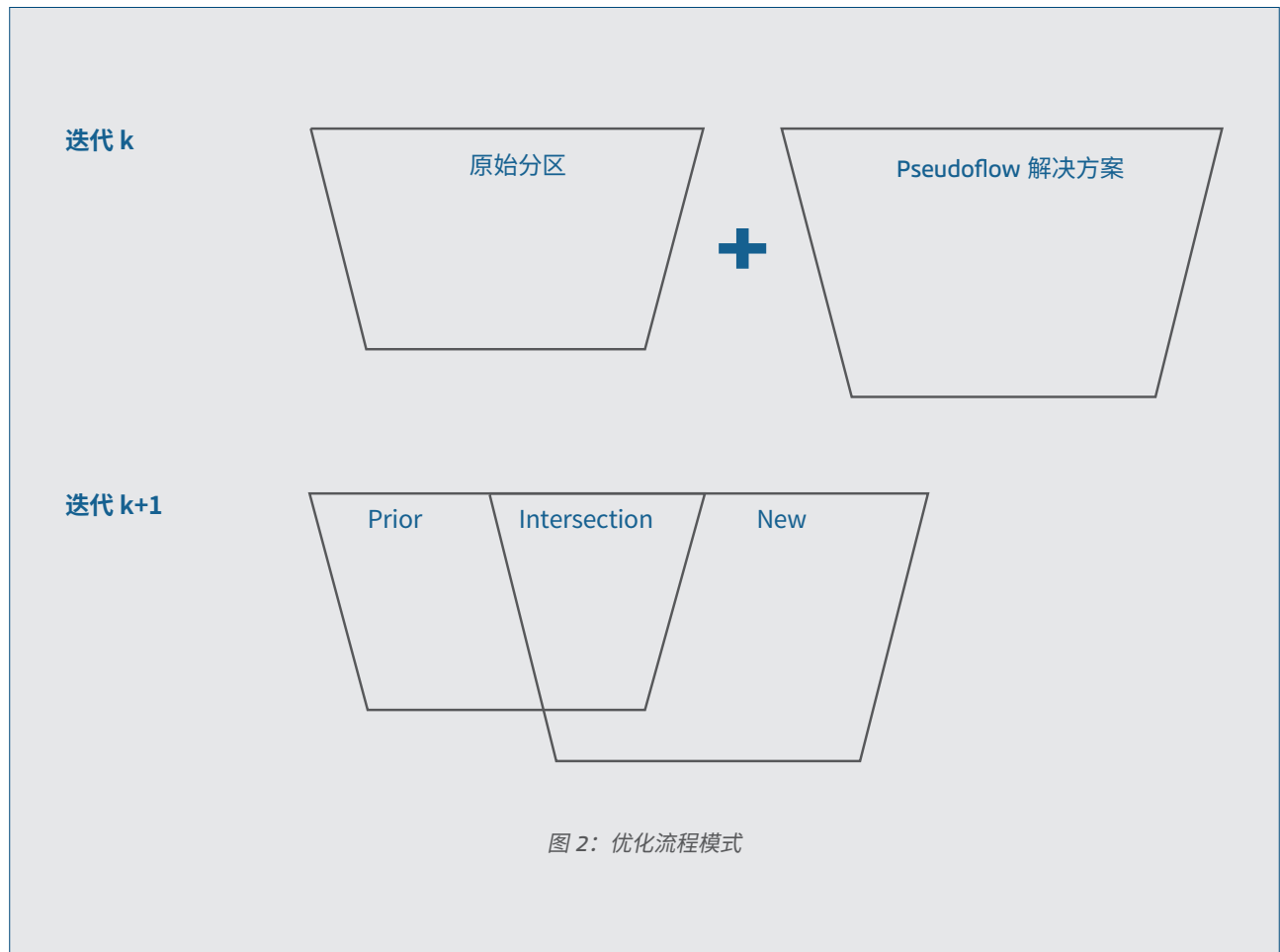


图 2：优化流程模式

## 运行时特性

BZ 算法可以高效地解决矿业规划问题的 LP 表述。为了证明这种效率，我们对文献问题 KD 的结果进行了比较，也就是使用 BZ 算法和商业求解器来求解 LP 的结果。KD 是一个具有 14,153 个矿块的铜矿，其中 12,154 个出现在最终矿井中，需要在 12 个时段内进行规划。每个矿块可前往两个目的地，分别作为矿石或废料进行处理。

使用 Simplex 或 Barrier 算法求解 LP 的商用 MILP 软件包，可在 4,710 秒内完成 LP 问题的求解。借助我们最新的 GMX 实现，我们只需 2.6 秒就能解决 LP 问题，速度提高了 1,811 倍！

接下来，我们将详细介绍我们用来加速原始算法的方法，包括文献中的方法以及我们自己的方法。

# BZ 算法加速

关于文献中的众多加速方法，我们可以参阅 Muñoz 等人的论文[16]。我们的工作包括了这些成果以及产生快速结果的新工作。整体而言，与 2010 年论文[2]中算法的原始实现相比，我们在九个标准 Minelib 文献测试问题上的总运行时间锐减了 87 倍。

用于加速 BZ 算法的方法总结如下：

## 预处理

**UPIT：**我们首先求解最终矿井的极限问题，然后对解决方案区块进行调度。事实证明，这一步骤不会降低求解的最优性[3]，而且往往能显著缩小所考虑问题的规模。

**减少浪费：**可以排除那些永远不会被加工为矿石的废矿块的可能目的地。如果在吨位和利润方面无利可图，则不将该废块作为矿石进行处理。

**尽早开始：**考虑开采约束能力，确定一个集群的最早移除时间。我们利用这些最早开始时间，通过移除 LP 主问题中的变量和 Pseudoflow 问题中不必要的节点，来强化问题表述。

## 定价子问题（Pseudoflow）

BZ 算法的大部分工作是解决定价子问题，其中包括使用 Pseudoflow 算法解决网络流问题。在测试问题 KD 中，解决 Pseudoflow 问题占我们实现工作的 56%。

**Mineflow：**就我们的目的而言，我们在 Deutsch、Dağdelen 和 Johnson [6]的最新研究成果基础上创建了一个新的快速 Pseudoflow 解算引擎。通过专门针对最终矿井极限问题定制算法，Mineflow 显示出比原始 Pseudoflow 引擎更快的速度。

**热启动：**正如 Hochbaum [9] 所概述的那样，Pseudoflow 算法可以使用存储为规范化树的前一个解进行热启动。在我们的工作中，当从一次迭代到下一次迭代的解预计相似时，我们就会热启动 Pseudoflow 算法。只要最优性差距低于容差，我们就认为解决方案是相似的。

**路径收缩：**有待解决的 Pseudoflow 问题只包含优先级约束和硬约束，这些约束用于修改从一个迭代到下一个迭代的目标。在一个时段内，一个集群矿仓的每个目的地只依赖于该矿仓的其他目的地，事实上，它们形成了一个链条。根据 Muñoz 等人的研究成果[16]，对于一组给定的拉格朗日乘数法，可以对每个时段的这些矿仓目的地进行预处理，以确定如果该矿仓在该时段内被移除，其最佳目的地是哪里。然后，利用优先级关系，可将每个时段的矿仓目的地节点收缩为一个集群，从而显著减少待解 Pseudoflow 问题的规模。

## LP 主问题

在 BZ 算法的每次迭代中，都会创建并解决一个小型 LP 问题。在我们的实现中，我们使用了问题的“By”表述，我们发现这种表述比“Att”问题表述更为可取。

**LP 创建：**我们从最后一个时段到第一个时段，从一个矿仓的最后一个目的地到第一个目的地创建 LP。这样可以减少工作量，因为我们深知，如果一个变量在一个实例中没有进行定义，那么根据优先级约束的链式属性，其也不会在前一个实例中进行定义。此外，我们还会跟踪已添加到 LP 中的约束，以避免重复。

**热启动：**由于分区的优化方式，我们借鉴了 Muñoz 等人的研究成果[16]，将初始分区值设为上一次迭代中的值，并将新的分区值设为零。

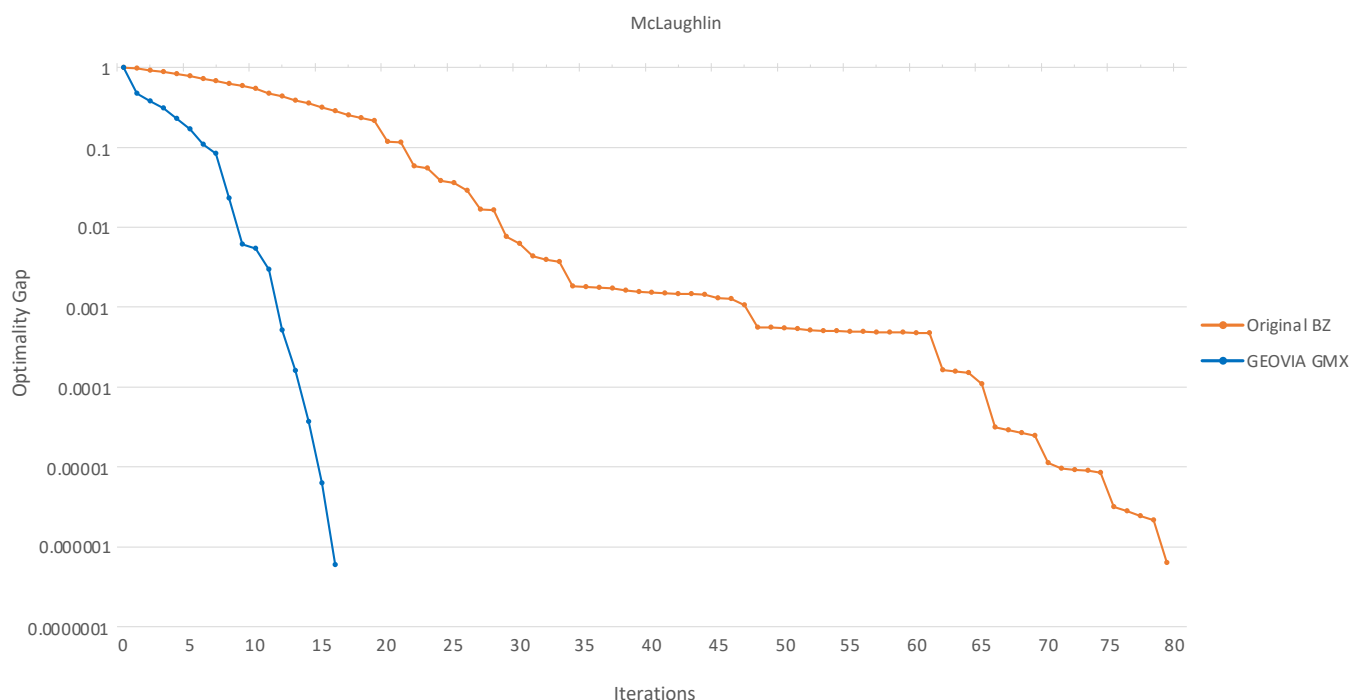
## 优化流程（集合操作）

**初始分区：**一个好的初始可行解可提供出色的初始分区集，从而大幅减少 BZ 的迭代次数。我们使用一种新的启发式方法来初始化分区集，这种方法也包含在我们的方案中，可以减少 BZ 的迭代次数。

**计算：**每个分区都存储为一个未排序的标识符列表。此外，我们还存储了标识符与所属分区的映射，且保持时效。通过将 Pseudoflow 解决方案与这些数据结构结合使用，可以高效地确定从一次迭代到下一次迭代的分区。

# 收敛

我们采用一种新的专有方案来显著减少 BZ 的迭代次数。图表 1 举例说明了这种方法的有效性。对于最大的 Minelib [7] 文献问题，麦克劳夫林（McLaughlin）问题，使用我们实现的原始 BZ 算法的迭代次数为 79 次，而新版本则锐减至 16 次。值得注意的是，最优性差距从一次迭代到下一次迭代的变化率，重点强调了更快的收敛速度和更快的运行时间。



图表 1: 麦克劳夫林 (McLaughlin) 问题解法的收敛性比较

# 堆料

矿堆具有固有的非线性特性。堆放矿石的品位是已知的，但一旦从矿堆中移除一定数量的矿石，然后再在其上堆放其他矿石，矿堆的平均品位就会变成一个非线性函数。尽管可以创建非线性模型来代表矿堆，但事实证明，除了小规模的问题外，这些模型还是很难进行处理的[14]。

为了利用线性约束建立矿堆模型，我们借鉴了 Moreno 等人最近的研究成果[14]。在这里，我们指定了每期移除的矿石的平均品位，并新增了一些约束，以确保前几个时段堆放的矿石的平均品位至少达到指定值。由于堆放品味更高的矿石可能比原有矿堆所产生的利润更低，因此，在解决方案中，对于两种不同品位的矿石要进行匹配。

尽管这种建模方法很常用，但我们发现 Moreno 等人[14]的求解程序也能得出很不错的结果。

# 计算结果

表 1 报告了九个Minelib文献问题[7]的结果。Minelib 问题是各种规模的直接区块调度（DBS）问题，最多有两个处理或开采能力约束。所有测试运行都是在标准的联想 ThinkPad P53 笔记本电脑上完成的，该笔记本电脑配备第 9 代 i7 CPU、6 核、运行频率为2.6 Ghz、内存容量为 32 GB。下限来自 LP 主问题的求解，上限来自定价子问题的求解。如果最优性差距（1.0-下限/上限）最多为  $10^{-6}$ ，那么我们就可以说BZ算法已经收敛。在最后两列中报告迭代次数和完成这项工作所需的时间（以秒为单位）。

一旦解决了 LP 问题，我们会运行 TopoSort 启发式算法以及优化-目的地启发式算法[17]的十种变量，以创建接近最优的解决方案。这些启发式方法的总运行时间很短，平均不到一分钟就能完成。

数据	区块	时段	目的地	优先级	上限	可行解	差距	文献差距	BZ 迭代	BZ 运行时间 (秒)
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
mcLaughlin_ 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_medium	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

表 1: Minelib 问题的 GMX 结果

将最优性差距（Gap）与最佳文献结果（Literature Gap）进行比较，我们注意到在每种情况下都能生成最佳/接近最佳的解决方案。此外，生成一个解决方案的最长运行时间不足 5 分半钟。最后，我们的结果与专门的元启发式方法的结果相比非常有利，后者可能需要数小时才能运行。



## 项目研究

为了公正地评估 GMX 引擎的有效性，最好将其与功能完备的商业工具进行比较。现有的许多研究都与 GEOVIA Whittle 进行了比较，主要集中在 Milawa 引擎上。Milawa 是一种广泛使用的优化器，专门用于优化面板调度（回推/工作台面级问题）。不过，Prober B 是 Whittle 提供的更出色的优化引擎，是 Simultaneous Optimizer (SIMO) 角色的一部分。该求解器在面板级进行调度，按矿仓品味进行分块汇总，并提供比 Milawa 更广泛的优化范围，可并行处理边界品位、矿堆和配矿问题。

G. Whittle 在 2016 年发表的论文《Advanced SIMO vs Milawa and SPCO》[18]可为 Prober B 引擎在 SIMO 中的有效性提供支持。该论文将 Prober B 与其他 Whittle 调度引擎进行了比较。在这篇论文中提到了一项研究，研究结果表明，使用 Prober B 的净现值提高了 41.4%，而使用 Milawa 和 SPCO 的净现值则只提高了 28.4%。

我们对 GMX 解决方案引擎与 Prober B (SIMO) 在金银矿项目研究中的结果进行了比较。该矿场由两个矿井组成，共有 24,331 个矿仓，开采期超过 11 年。我们提供了一个三阶段案例和一个七阶段案例，它们分别由 49 个和 119 个面板组成。对于七阶段案例，还探讨了堆场问题。

在这些案例中，GMX 可以使用原有的矿仓数量，而 Prober B 则需要进一步对矿仓进行汇总，以避免更长的运行时间。此外，Prober B 还需要配置一些参数才能获得最佳结果。

在这三个案例中，与使用 Prober B 相比，经验丰富的用户使用 GMX 可实现 3.1% 至 15.7% 的净现值提升。

另外，从这些案例的测试结果和另外 20 个项目研究的研究结果中，我们还发现 GMX 的运行速度平均比 Prober B 快 22 倍。这种运行速度可允许工程团队在制定战略矿业规划时考虑多种不同的场景。

## 多维场景设计

GEOVIA 最近实施了多个解决方案，以改进露天矿战略规划工作流程。GMX 是由 3DEXPERIENCE 平台上的 GEOVIA Strategic Mine Planner 和 GEOVIA Pit Optimizer 角色提供支持的解决方案引擎。除了作为单一真实数据源外，3DEXPERIENCE 生态系统还可提供一系列功能强大的工具，包括设计探索、参数分析和多学科工程优化。

使用 3DEXPERIENCE 工具套件可进行多维场景设计。可在一个看板中创建使用不同数据源和不同参数值的试验，能够探索许多不同的财务和技术参数。

可以快速解决以下问题：

- 如果净现值折现率从 10% 增长到 12%，会对设计产生怎样的影响？
- 如果在第五年提高卡车运输能力会怎样？
- 如果以后几年的碳排放量减少 10%，又会怎样？

进行这种分析可以让采矿工程师探索许多不同的场景，否则他们就没有时间这样做。多维场景设计可以降低项目风险，提高效率，并获得更高的利润净现值。就项目工作而言，在某些情况下，我们看到净现值利润增加了 20-30%。

## 结论

在这篇论文中，我们对BZ算法及其加速方法进行了通俗易懂的介绍。这项工作是在新的GMX解决方案引擎中实现的，该引擎由 **3DEXPERIENCE** 平台上的 **GEOVIA Pit Optimizer** 和 **GEOVIA Strategic Mine Planner** 角色提供独家支持。

我们对 **GMX** 和商业 **LP** 解决方案软件包的运行时间进行了比较，结果表明 **GMX** 在处理文献问题 **KD** 时要快 **1,811** 倍。此外，我们还展示了九个文献 **Minelib** 问题 [7] 的快速运行时间，在每种场景下都生成了最佳/接近最佳的解决方案。最后，我们比较了 **GMX** 和 **Prober B (Whittle)** 的解决方案，客户研究案例使用 **GMX** 实现了 **3.1%** 至 **15.7%** 的净现值提升，同时还获得了解决方案运行时间上的优势。综上所述，与使用商业软件包表述和求解 **MILP** 以及直接应用启发式方法相比，这些结果证明了新的 **GMX** 解决方案引擎的强大功能和优势。

**BZ** 算法是解决矿业规划问题的一个重要里程碑。然而，我们的研究工作在这里还远没有结束。回顾一下，**BZ** 算法并不解决矿业规划问题，而是可提供一种高效的方法来解决该问题的线性规划 (**LP**) 公式。为了进一步推进我们的工作，我们希望将 **BZ** 算法作为一个高效的子程序，用于解决难度更大的 **MILP** 问题。

正如我们的 **KD** 测试问题的试验结果所表明的那样，随着问题规模的增大，使用标准 **Simplex** 或 **Barrier** 方法来解决矿业规划程序会成为一个巨大的瓶颈。因此，我们未来的工作并不是在预处理步骤中使用 **BZ** 算法来解决小型的 **MILP** 问题，而是研究一种定制的 **MILP** 解决方法，将使用 **BZ** 算法作为 **MILP** 解决方法的核心。此外，使用并行化将显著缩短大型复杂问题的最佳/近似最佳解决方案的生成时间，这正是采矿工程师所希望的。

## 客户案例

- [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](http://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/>

## 关于作者



**Stephen Hill**

运营研究学博士，达索系统澳大利亚办事处 GEOVIA 研发部矿业规划与优化专家。自 1998 年起开始从事矿业优化工作。



**Victor Xiaoyu Bai**

地质统计学与矿业优化博士，达索系统澳大利亚办事处 GEOVIA 研发部软件工程高级经理。自 2009 年起开始从事矿业优化工作。



**Iain Rudge**

工程学学士、数学学士，达索系统澳大利亚办事处 GEOVIA 研发部软件工程专家。自 2022 年起开始从事矿业优化工作。



**David Place**

地球科学博士、数学和物理学学士，达索系统澳大利亚办事处 GEOVIA 研发部软件工程总监。自 2022 年起开始从事采矿软件工作。



**Cristian Poblete**

采矿工程师，工程科学硕士，达索系统澳大利亚办事处 GEOVIA 研发部采矿与冶金应用组合高级经理。自 2011 年起开始从事战略矿业规划工作。

