Mixed-Integer Programming Techniques for Strategic Open-Pit Mine Planning

Gonzalo Muñoz – Polytechnique Montréal

COSMO – McGill University – May 16, 2018

This talk describes joint work with:

Eduardo Moreno
Universidad Adolfo Ibáñez

Marcos Goycoolea
Universidad Adolfo Ibáñez

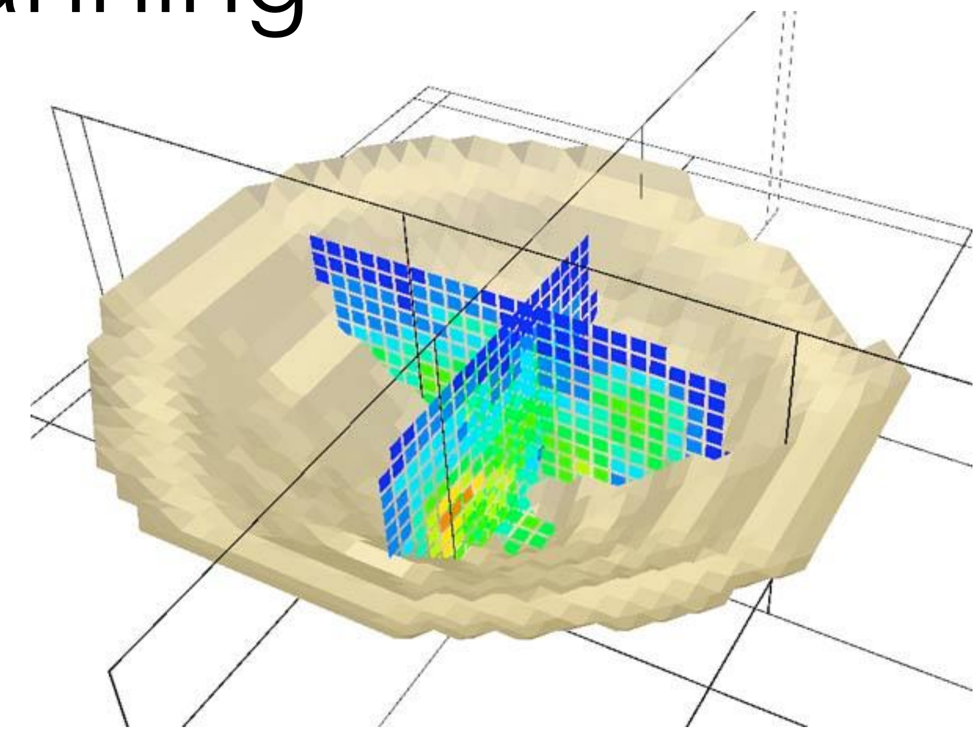
Daniel Espinoza
Gurobi Optimization

Orlando Rivera
Universidad Adolfo Ibáñez

Maurice Queyranne
University of British Columbia

Strategic Open Pit Mine Planning

Which blocks should be extracted?
When should they be extracted?
What should their destination be?



WikiMedia Commons (File:RC drill rig.jpg)

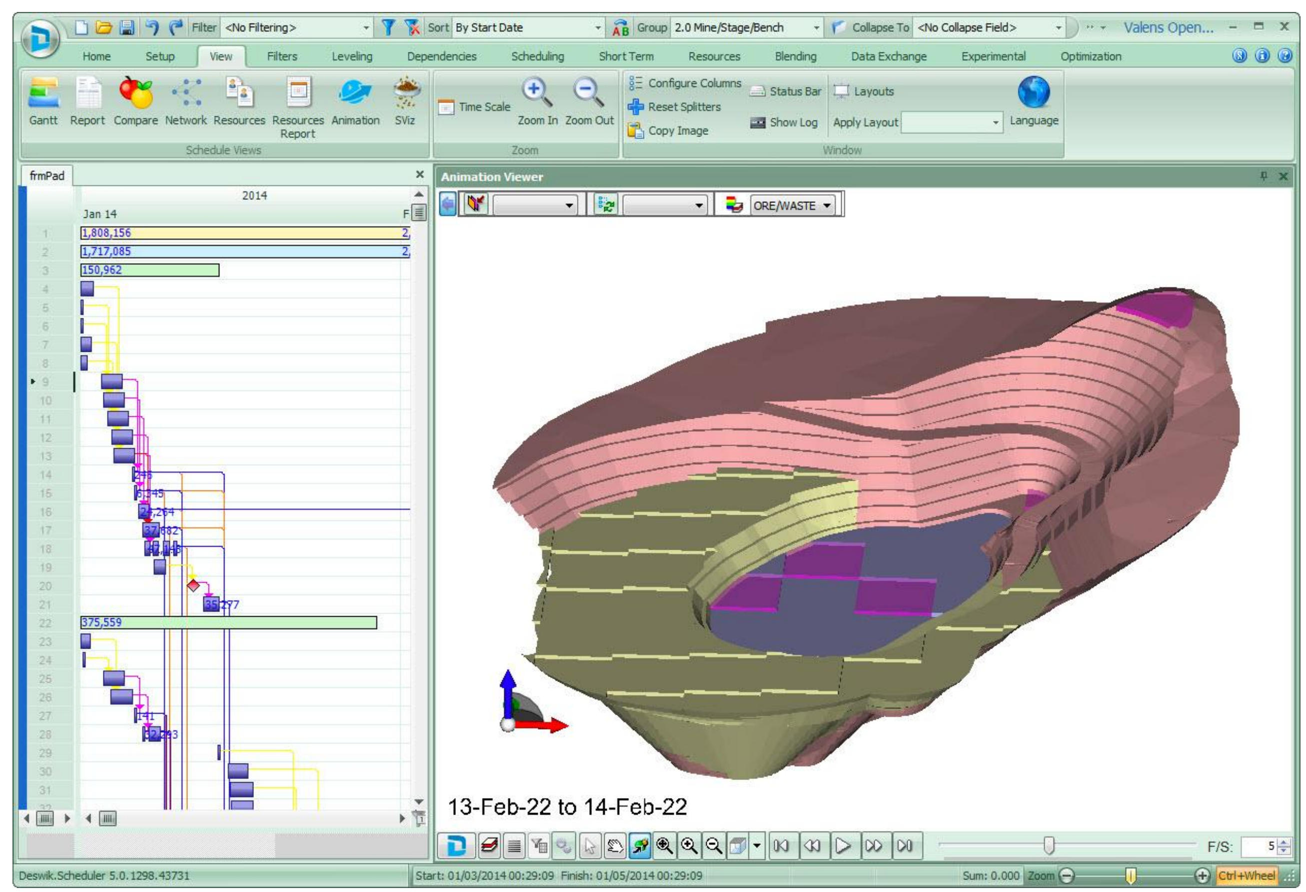
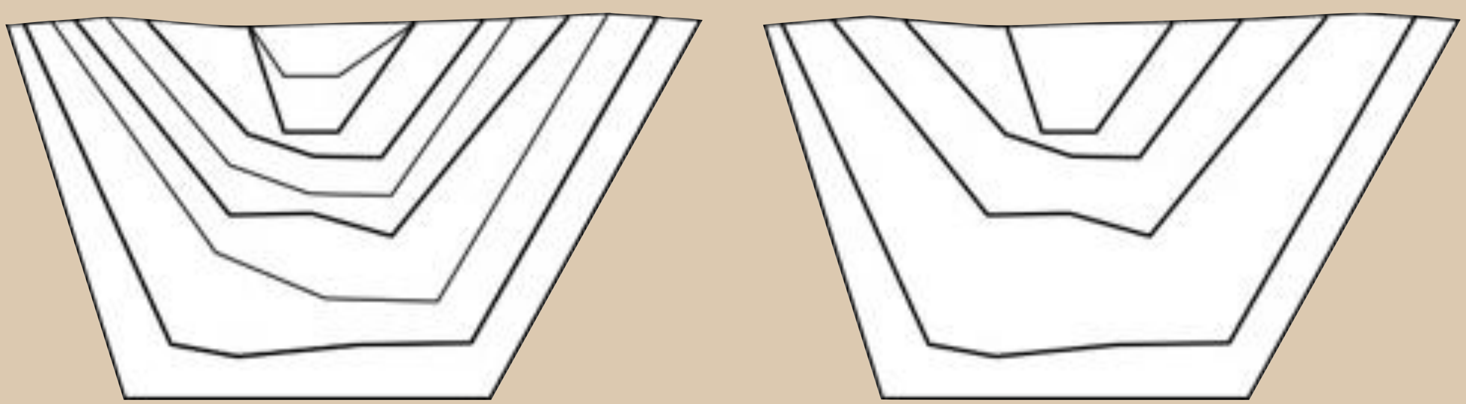


Image courtesy of Deswik Pty

Traditional Approach



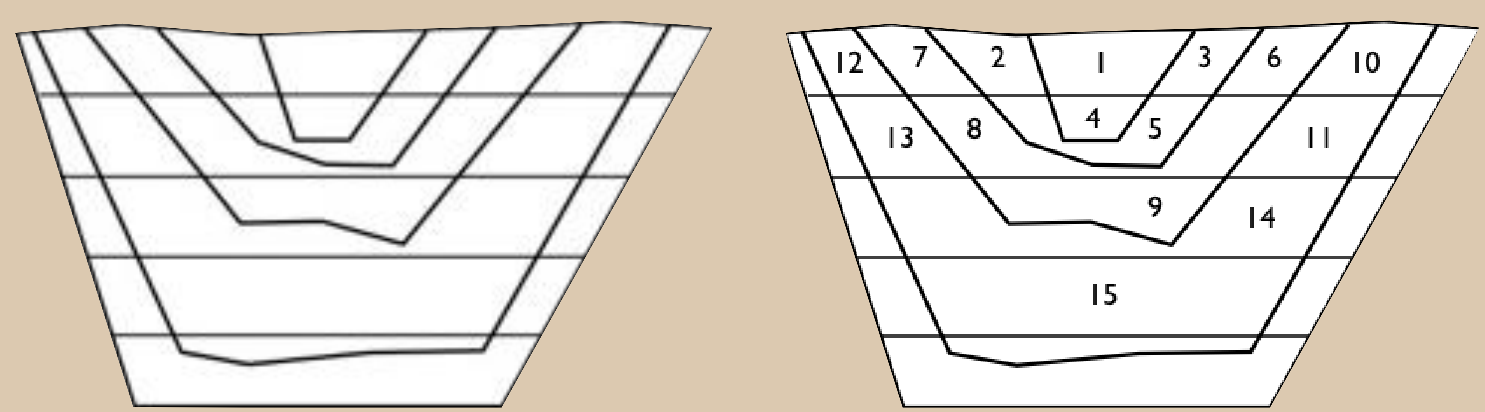
Step 1. Phase Definition.

Lerchs and Grossman 1965

Define phases using a sequence of nested pits (Compute arcs and use parametric version of LG algorithm, increasing prices) [use fixed cut-off grade to distinguish ore from waste]

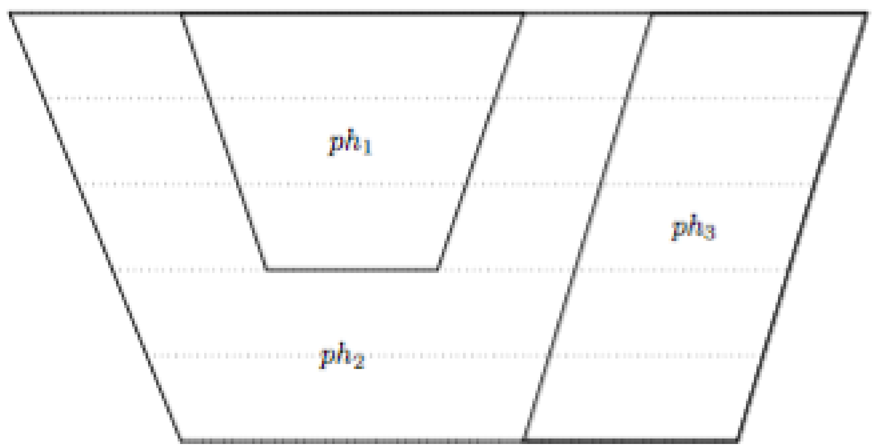
Kenneth Lane 1964

Schedule extraction over time at an increment (bench-phase) level using a dynamic programming approach that maximizes net-present value while satisfying constraints.

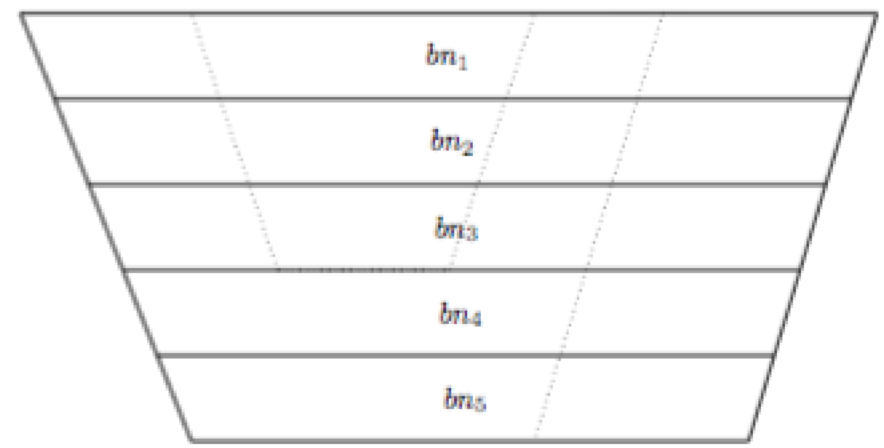


Step 2. Production Scheduling.

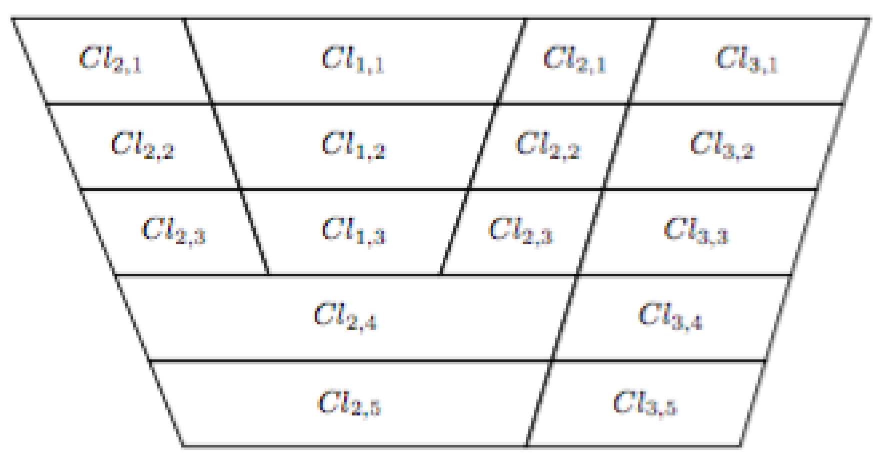
Phase Design and Production Scheduling



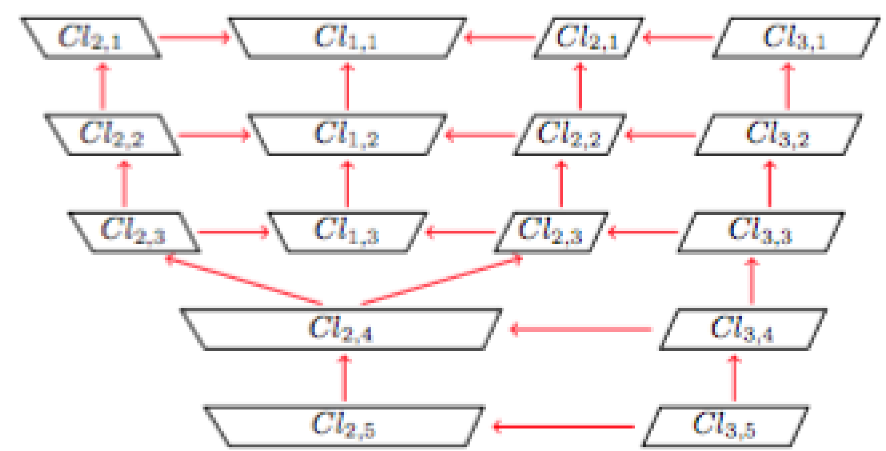
(a) Phases



(b) Benches

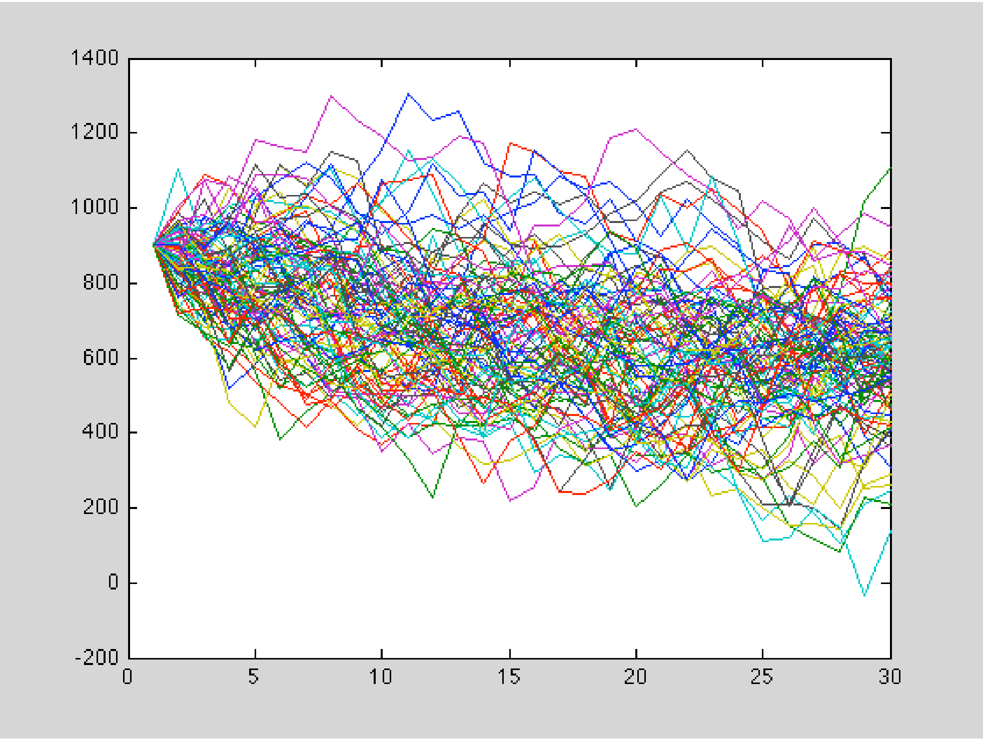


(c) Clusters

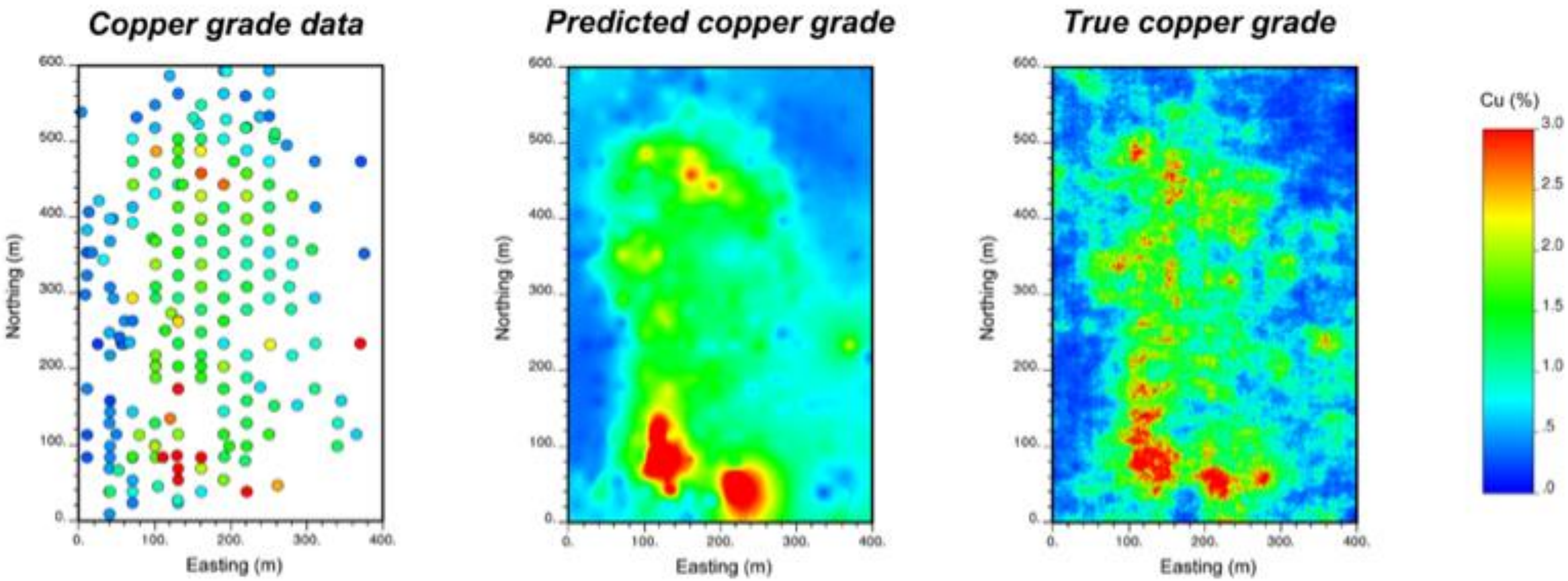


(d) Immediate Precedences

Can this be done with integer programming?



Where is the uncertainty!?

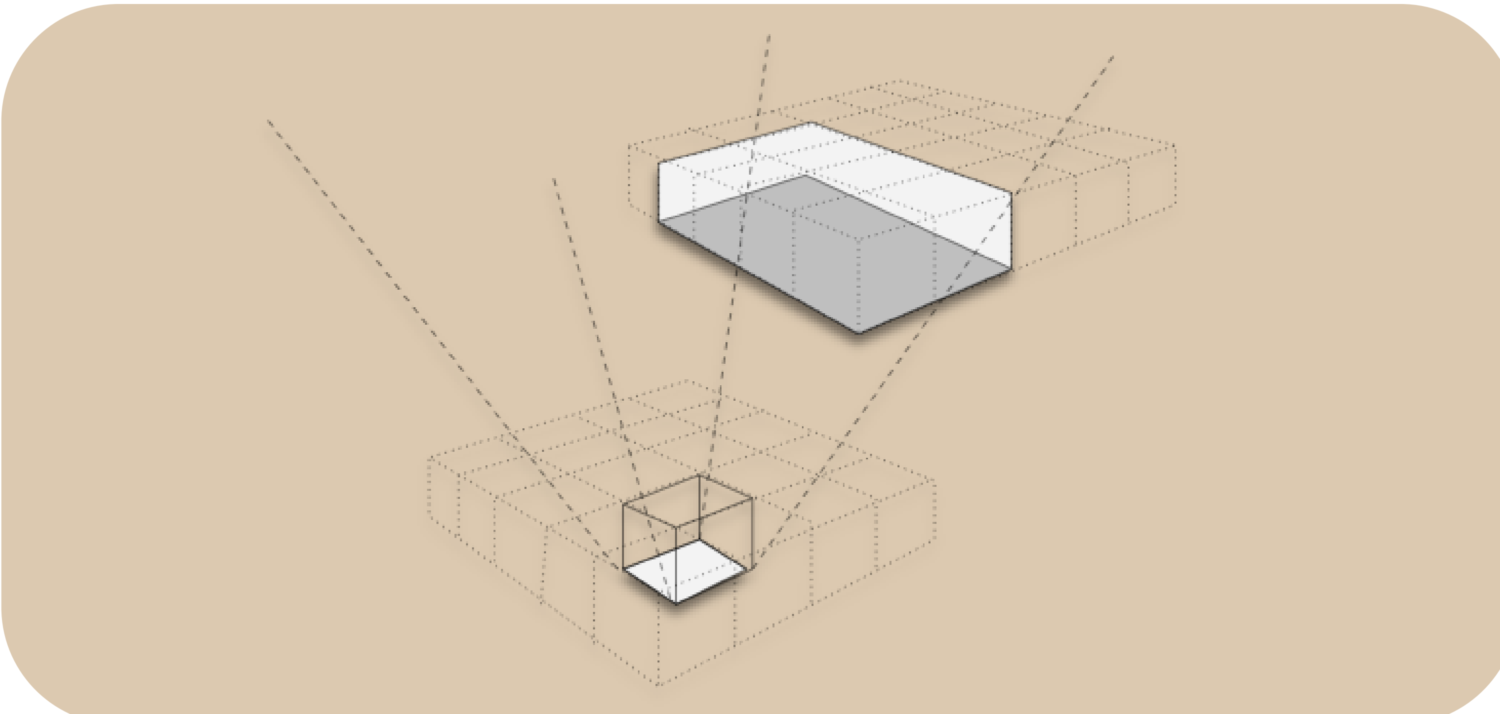


Why we are not considering uncertainty (for now)

- Size of mines → *exact* optimization is intractable
- Ultimate goal = stochastic version, but even the deterministic version is approximated for solving
- Many approaches use elaborate heuristics for obtaining good solutions, but unfortunately obtaining optimality guarantees seems impractical
- Solving deterministic instances to provable optimality, can be used to *evaluate scenarios,* or as an *oracle* to other optimization routines

Phase Design Formulation

Phase Design Formulation



$$x_{b,t} = \sum_{d=1}^{D} y_{b,d,t}$$

$x_{b,t} \in \{0,1\}$ Extract block "b" in time "t"

$y_{b,d,t} \in [0,1]$ Proportion of "b" sent to "d" in "t"

$$\sum_{t=1}^{T} x_{b,t} \leq 1 \qquad \forall b \qquad\qquad \sum_{s=1}^{t} x_{b,s} \leq \sum_{s=1}^{t} x_{a,s} \qquad \forall (a,b)$$

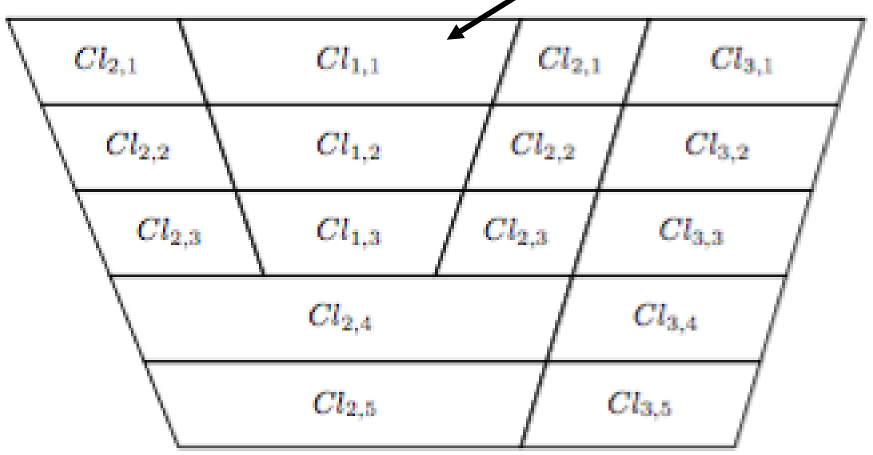
+ Resource consumption constraints that limit extraction per period and tons of material sent to each destination (mill, waste-pile, etc.).

Production Scheduling Formulation

Production Scheduling Formulation

$x_{c,t} \in [0,1]$ Percentage of cluster ‘c’ extracted in time ’t'

Each cluster is a bench-phase comprised of multiple blocks

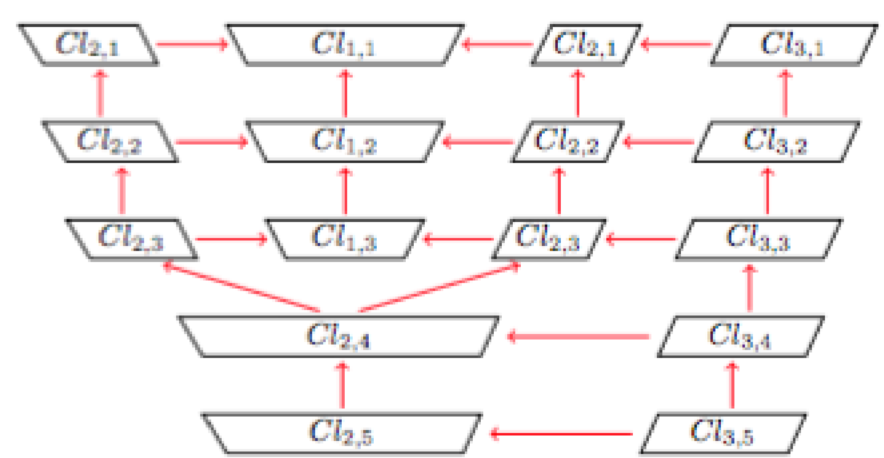


Entirety of a cluster constraint:

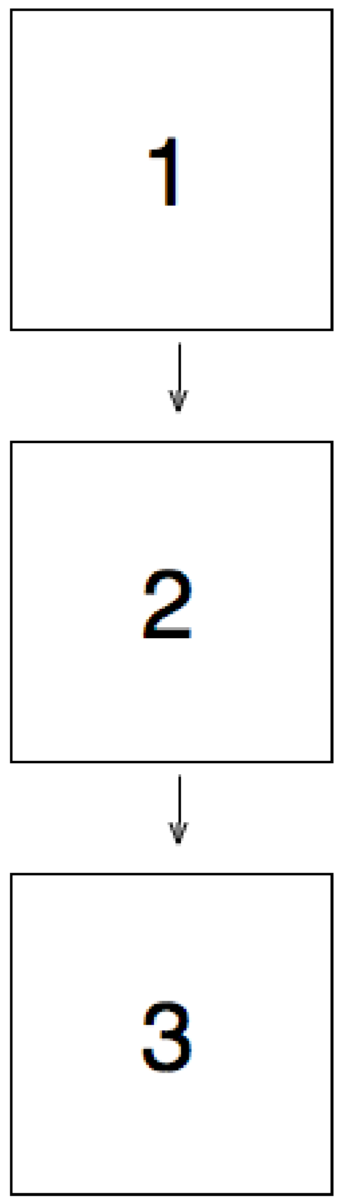
$$\sum_{t \in T} x_{c,t} \leq 1$$

If cluster ‘c’ is a predecessor of cluster ‘d’ then:

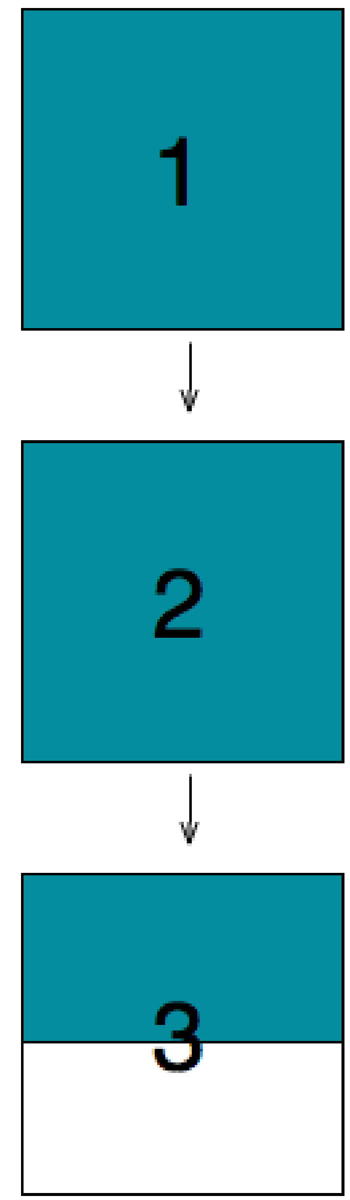
$$\sum_{t=1}^{\tau} x_{d,t} > 0 \Rightarrow \sum_{t=1}^{\tau} x_{c,t} = 1$$



Tricky integrality condition



$$x_{c_1,t} \leq x_{c_2,t} \leq x_{c_3,t}$$



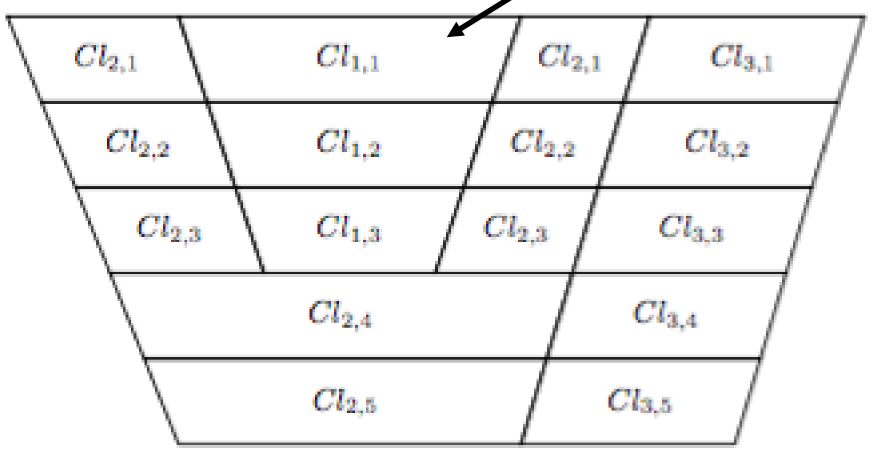
$$x_{c_3,t} > 0 \Rightarrow x_{c_2,t} = 1$$

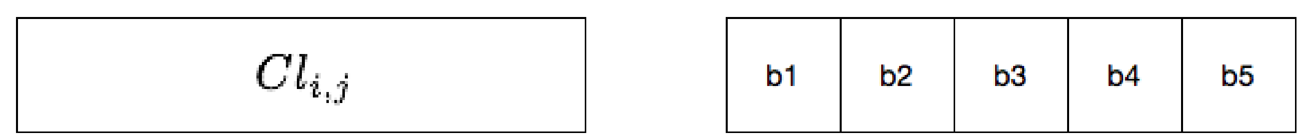
$$x_{c_2,t} > 0 \Rightarrow x_{c_1,t} = 1$$

Production Scheduling Formulation

$y_{b,d,t} \in [0, 1]$ Percentage of block ‘b’ sent to destination ‘d’ in time ’t'

Each cluster is made up of several individual blocks





Blocks in the same cluster must be extracted simultaneously:

$$x_{c,t} = \sum_{d \in D} y_{b,d,t} \qquad \forall b \in c$$

But destination decisions are made for each block individually!

$$\max \quad p \cdot y$$

s.t.

$$x_{c,t} = \sum_{d \in D} y_{b,d,t} \qquad \text{for all } c \in C,\ b \in c,\ t \in T$$

$$\sum_{t \in T} x_{c,t} \leq 1 \qquad \text{for all } c \in C$$

$$\sum_{t=1}^{\tau} x_{c,t} \leq \sum_{t=1}^{\tau} x_{c',t} \qquad \text{for all } (c, c') \in A,\ \tau \in T$$

$$Gy \leq g$$

$$y_{b,d,t} \geq 0 \qquad \text{for all } b \in B, d \in D, t \in T$$

$$x_{c,t}, y_{b,d,t} \text{ integral.} \qquad \text{for all } c \in C, b \in c, d \in D, t \in T$$

Very large multi-modal, batch RCPSP problem with funny integrality constraints, and possibly mean side-constraints (G).

SCALE

2-10 million blocks
1-5 elements of interest
2-5 destinations
20-100 time periods

Size of Full Formulations

	Phase Design			Production Scheduling		
	Periods	**Blocks**	**Variables**	**Clusters**	**Blocks**	**Variables**
calbuco	21	5,016,971	316,069,173	324	200,241	12,615,183
chaiten	25	339,199	16,959,950	273	288,073	14,403,650
guallatari	21	1,672,198	105,348,474	272	57,527	3,624,201
kd	12	14,153	339,672	53	10,128	243,072
marvinml	20	53,271	2,130,840	56	8,515	340,600
mclaughlin	20	2,140,342	85,613,680	173	180,749	7,229,960
mclaughlinlimit	15	112,687	3,380,610	166	110,768	3,323,040
palomo	40	772,800	61,824,000	74	190,319	15,225,520
ranokau	81	1,873,035	303,431,670	186	317,907	51,500,934
tronador	20	329,859	18,801,963	220	30,099	1,805,940

Espinoza, Goycoolea, Moreno, Newman. MineLib: A library of Open Pit Mining Problems. Annals of OR (2012).

Size of Full Formulations

	Phase Design			Production Scheduling		
	Periods	**Blocks**	**Variables**	**Clusters**	**Blocks**	**Variables**
calbuco	21	5,016,971	316,069,173	324	200,241	12,615,183

```
*** CPLEX ERROR 1001:  Out of memory.
```

	Periods	**Blocks**	**Variables**	**Clusters**	**Blocks**	**Variables**
mclaughlin	20	2,140,342	85,613,680	173	180,749	7,229,960
mclaughlinlimit	15	112,687	3,380,610	166	110,768	3,323,040
palomo	40	772,800	61,824,000	74	190,319	15,225,520
ranokau	81	1,873,035	303,431,670	186	317,907	51,500,934
tronador	20	329,859	18,801,963	220	30,099	1,805,940

Espinoza, Goycoolea, Moreno, Newman. MineLib: A library of Open Pit Mining Problems. Annals of OR (2012).

Solving large problems: A MIP approach.

Pre-Processing
Specialized Linear Programming Solvers
Heuristics
Cutting-Planes
Branching

O - Open
M - Mine
P - Planner

A specialized solver designed to solve a broad class of scheduling problems using open MIP techniques.

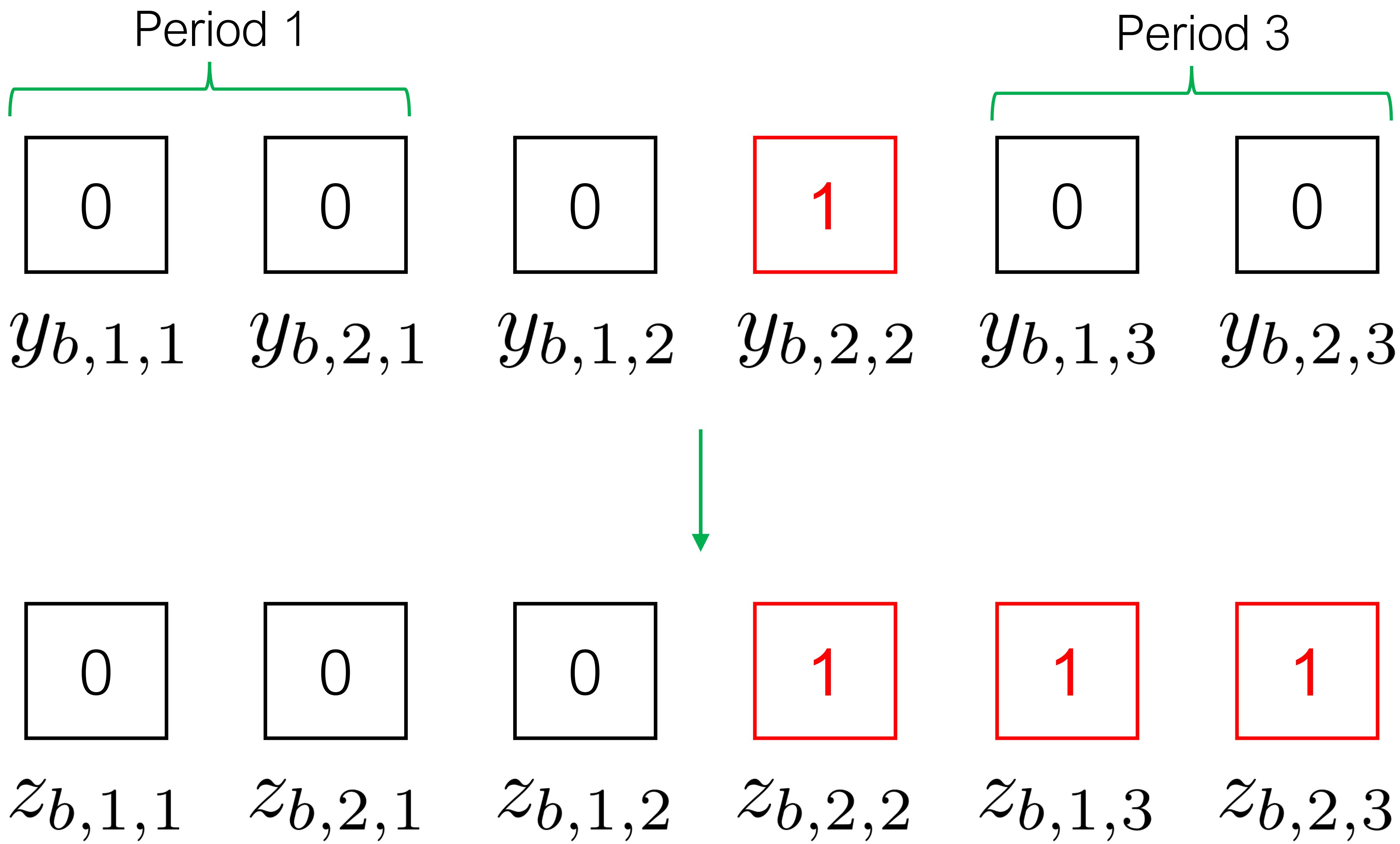
First ingredient: a powerful LP-Solver

(must be able to solve very large problems quickly)

$$
\begin{aligned}
\max \quad & p \cdot y \\
\text{s.t.} \quad & \\
& x_{c,t} = \sum_{d \in D} y_{b,d,t} && \text{for all } c \in C,\ b \in c,\ t \in T \\
& \sum_{t \in T} x_{c,t} \leq 1 && \text{for all } c \in C \\
& \sum_{t=1}^{\tau} x_{c,t} \leq \sum_{t=1}^{\tau} x_{c',t} && \text{for all } (c, c') \in A,\ \tau \in T \\
& Gy \leq g && \\
& y_{b,d,t} \geq 0 && \text{for all } b \in B, d \in D, t \in T \\
& x_{c,t}, y_{b,d,t} \text{ integral.} && \text{for all } c \in C, b \in c, d \in D, t \in T
\end{aligned}
$$

Very large multi-modal, batch RCPSP problem with funny integrality constraints, and possibly mean side-constraints (G).

A useful change of variables



A useful change of variables

$$z_{b,d,t} \leq z_{b,d+1,t}$$

$$z_{b,D,t} \leq z_{b,1,t+1}$$

$$z_{b,D,T} \leq 1$$

$$z_{b,D,t} \leq z_{a,D,t}$$

$$w_{c,t} = z_{b,D,t} \quad \forall b \in c$$

$$Hz \leq h$$

$$z_i \leq z_j \quad \forall (i,j)$$

$$z_{b,d,t} = \sum_{i=1}^{D} \sum_{j=1}^{t-1} y_{b,i,j} + \sum_{i=1}^{d} y_{b,i,t}$$

$$z_{b_1,D,t} = z_{b_2,D,t} \qquad \text{if } b_1, b_2 \in c$$

An equivalent formulation

$$
\begin{aligned}
\max \quad & c \cdot z \\
\text{s.t.} \quad & z_i \leq z_j \quad \forall (i,j) \in I \\
& Hz \leq h \\
& 0 \leq z \leq 1
\end{aligned}
$$

An equivalent formulation

$$\max \quad c \cdot z$$

$$\text{s.t. } z_i \leq z_j \quad \forall (i, j) \in I$$

$$Hz \leq h$$

$$0 \leq z \leq 1$$

Max closure problem

Millions

Hundreds

An equivalent formulation

$$\boxed{\begin{aligned} &\max \quad c \cdot z \\ &\text{s.t. } z_i \le z_j \quad \forall (i,j) \in I \end{aligned}}$$

Max closure problem

$$Hz \le h$$

Millions

$$0 \le z \le 1$$

Hundreds

Suitable for a decomposition method

We name the "easy" constraints $Az \le b$

Dantzig-Wolfe Decomposition

Master Problem (lower bound)

$$\max \quad c^t V \lambda$$

$$\text{s.t.}$$

$$HV\lambda \le h, \quad (\mu \ge 0)$$

$$1 \cdot \lambda = 1,$$

$$0 \le \lambda.$$

$$V = [v^1, \ldots, v^k]$$

$$Av^i \le b$$

Pricing Problem (upper bound)

$$\max \quad c^t v - \mu(Hv - h)$$

$$\text{s.t.}$$

$$Av \le b$$

In the mining problem:

Pricing is max-closure problem

"Easy" to solve, using Hochbaum's Pseudoflow algorithm.

Effectiveness of Dantzig-Wolf Decomposition

	Phase Design			Production Scheduling		
	DW			DW		
calbuco	2h 28m			8s		
chaiten	9h 34m			15s		
guallatari	733.94			5s		
kd	11s			250ms		
marvinml	14s			380ms		
mclaughlin	48m			2s		
mclaughlinlimit	9m			1s		
palomo	1h 30m			3s		
ranokau	3d 0h 42m			20m 30s		
tampakan	1788.18			5s		

Effectiveness of Dantzig-Wolf Stabilization

	Phase Design			Production Scheduling		
	DW	**DW+S**		**DW**	**DW+S**	
calbuco	2h 28m	40m		8s	13s	
chaiten	9h 34m	2h 27m		15s	17s	
guallatari	733.94	315.06		5s	8s	
kd	11s	8s		250ms	320ms	
marvinml	14s	10s		380ms	700ms	
mclaughlin	48m	15m		2s	2s	
mclaughlinlimit	9m	4m 35s		1s	1s	
palomo	1h 30m	36m 42s		3s	3s	
ranokau	3d 0h 42m	10h 52m		20m 30s	14m	
tampakan	1788.18	534.09		5s	5s	

In-Out Separation and Column Generation Stabilization by Dual Price Smoothing. Pessoa et al. 2013

Solving the Linear Programming relaxation

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Computers & Operations Research 36 (2009) 1064–1089

computers & operations research

www.elsevier.com/locate/cor

LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity

Natashia Boland[a], Irina Dumitrescu[b,*], Gary Froyland[b], Ambros M. Gleixner[c]

[a] Department of Mathematics and Statistics, The University of Melbourne, Parkville, VIC 3010, Australia
[b] School of Mathematics and Statistics, The University of New South Wales, Sydney, NSW 2052, Australia
[c] Institut für Mathematik, Technische Universität Berlin, D-10623 Berlin, Germany

Available online 14 December 2007

Abstract

Given a discretisation of an orebody as a block model, the open pit mining production scheduling problem (OPMPSP) consists of finding the sequence in which the blocks should be removed from the pit, over the lifetime of the mine, such that the net present value (NPV) of the operation is maximised. In practice, due to the large number of blocks and precedence constraints linking them, blocks are typically aggregated to form larger scheduling units. We aim to solve the OPMPSP, formulated as a mixed integer programme (MIP), so that aggregates are used to schedule the mining process, while individual blocks are used for processing decisions. We propose an iterative disaggregation method that refines the aggregates (with respect to processing) up to the point where the refined aggregates defined for processing produce the same optimal solution for the linear programming (LP) relaxation of the MIP as the optimal solution of the LP relaxation with individual block processing. We propose several strategies of creating refined aggregates for the MIP processing, using duality results and exploiting the problem structure. These refined aggregates allow the solution of very large problems in reasonable time with very high solution quality in terms of NPV.

A New LP Algorithm for Precedence Constrained Production Scheduling

Daniel Bienstock* Mark Zuckerberg†

August, 2009
Version Tues Aug 18 09:41:12 AEST 2009

Abstract

The precedence constrained production scheduling problem is the problem of scheduling the performance of jobs over a number of scheduling periods subject to precedence constraints among the jobs. The jobs can each be performed in a number of ways, and it also needs to be determined which processing option (or options) is to be chosen for each job. There can also be arbitrary side constraints among these variables. The side constraints typically represent either period capacity constraints, or profile constraints on the aggregate product produced in each period.

These problems, as they occur in the mining industry, typically have a small number of side constraints - often well under 100, but may contain millions of jobs and tens of millions of precedences. Thus despite the fact that the integrality gap is often small in practice, the LP itself is beyond the practical reach of commercial software.

We present a new iterative lagrangian-based algorithm for solving the LP relaxation of this problem. This algorithm can be proven to converge to optimality and in practice we have found that even for problems with millions of variables and tens of millions of constraints, convergence to proved optimality is usually obtained in under 20 iterations, with each iteration requiring only a few seconds to solve with current computer hardware.

Specialized Lagrangian-Based algorithm for solving precedence-constrained problems.

The BZ algorithm: idea

Master Problem (lower bound)

$$\max \quad c^t(v_o + V\lambda)$$

s.t.

$$A(v_o + V\lambda) \le b,$$

$$H(v_o + V\lambda) \le h, \quad (\mu \ge 0)$$

$[v_o, V]$ Can be almost anything

Ideally, spanned linear space should contain all past generated columns

Pricing Problem (upper bound)

$$\max \quad c^t v - \mu(Hv - h)$$

s.t.

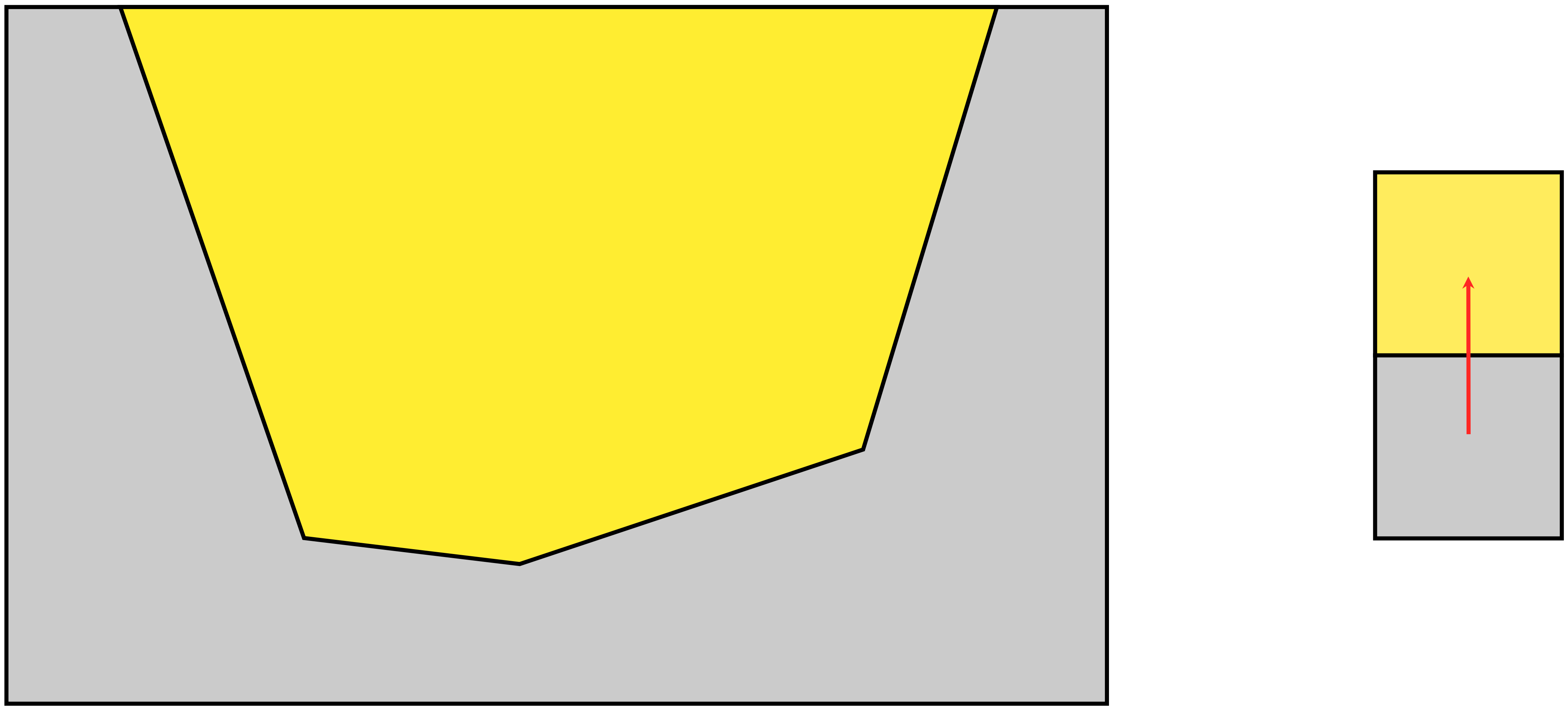
$$Av \le b$$

In the mining problem:

Pricing is max-closure problem

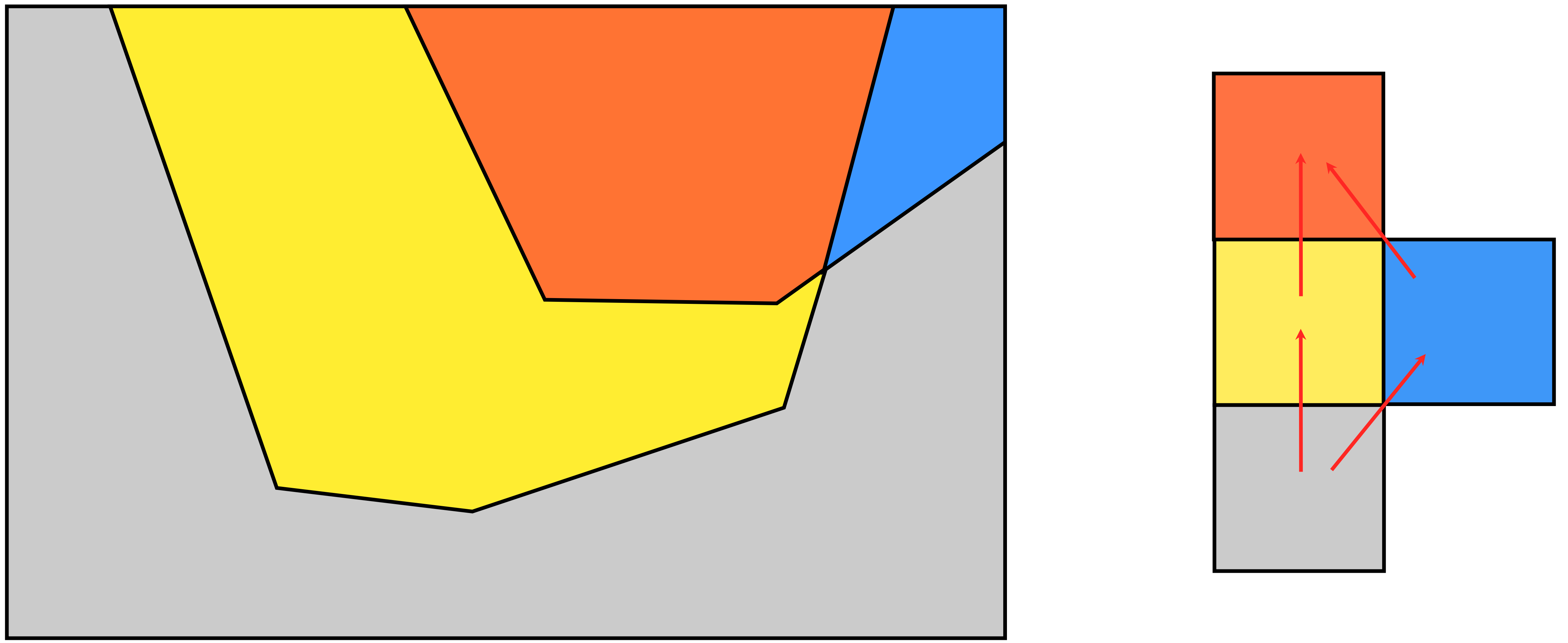
"Easy" to solve, using Hochbaum's Pseudoflow algorithm.

The BZ algorithm



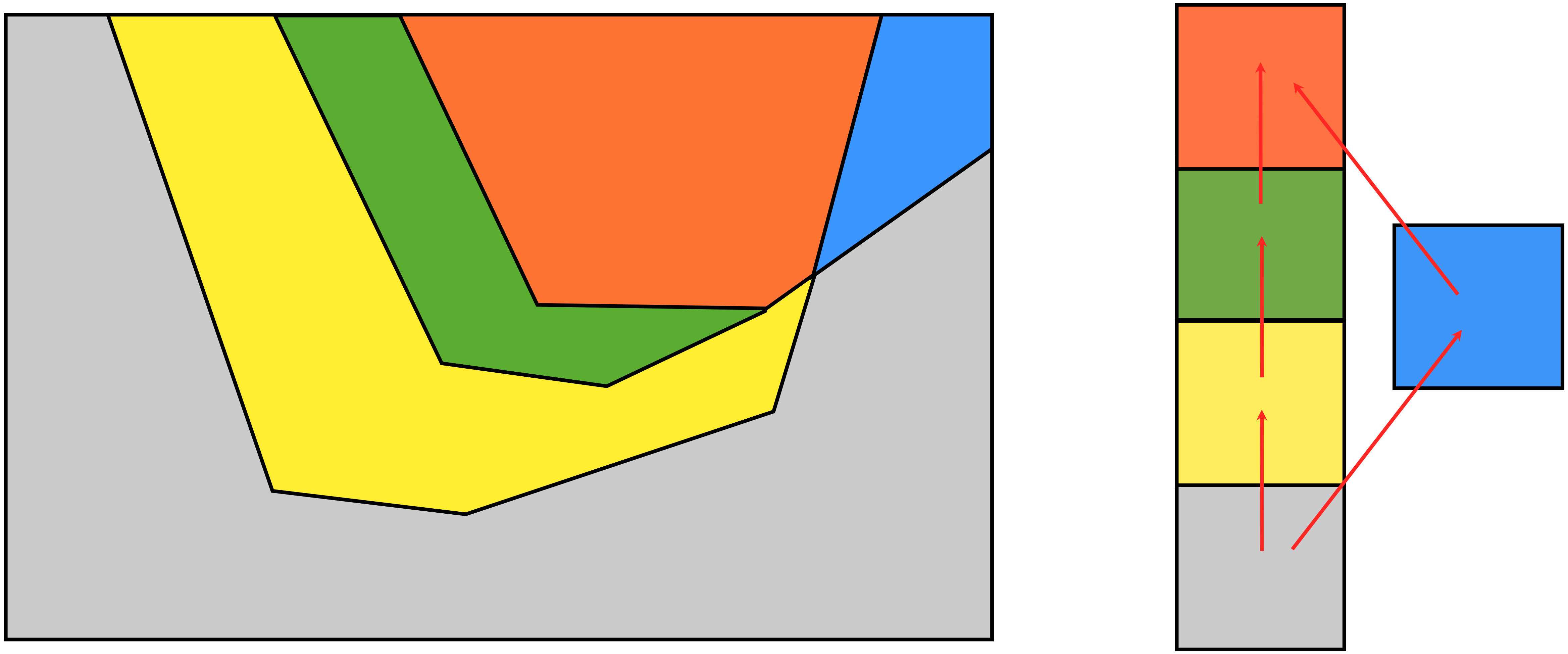
The BZ algorithm

The BZ algorithm

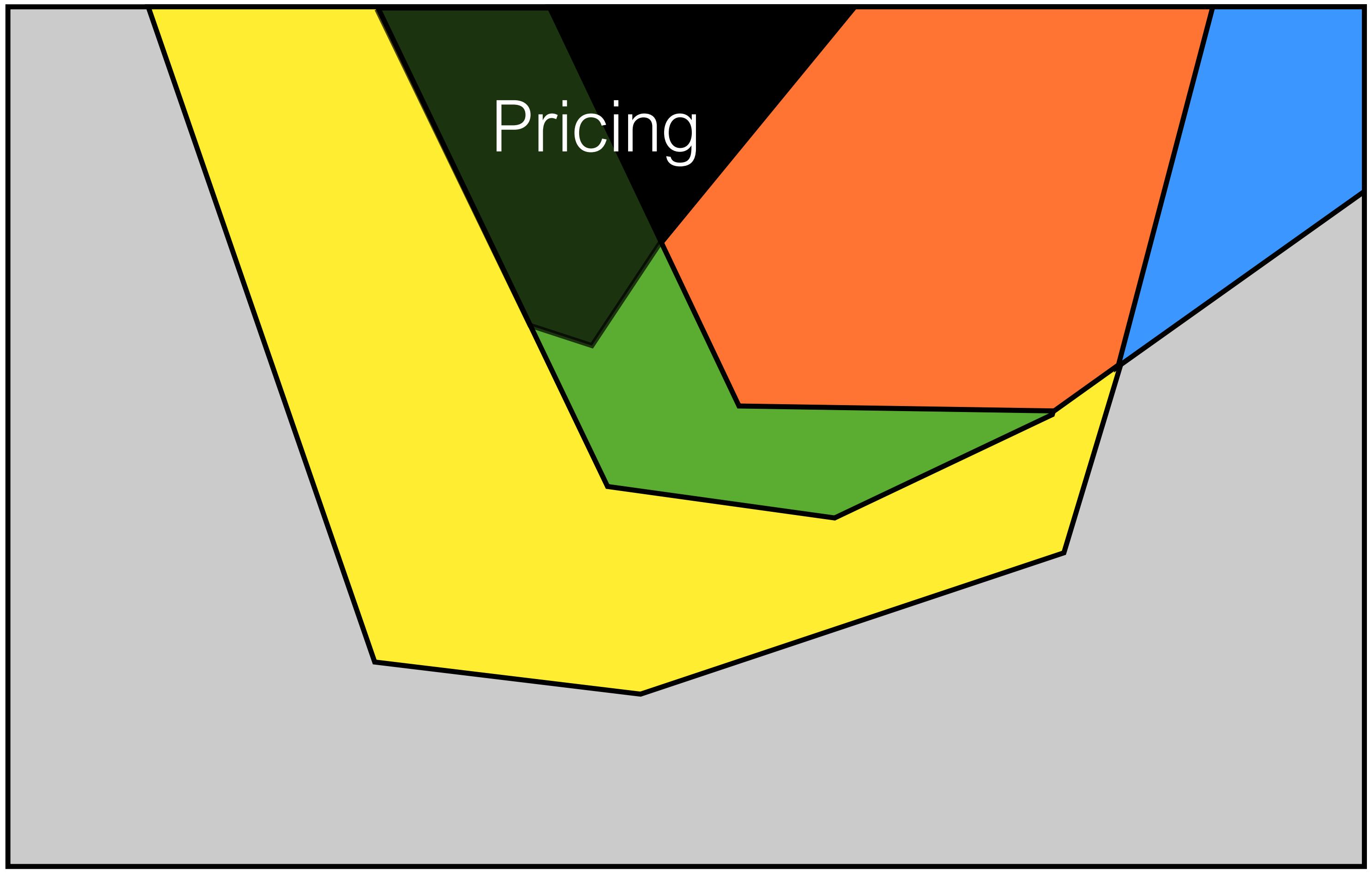


The BZ algorithm

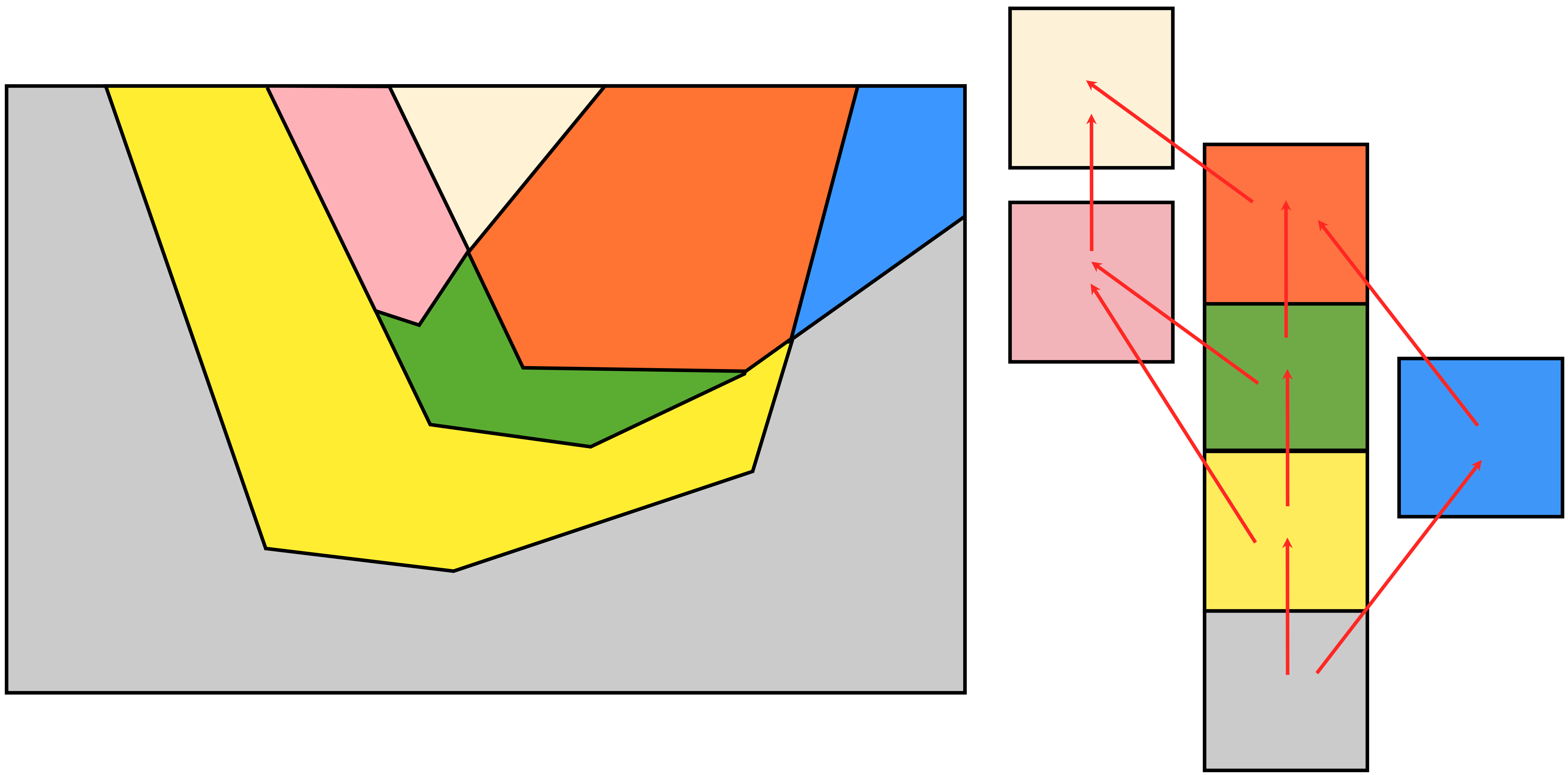
The BZ algorithm



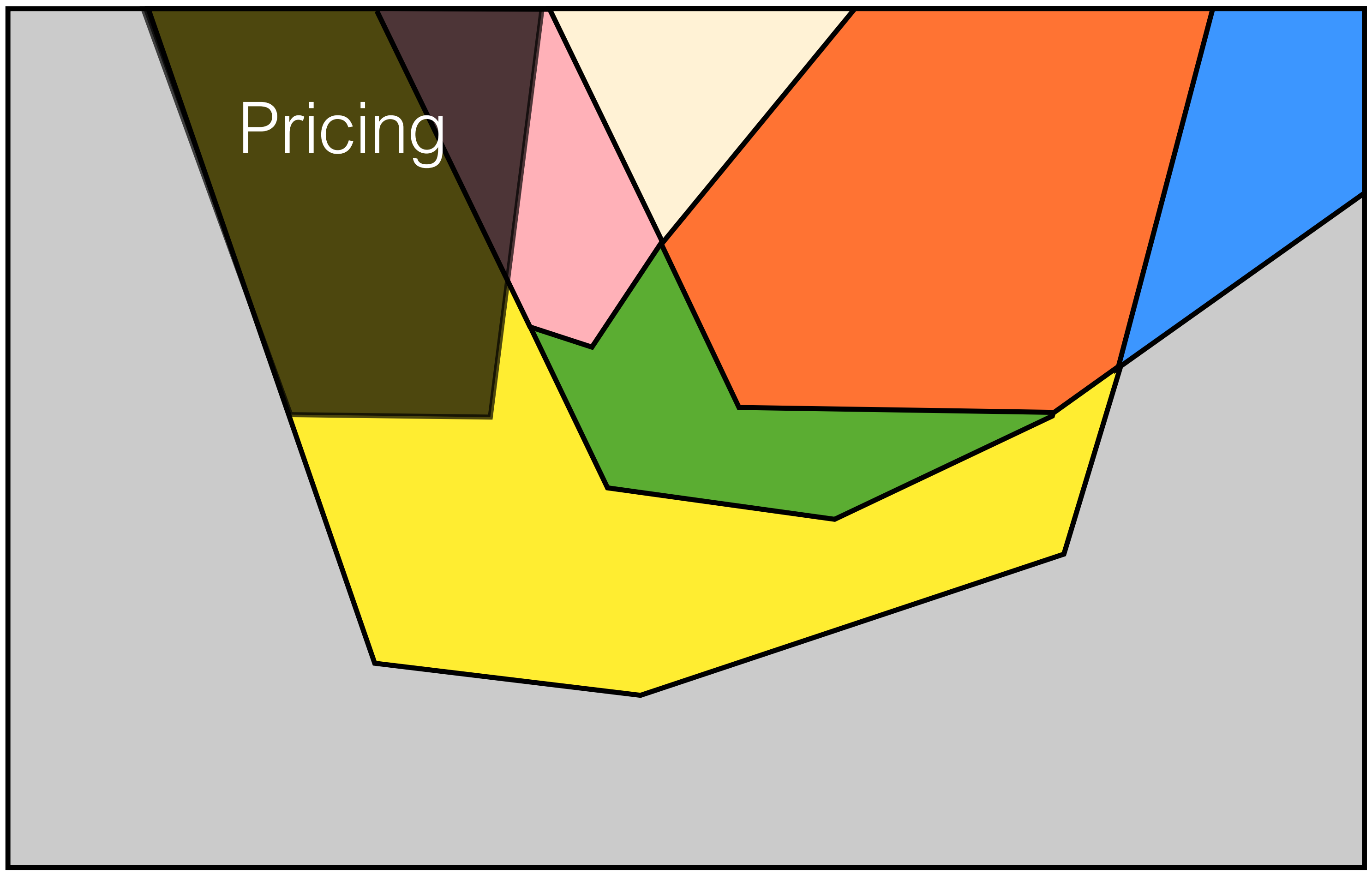
The BZ algorithm



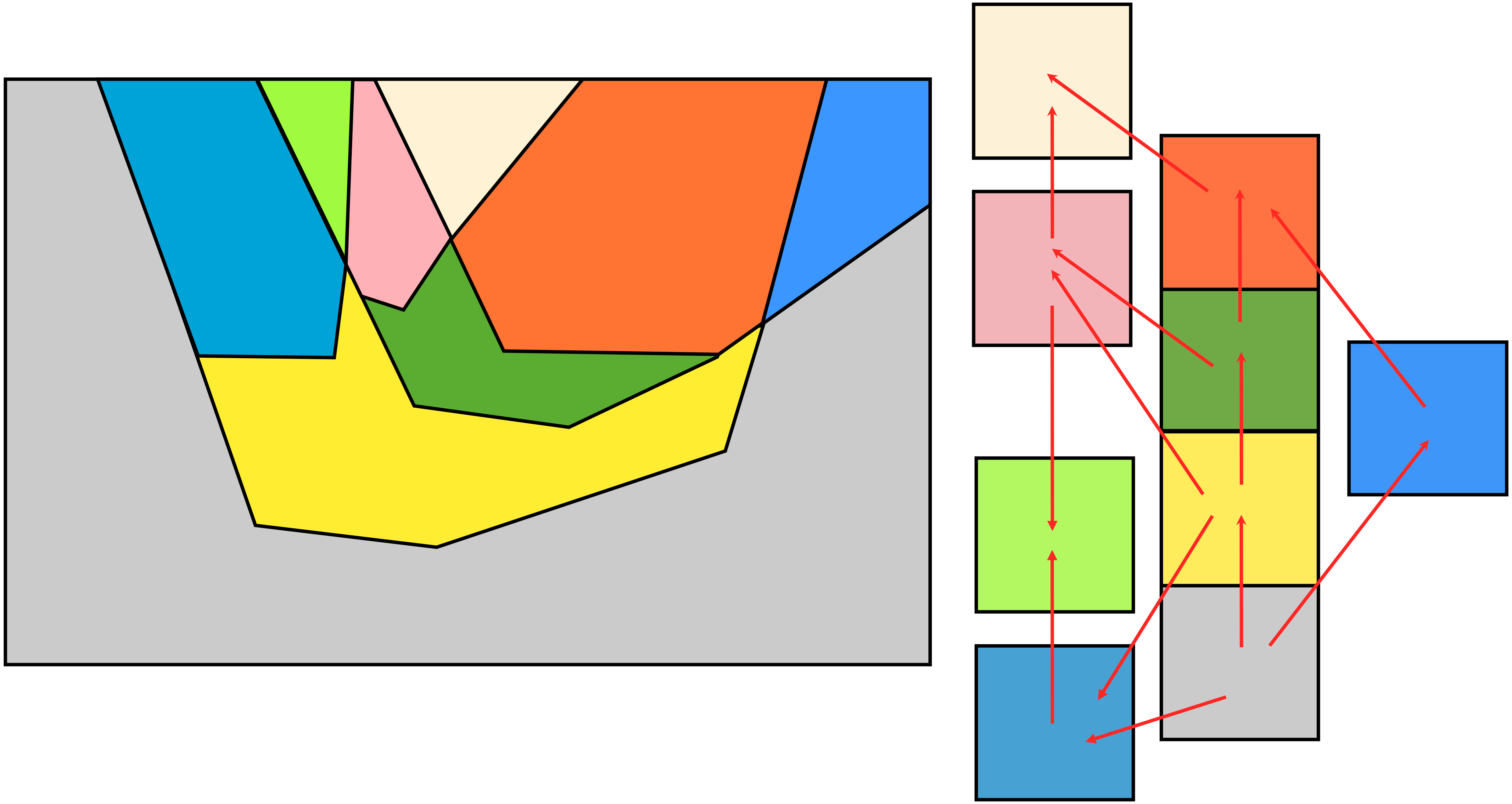
The BZ algorithm



The BZ algorithm



The BZ algorithm



Effectiveness of BZ algorithm

	Phase Design			Production Scheduling		
	DW	**DW+S**	**BZ**	**DW**	**DW+S**	**BZ**
calbuco	2h 28m	40m	11m 40s	8s	13s	10s
chaiten	9h 34m	2h 27m	26m 55s	15s	17s	10s
guallatari	733.94	315.06	1m 50s	5s	8s	4s
kd	11s	8s	2s	250ms	320ms	200ms
marvinml	14s	10s	3s	380ms	700ms	500ms
mclaughlin	48m	15m	4m 40s	2s	2s	2s
mclaughlinlimit	9m	4m 35s	1m 30s	1s	1s	1s
palomo	1h 30m	36m 42s	11m	3s	3s	3s
ranokau	3d 0h 42m	10h 52m	9h 39m	20m 30s	14m	8m 40s
tampakan	30m	8m 50s	2m 7s	5s	5s	3s

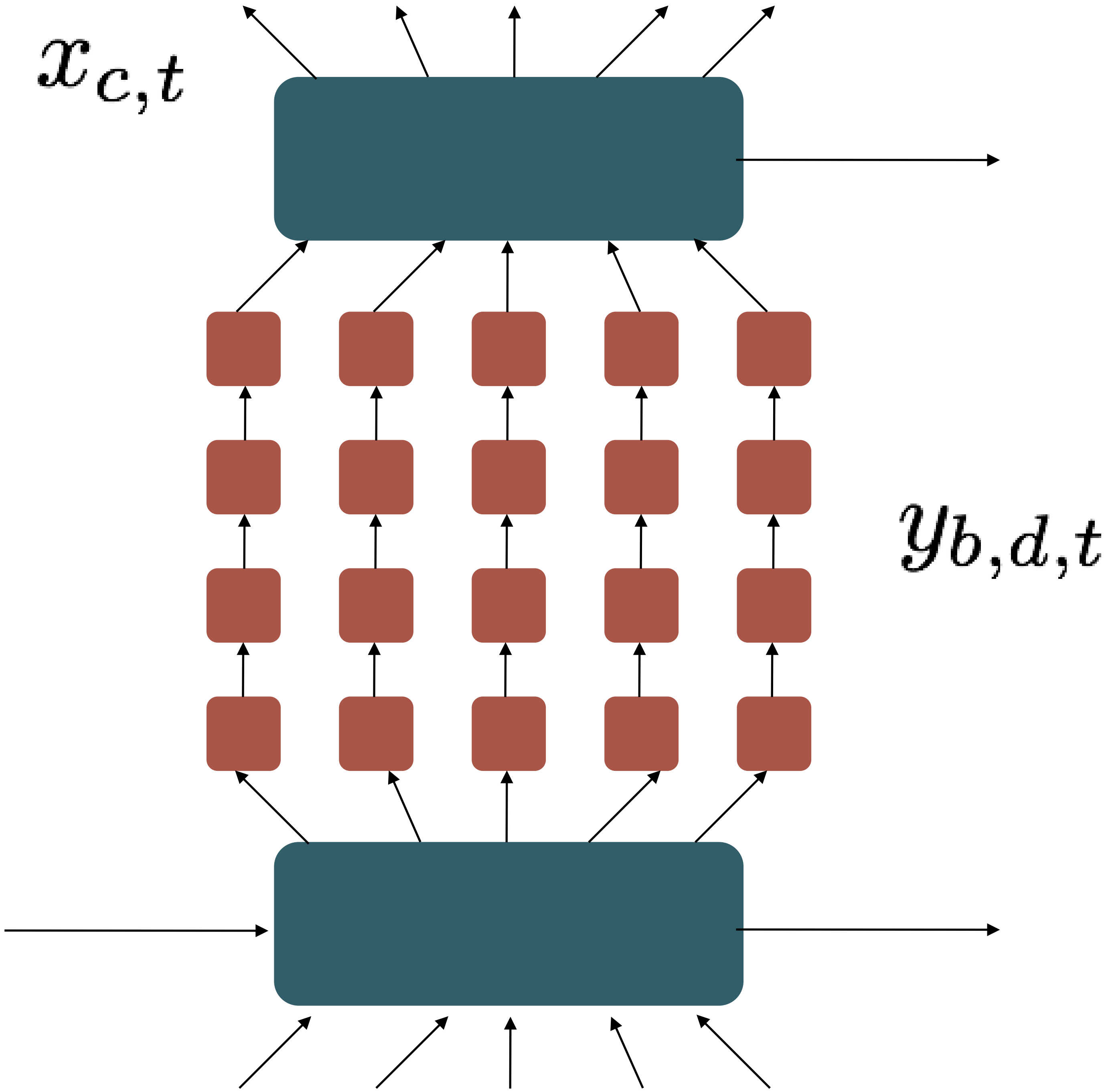
Speeding up the BZ algorithm:

$$\boxed{\begin{aligned}
&z_{b,d,t} \leq z_{b,d+1,t} \\
&z_{b,D,t} \leq z_{b,1,t+1} \\
&z_{b,D,T} \leq 1 \\
&z_{b,D,t} \leq z_{a,D,t} \\
&w_{c,t} = z_{b,D,t} \quad \forall b \in c \\
&Hz \leq h
\end{aligned}}$$

$$z_i \leq z_j \quad \forall (i, j)$$

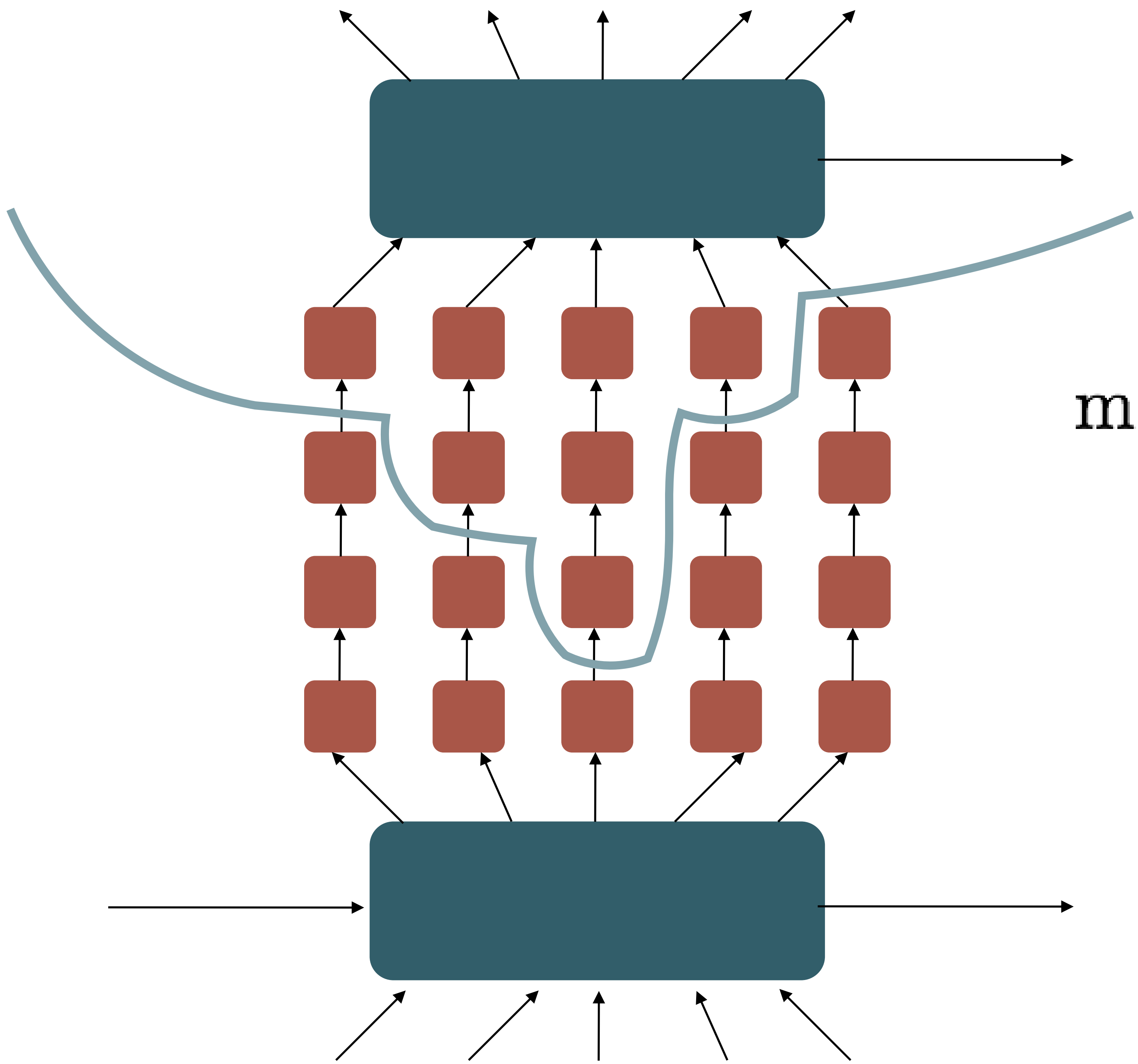
$z_{b,1,1}$	$z_{b,2,1}$	$z_{b,1,2}$	$z_{b,2,2}$	$z_{b,1,3}$	$z_{b,2,3}$
0	0	0	1	1	1

Common Structure in Production Scheduling



Sub-structure of precedence graph

A closure in the precedence graph



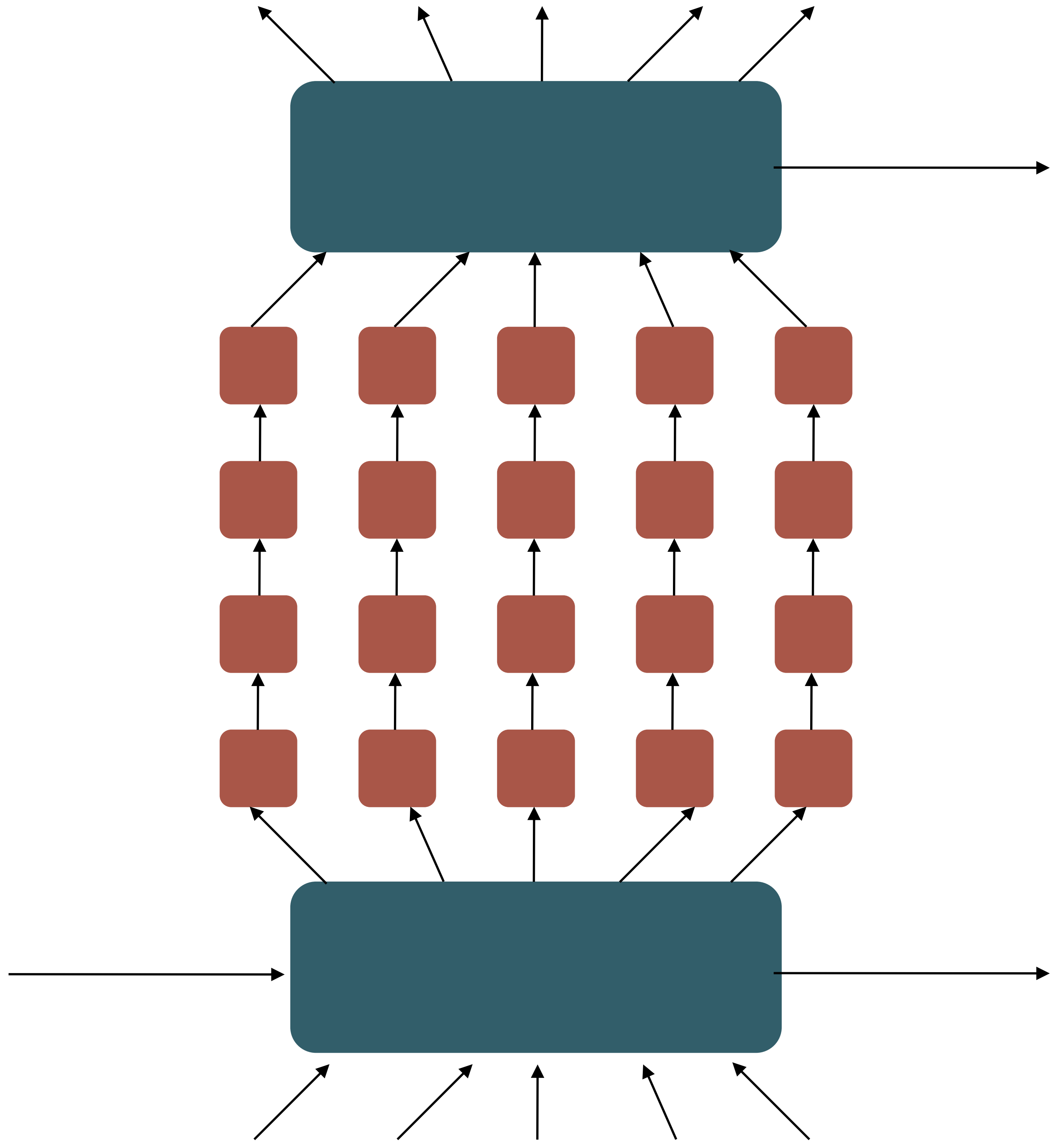
$$\max c^t z$$

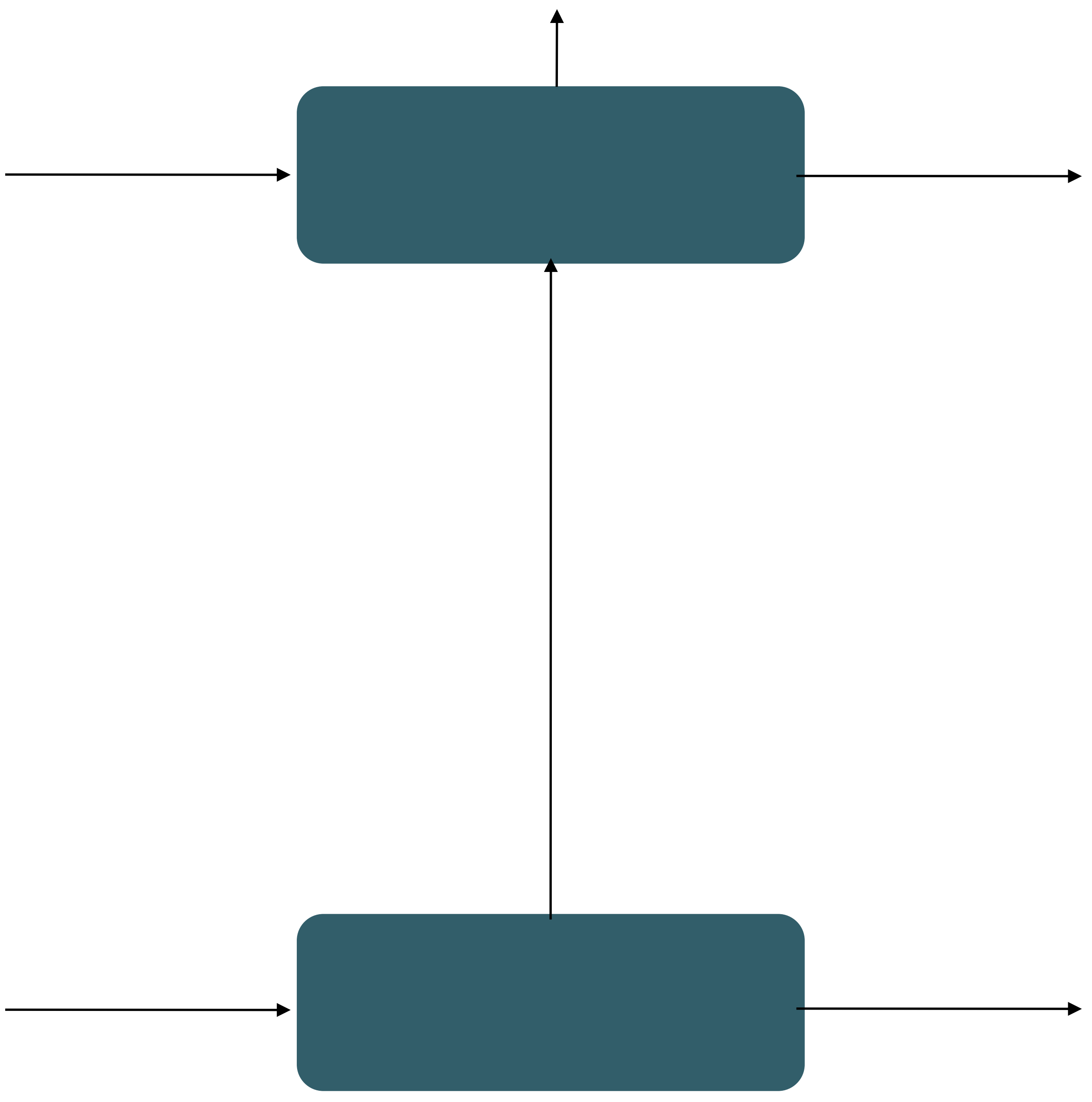
$$z_i \leq z_j \quad \forall (i,j) \in A$$

$$z_i \in \{0,1\}$$

Each path either does not touch the closure, or, is cut into two pieces

A significantly smaller graph





Sub-structure of precedence graph

Effect of path compression

	Number of Nodes	
	Before	**After**
calbuco	12,615,183	6,804
chaiten	14,403,650	6,825
guallatari	3,624,201	5,712
kd	243,072	636
marvinml	340,600	1,120
mclaughlin	7,229,960	3,460
mclaughlinlimit	3,323,040	2,490
palomo	15,225,520	2,960
ranokau	51,500,934	15,066
tronador	1,805,940	4,400

M., Espinoza, Goycoolea, Moreno, Queyranne, Rivera. COAP (2017).
A study of the Bienstock-Zuckerberg algorithm, Applications in Mining and Resource Constrained Project Scheduling.

Second ingredient: Cutting-Planes

(designed to exploit problem-specific structures)

Gap without adding any cuts

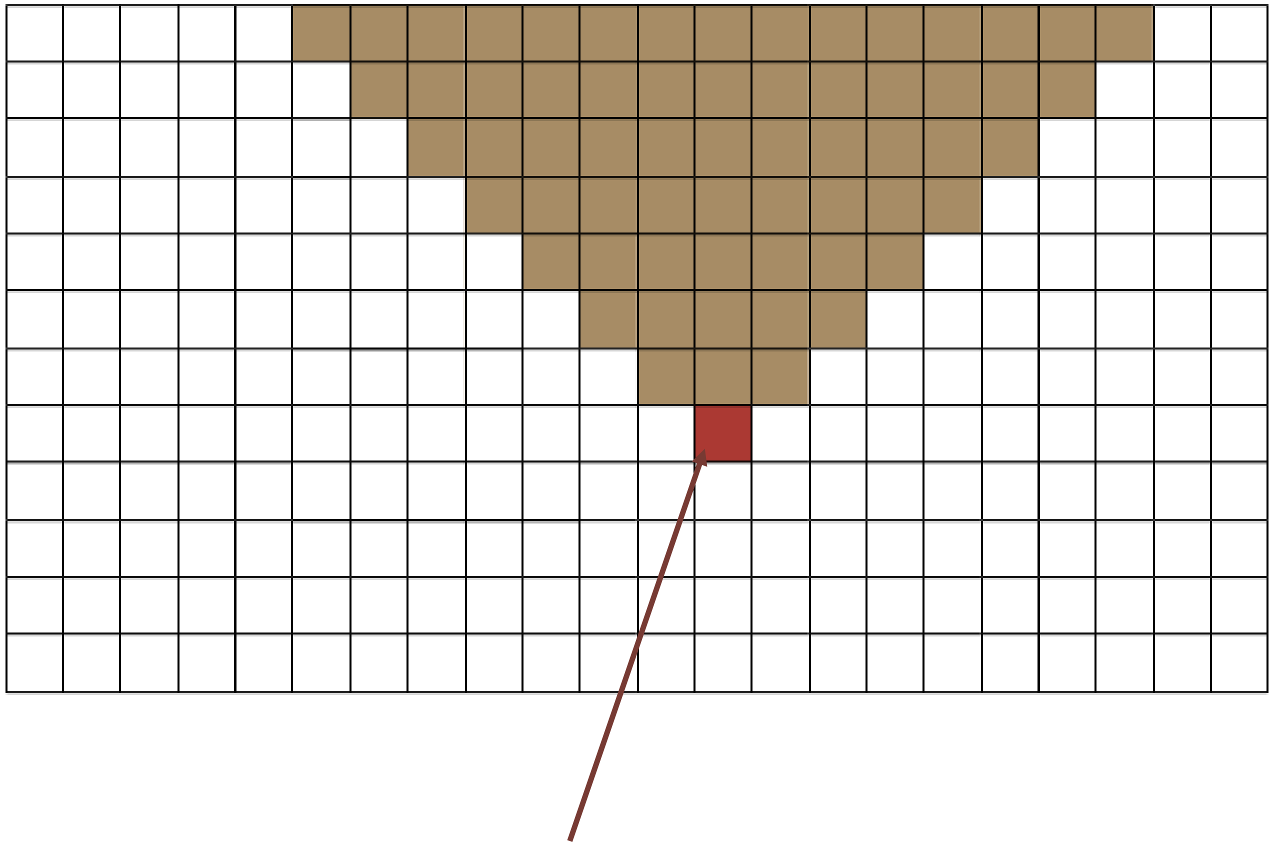
	Phase Design	Production Scheduling
	LP / Best	LP / Best
calbuco	102.06%	108.28%
chaiten	100.33%	117.26%
guallatari	101.22%	102.02%
kd	100.87%	101.75%
marvinml	102.49%	105.75%
mclaughlin	100.21%	102.52%
mclaughlinlimit	100.16%	102.39%
palomo	101.10%	114.87%
ranokau	102.22%	131.48%
tronador	102.47%	108.84%
Geo Mean	*101.31%*	*109.17%*

Gap relative to the best known lower bound (feasible solution)

Early-Start Cuts

(Gaupp (2008), Lambert et al. (2014) and many others)

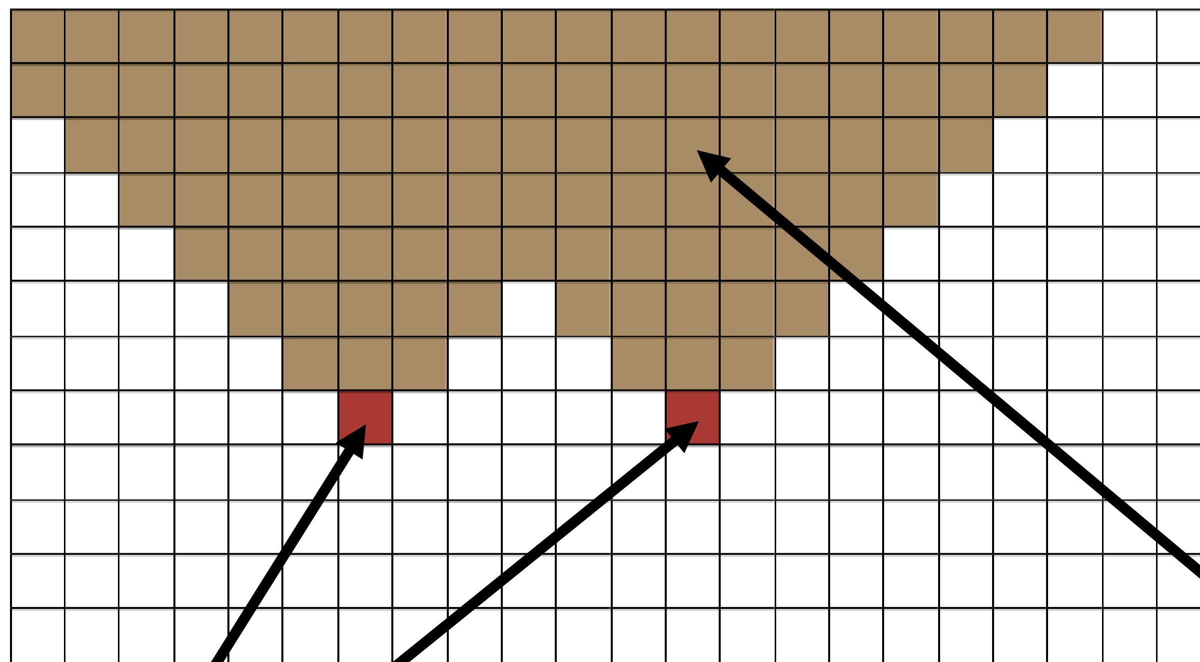
Classical variable elimination method



To extract this block, we need to extract everything above it. This results in an earliest possible extraction time for the block.

Clique Cuts

(proposed originally for the single knapsack case Boyd (1993))



This is more than can be extracted in time “t” or before [brown region].

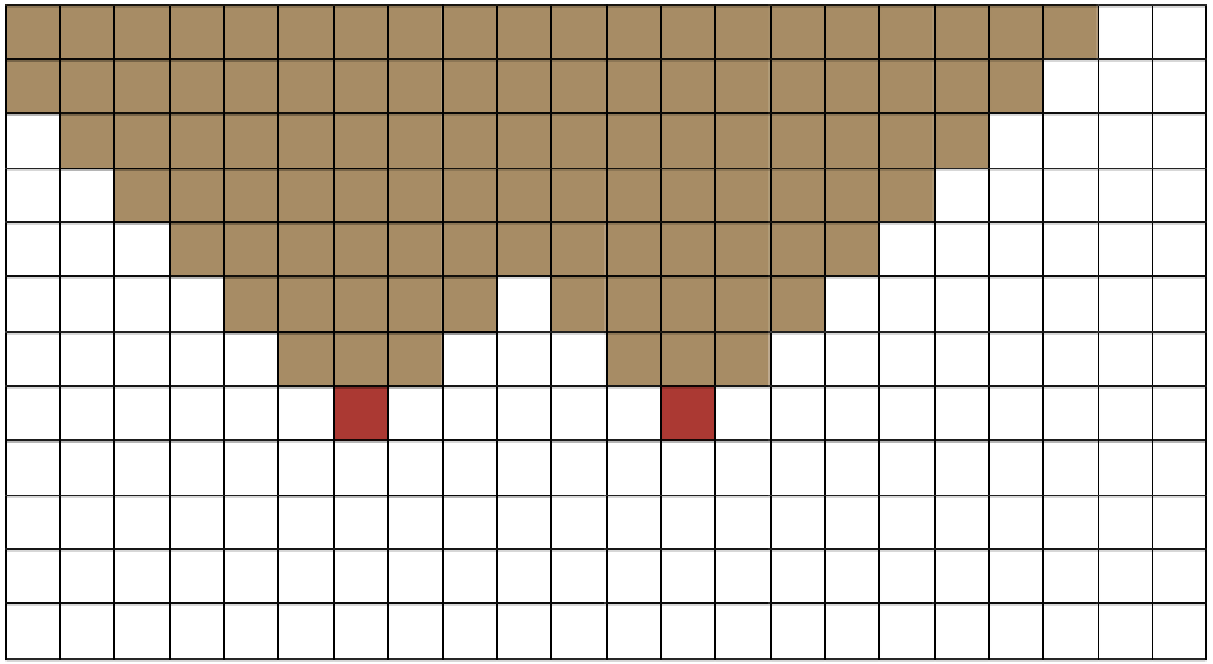
$$y_{b_1,t} + y_{b_2,t} \leq 1$$

If the sum is greater than one it is because all of the brown region was extracted in “t” or before.

$$y_{b_1,t} + y_{b_2,t} + y_{b_3,t} \leq 1$$

Clique Cuts

(proposed originally for the single knapsack case Boyd (1993))

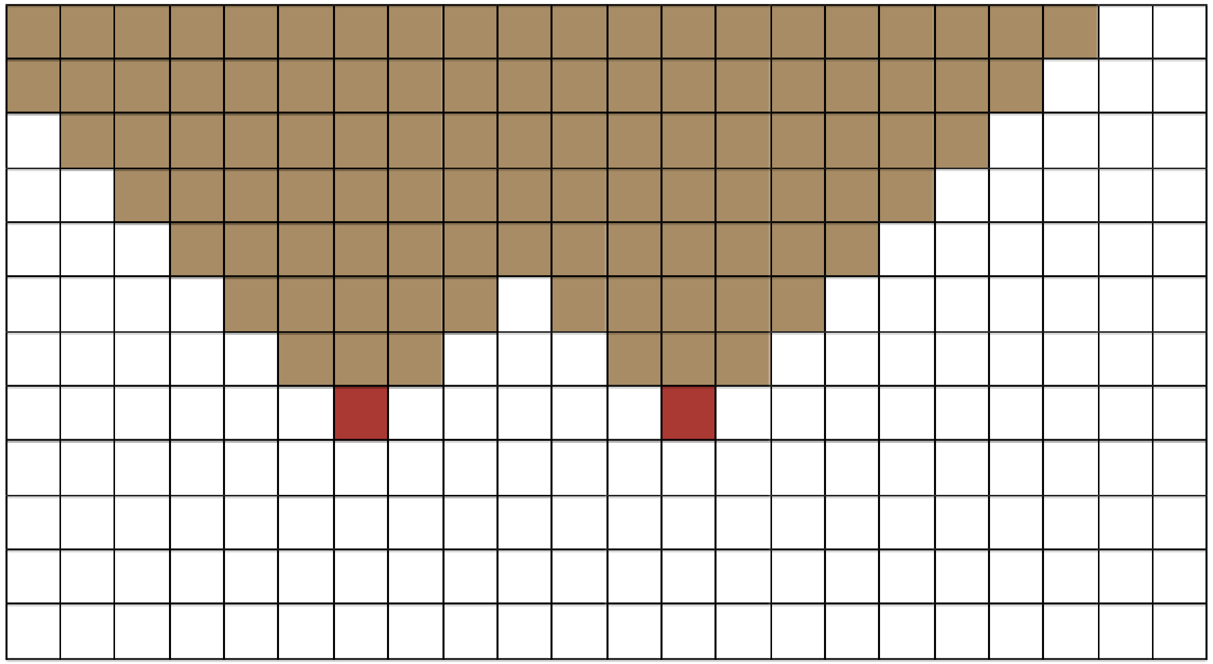


If c_1 and c_2 are such that: $$\sum_{c' \in cl(c_1) \cup cl(c_2)} q_{c'} \geq \sum_{t'=1}^{t} U_t$$

$$\sum_{t'=1}^{t} (x_{c_1,t'} + x_{c_2,t'}) \leq 1$$ is valid

Clique Cuts

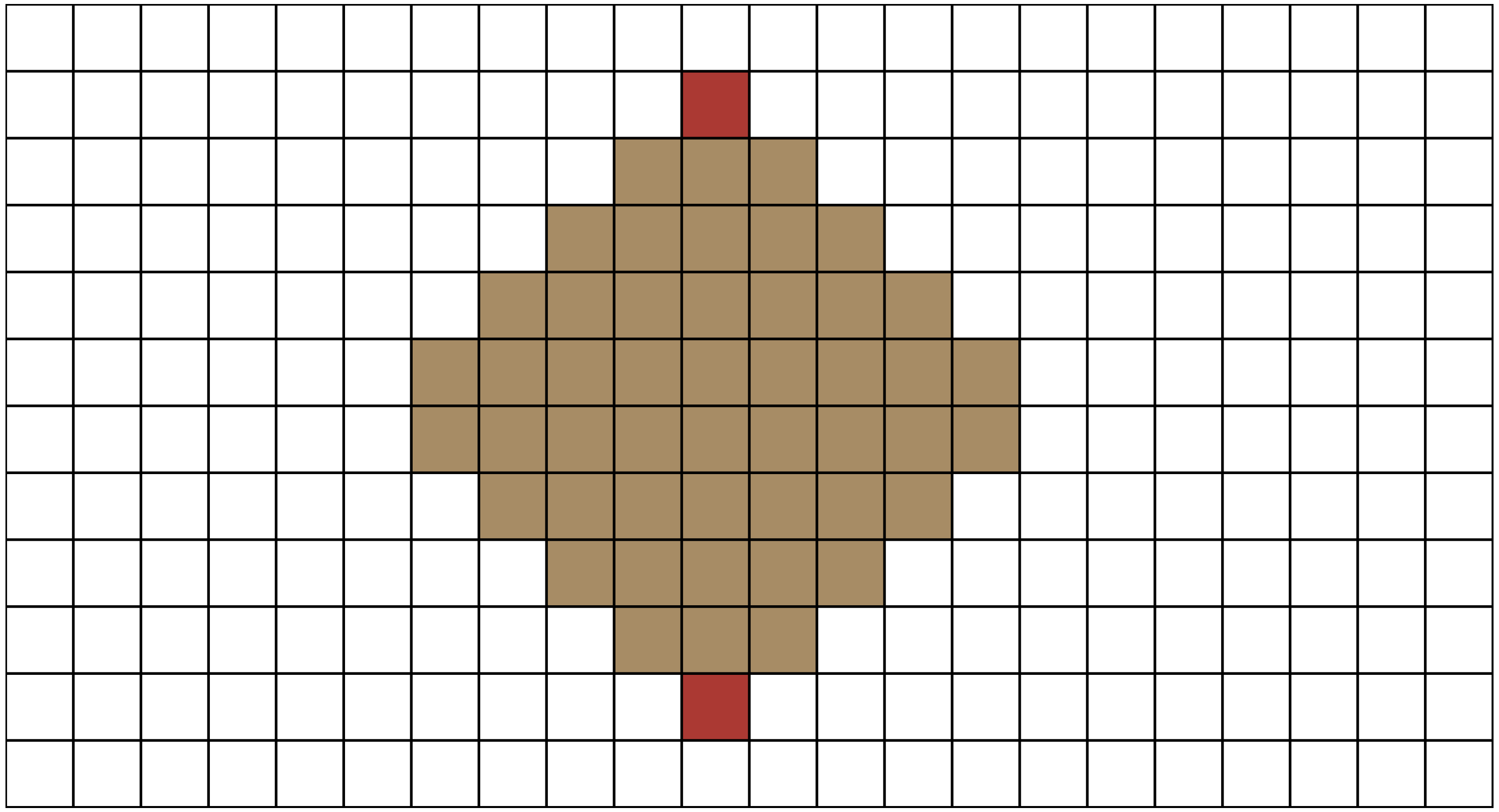
(proposed originally for the single knapsack case Boyd (1993))



The inequalities can be easily generalized to any group of clusters $c_1, c_2, \ldots, c_k$

Diamond Cuts

(similar to Zhu et al. (2006) for resource constrained scheduling)



The intersection of *closure* and *reverse closure* of two clusters induce a “lag” between their extraction

Gap after adding Extraction cuts

	No Cuts	E. Cuts
calbuco	108.28%	108.28%
chaiten	117.26%	100.88%
guallatari	102.02%	100.87%
kd	101.75%	101.75%
marvinml	105.75%	103.06%
mclaughlin	102.52%	102.52%
mclaughlinlimit	102.39%	102.39%
palomo	114.87%	111.37%
ranokau	131.48%	104.96%
tronador	108.84%	100.90%
Geo. Mean	*109.17%*	*103.65%*

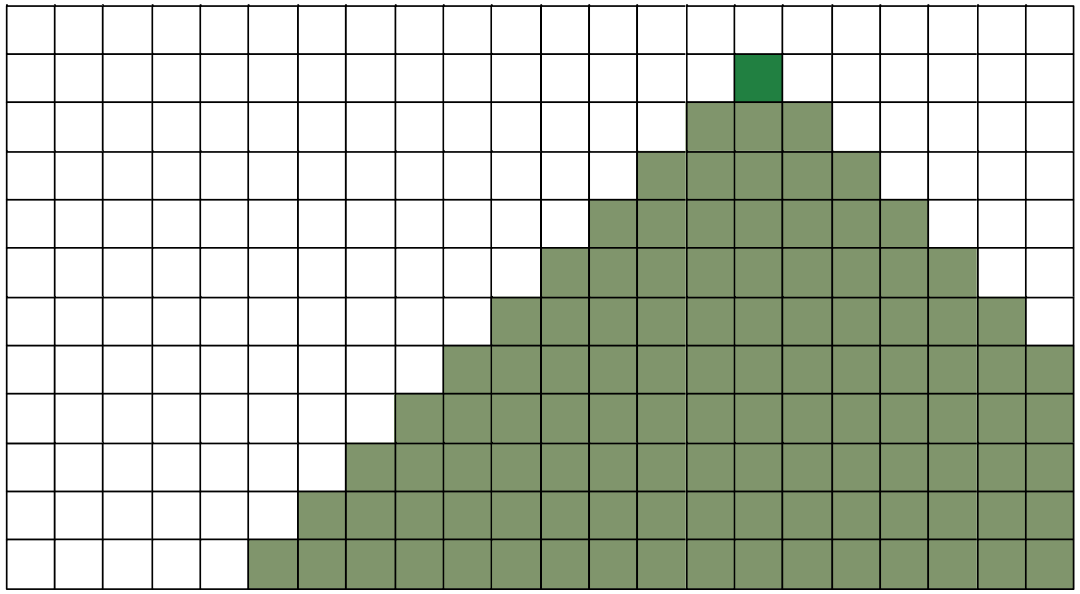
Gap relative to the best known lower bound (feasible solution)

VRHS Cuts

(combines precedences and production capacities)

The following inequality is always valid:

$$\sum_{b \in b(rcl(c))} q_b y_{b,d,t} \leq U_d^t \sum_{\tau=1}^{t} x_{c,\tau}.$$

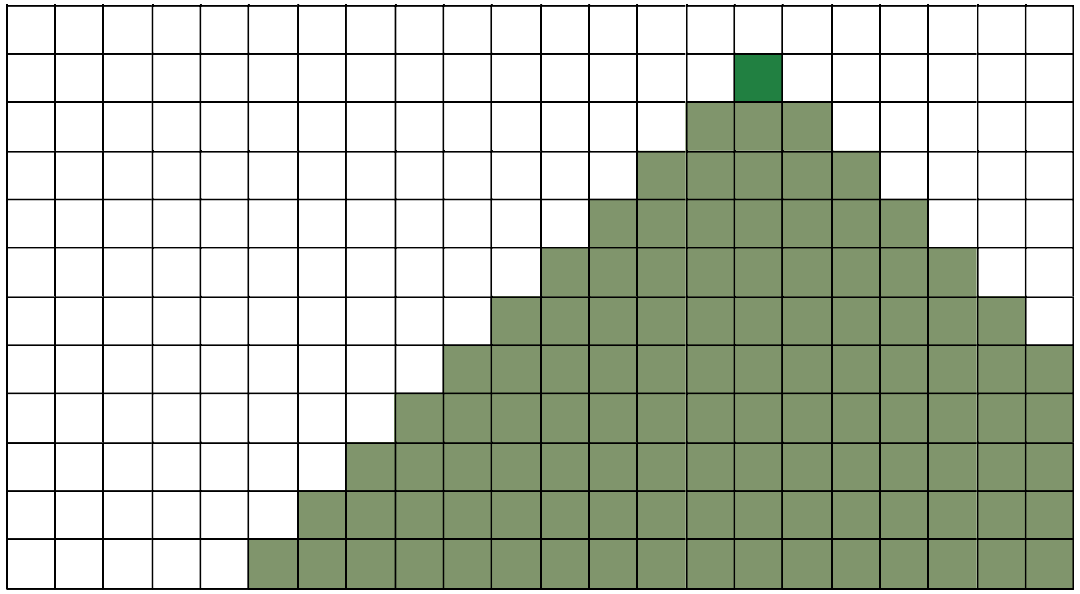


VRHS Cuts

(combines precedences and production capacities)

Its most general version:

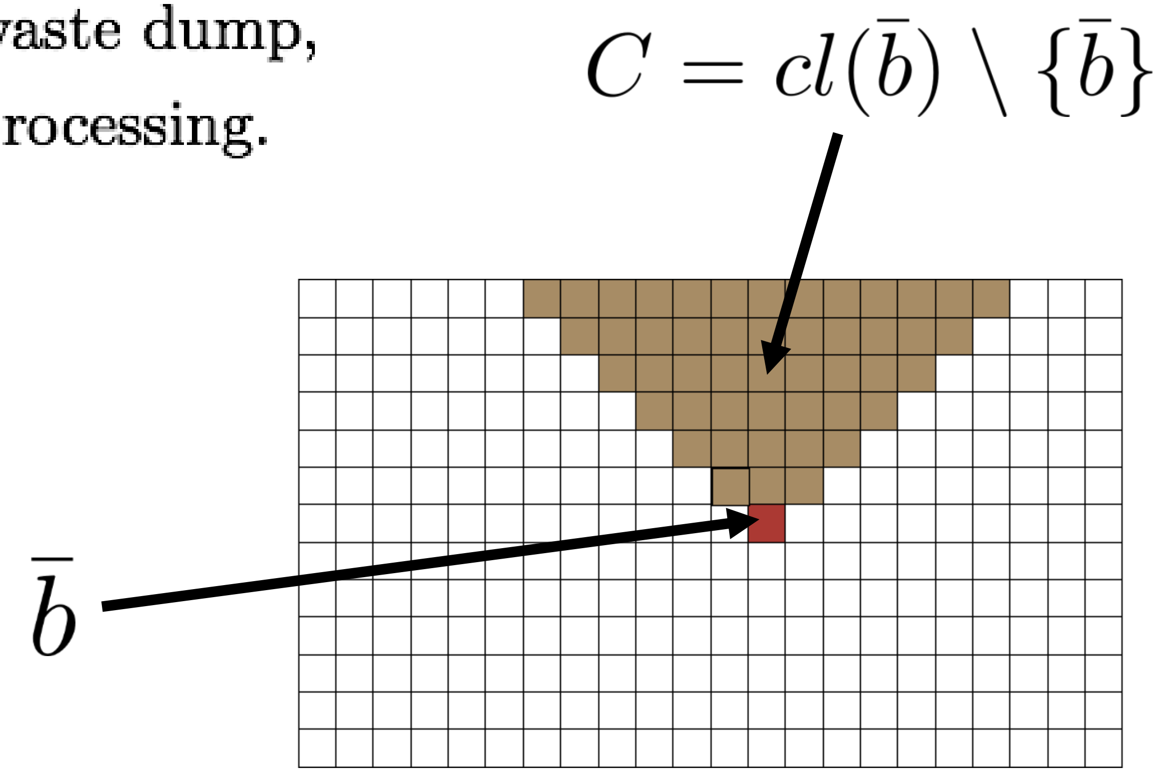
$$\sum_{k=1}^{n-1}\left(\sum_{c\in\Delta_k}\sum_{b\in c} q_b y_{b,d,t}\right) + \sum_{b\in c_n} \alpha_b q_b y_{b,d,t} + \sum_{c\in rcl(c_n)\setminus\{c_n\}} \sum_{b\in c} q_b y_{b,d,t} \leq \sum_{k=1}^{n} \delta_k w_{c_k,t}$$



Hour-Glass Cuts

for each block $b \in B$:

$$\begin{aligned} x_b &= \text{ proportion of block } b \text{ that is extracted,} \\ y_{b,w} &= \text{ proportion of block } b \text{ that is sent to waste dump,} \\ y_{b,p} &= \text{ proportion of block } b \text{ that is sent to processing.} \\ q_b &= \text{ tonnage of block } b. \end{aligned}$$



constraints :

$$\sum_{b \in B} q_b y_{b,p} \leq U$$

$$x_b = y_{b,w} + y_{p,w} \quad \forall b \in B$$

$$x_{\bar{b}} > 0 \Rightarrow x_b = 1 \quad \forall b \in C = cl(\bar{b}) \setminus \{\bar{b}\}$$

assume $q(C) > U$

then

$$x_{\bar{b}}(q(C) + q_{\bar{b}} - U) \leq \sum_{b \in C} q_b y_{b,w}$$

Hour-Glass Cuts

for each block $b \in B$:

$$x_b = \text{ proportion of block } b \text{ that is extracted,}$$
$$y_{b,w} = \text{ proportion of block } b \text{ that is sent to waste dump,}$$
$$y_{b,p} = \text{ proportion of block } b \text{ that is sent to processing.}$$
$$q_b = \text{ tonnage of block } b.$$

constraints :

$$\sum_{b \in B} q_b y_{b,p} \leq U$$

$$x_b = y_{b,w} + y_{p,w} \quad \forall b \in B$$

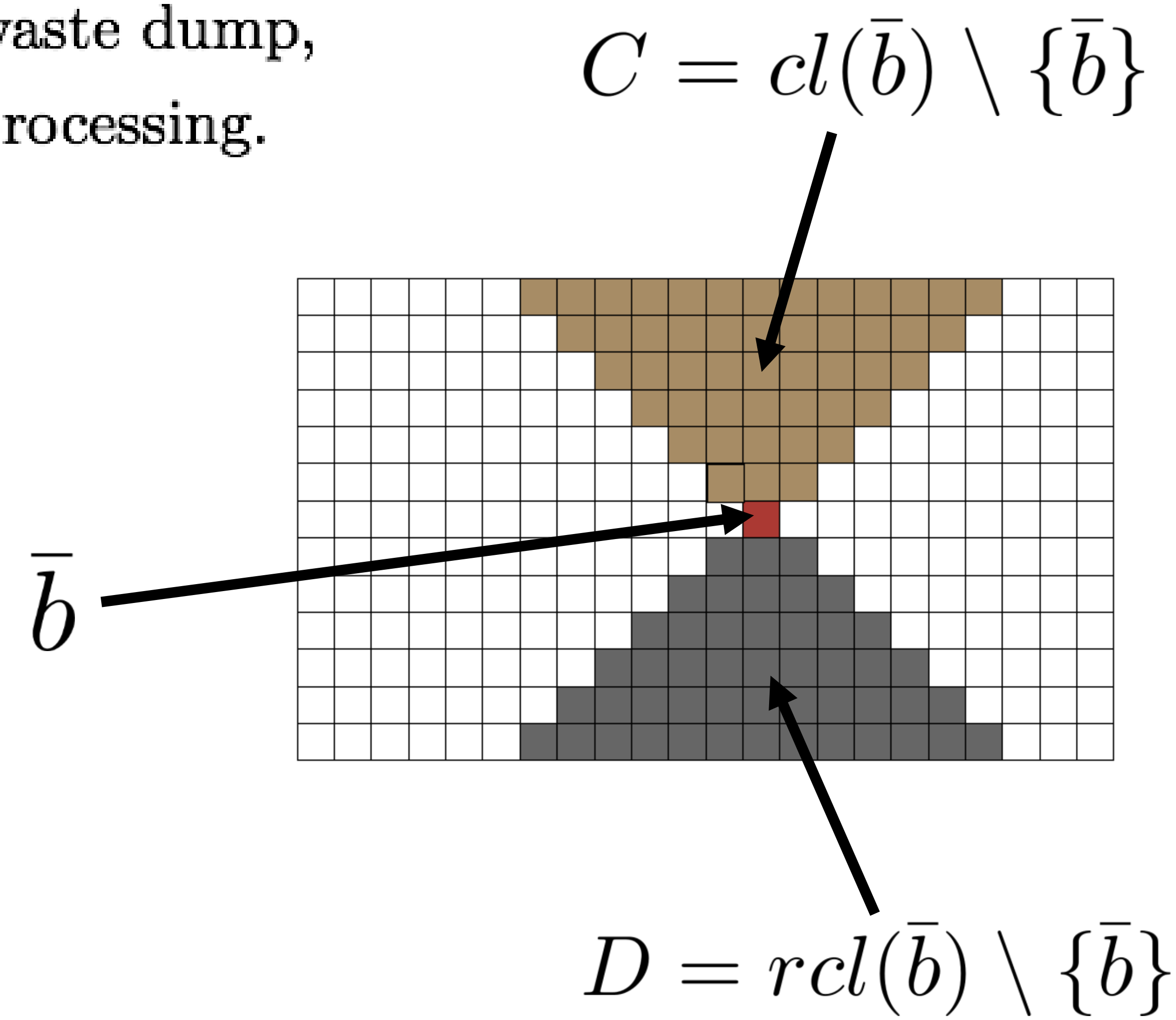
$$x_{\bar{b}} > 0 \Rightarrow x_b = 1 \quad \forall b \in C = cl(\bar{b}) \setminus \{\bar{b}\}$$

$$x_{\bar{b}} < 1 \Rightarrow x_b = 0 \quad \forall b \in D = rcl(\bar{b}) \setminus \{\bar{b}\}$$

assume $q(C) > U$

then

$$\sum_{b \in D} q_b y_{b,p} + x_{\bar{b}}(q(C) + q_{\bar{b}} - U) \leq \sum_{b \in C} q_b y_{b,w}$$



Gap after adding different classes of cuts

	No Cuts	E. Cuts	P. Cuts	All Cuts
calbuco	108.28%	108.28%	102.42%	102.42%
chaiten	117.26%	100.88%	109.23%	100.00%
guallatari	102.02%	100.87%	101.09%	100.54%
kd	101.75%	101.75%	100.21%	100.21%
marvinml	105.75%	103.06%	101.10%	100.61%
mclaughlin	102.52%	102.52%	100.34%	100.34%
mclaughlinlimit	102.39%	102.39%	100.25%	100.25%
palomo	114.87%	111.37%	103.62%	101.26%
ranokau	131.48%	104.96%	105.20%	101.82%
tronador	108.84%	100.90%	104.00%	100.80%
Geo. Mean	*109.17%*	*103.65%*	*102.71%*	*100.82%*

Gap relative to the best known lower bound (feasible solution)

Rivera, Espinoza, Goycoolea, Moreno, M., Submitted (2018). Available upon request.

Third ingredient: Heuristics

TopoSort Heuristic

(uses LP solution to guide a greedy algorithm)

$$\sum_{t \in \mathcal{T}} x_{c,t} \leq 1$$

Interpret x as “probability”

$$E[c] = \left(\sum_{t=1}^{T} t x^*_{c,t} \right) + (T+1) \left(1 - \sum_{t=1}^{T} t x^*_{c,t} \right)$$

Expected extraction time

Topologically sort the clusters, and break-ties using this weight.

1-Dest Heuristic

- If blending is present, TopoSort might output an infeasible schedule.
- As an alternative, for Production Scheduling, we use the LP solution to fix destinations and then use a MIP solver on the reduced instance

Computational Results

Effectiveness of the Overall Approach

(Phase Design)

	Gap	Time
calbuco	2.06%	13m 5s
chaiten	0.33%	26m 55s
guallatari	1.22%	2m 16s
kd	0.87%	2.8s
marvinml	2.49%	4.5s
mclaughlin	0.21%	4m 55s
mclaughlinlimit	0.16%	1m 36s
palomo	1.10%	12m 12s
ranokau	2.22%	9h 39m 13 s
tronador	2.47%	3m 13s
Geo Mean	*1.31%*	

Effectiveness of the Overall Approach

(Production Scheduling)

	Root	BB4
calbuco	2.70%	2.37%
chaiten	0.00%	0.00%
guallatari	0.63%	0.36%
kd	0.26%	0.00%
marvinml	0.71%	0.00%
mclaughlin	0.66%	0.41%
mclaughlinlimit	0.37%	0.01%
palomo	2.43%	1.33%
ranokau	2.06%	2.06%
tronador	0.80%	0.32%
Geo. Mean	*1.06%*	*0.68%*

Final GAP for Production Scheduling Problem,
obtained combining heuristics, cuts, and branching.

Effectiveness of the Overall Approach

(Production Scheduling)

	LP (BZ)	LP + Cuts	1-Dest	BB4
calbuco	10s	4m 42.9s	1m 3.9s	> 4h
chaiten	9.9s	1m 26.4s	5m 41.4s	8.1s
guallatari	3.5s	23.4s	5m 26.7s	> 4h
kd	0.2s	0.9s	0.7s	38.5s
marvinml	0.4s	2s	2.4s	15m
mclaughlin	2.1s	12.4s	6.3s	> 4h
mclaughlinlimit	1.1s	5.2s	5.9s	2h 19m
palomo	3.4s	29.6s	21.7s	> 4h
ranokau	9m 19.8s	6m 12.6s	13m 9.3s	> 4h
tronador	2.9s	9.8s	17.5s	> 4h

C implementation, CPLEX 12.6, Linux 2.6.32 x86 64, four 8-core Intel R Xeon R E5-2670 processors and 128 Gb of RAM

Effectiveness of the Overall Approach

Our instances also include versions with:

- Minimum processing constraints
- Flow balance constraints
  (production cannot change drastically)
- Blending

The methodology shows the same behaviour

Final thoughts

- Mine Planning is a challenging problem that is becoming *tractable* thanks to the community of researchers

- Combining new and old techniques we can obtain *optimality guarantees* in moderate times in the deterministic setting

- Current efforts are being made to successfully include stockpiling and better connectivity constraints

- We hope this can be used as a building block in more ambitious problems such as *Stochastic Integer Programming* models

Thank you!