Open-pit mining problem and the BZ algorithm

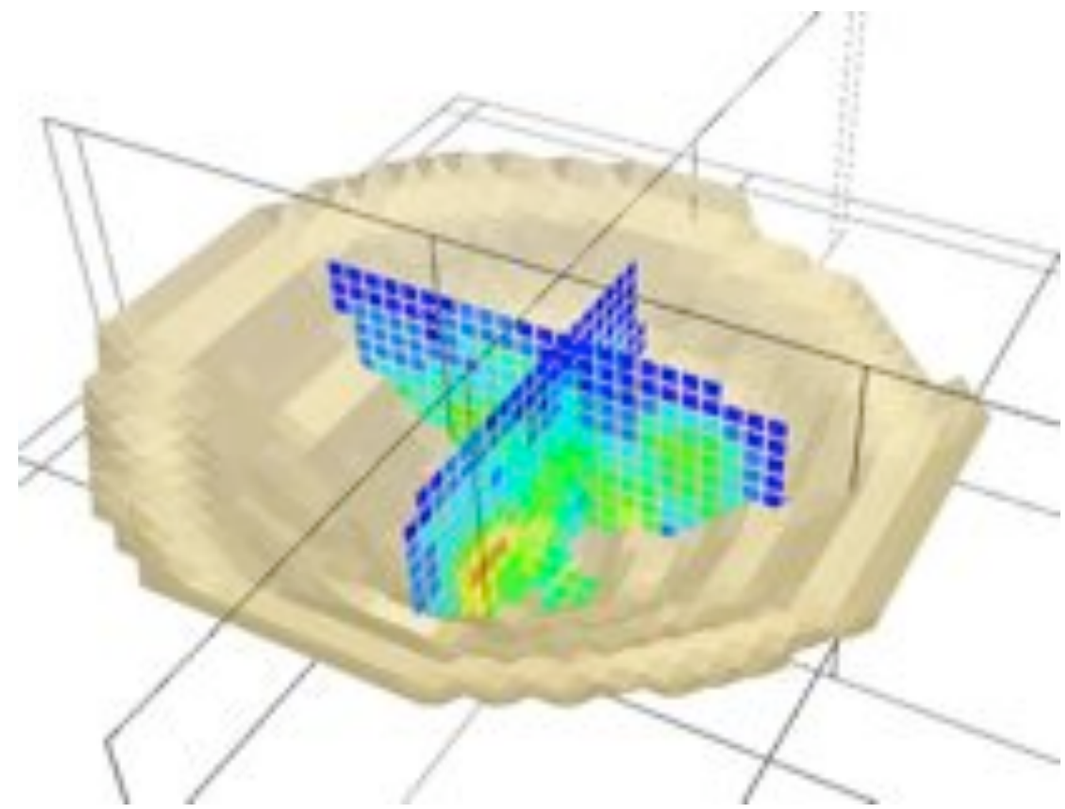
Eduardo Moreno (Universidad Adolfo Ibañez)
Daniel Espinoza (Universidad de Chile)
Marcos Goycoolea (Universidad Adolfo Ibañez)
Gonzalo Muñoz (Ph.D. student Columbia Univ.)

Outline

- The Open-Pit Mine Scheduling Problem
- BZ algorithm
- The General Algorithm Template
- Uncertainty in Open-pite mining
- BZ Algorithm & Stochastic Programming

Open pit mining scheduling problem

Which block should be extracted, when should they be extracted and how should they be processed



First integer programming model due to Thys Johnson ('69)

The problem

- Assumptions & Notation
    - blocks $\mathcal{B}$
    - T periods
    - Precedences $G = (\mathcal{B}, \mathcal{A})$
    - “By” decision variables $x_{b,t}$
    - Side constraints $Dx \leq d$
    - Discount rate $\hat{c}_{b,t} = \alpha^t c_b$

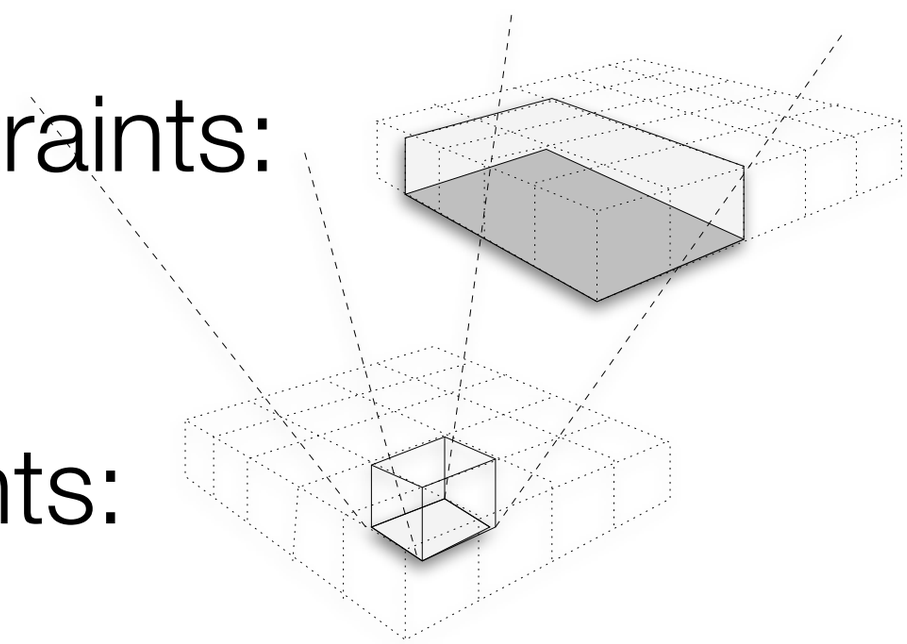
The problem

Time-Consistency constraints:

$$x_{b,t} \leq x_{b,t+1}$$

Precedence (wall-slope) constraints:

$$x_{a,t} \leq x_{b,t}$$



Capacity (knapsack) constraints:

$$\sum_{b \in B} a_b (x_{b,t} - x_{b,t-1}) \leq C_t$$

Multi-period precedence constrained knapsack

The problem

The “by” formulation

$$
\begin{array}{llll}
(PCP^{by}) & \max & cx & \\
 & & x_{b,t} \leq x_{b,t+1}, & \forall\, t < T-1 \\
 & & x_{a,t} \leq x_{b,t}, & \forall (a,b) \in \mathcal{A}, \forall t \\
 & & Dx \leq d & \\
 & & x_{b,t} \in [0,1] &
\end{array}
$$

Computational results (LP Relaxation)

We count with several REAL instances of this problem.
Most of them can't be solved using CPLEX 12.1
Intel(R) Xeon 2.33 GHz, 16 GB RAM.

Mine	Blocks	Prec.	Periods	CPX Time
Tinkerbell	1060	3922	6	8.44
PeterPan	9400	145640	20	327359.74
Pinochio	14153	219778	12	95631.97
Mowgli	16292	74260	20	29785.6
Hansel	29277	1271207	15	MEM!
Ariel	53271	656411	20	MEM!
SnowWhite	57958	1186665	21	MEM!
Grettel	96821	1053105	30	MEM!
Simba	99014	96642	30	5700.51
Cinderella-Limit	112687	3035483	15	MEM!
Dorothy	167937	5956121	51	MEM!
Rapunzel	304977	15665274	81	MEM!
Beauty	326428	1518326	80	MEM!
Aladdin	329859	7066518	60	MEM!
Alice	1101098	65185607	50	MEM!
BigBadWolf	1663499	92440037	30	MEM!
Cinderella	2140342	3035483	20	MEM!

Ad-hoc LP Algorithms

- *Critical Multiplier Algorithm*

$$\begin{aligned} &\max cx \\ &\text{st} \\ &x_i \leq x_j \quad \forall (i,j) \in A \\ &\sum a_i x_i \leq b \\ &x_i \in [0,1] \end{aligned}$$

$$\begin{aligned} &\max (c - \lambda a)x \\ &\text{st} \\ &x_i \leq x_j \quad \forall (i,j) \in A \\ &x_i \in [0,1] \end{aligned}$$

Explicit construction of optimal multipliers

- *BZ Algorithm*

The BZ algorithm

1. Start with $\mu^{-1} = 0$, $C_1^0 = \mathcal{B}$, $C^0 = \{C_1^0\}$ and $k = 0$.

2. Let y^k be the solution of $L(P, \mu^{k-1})$, problem where the side-constraints are penalized with multipliers μ^{k-1}.

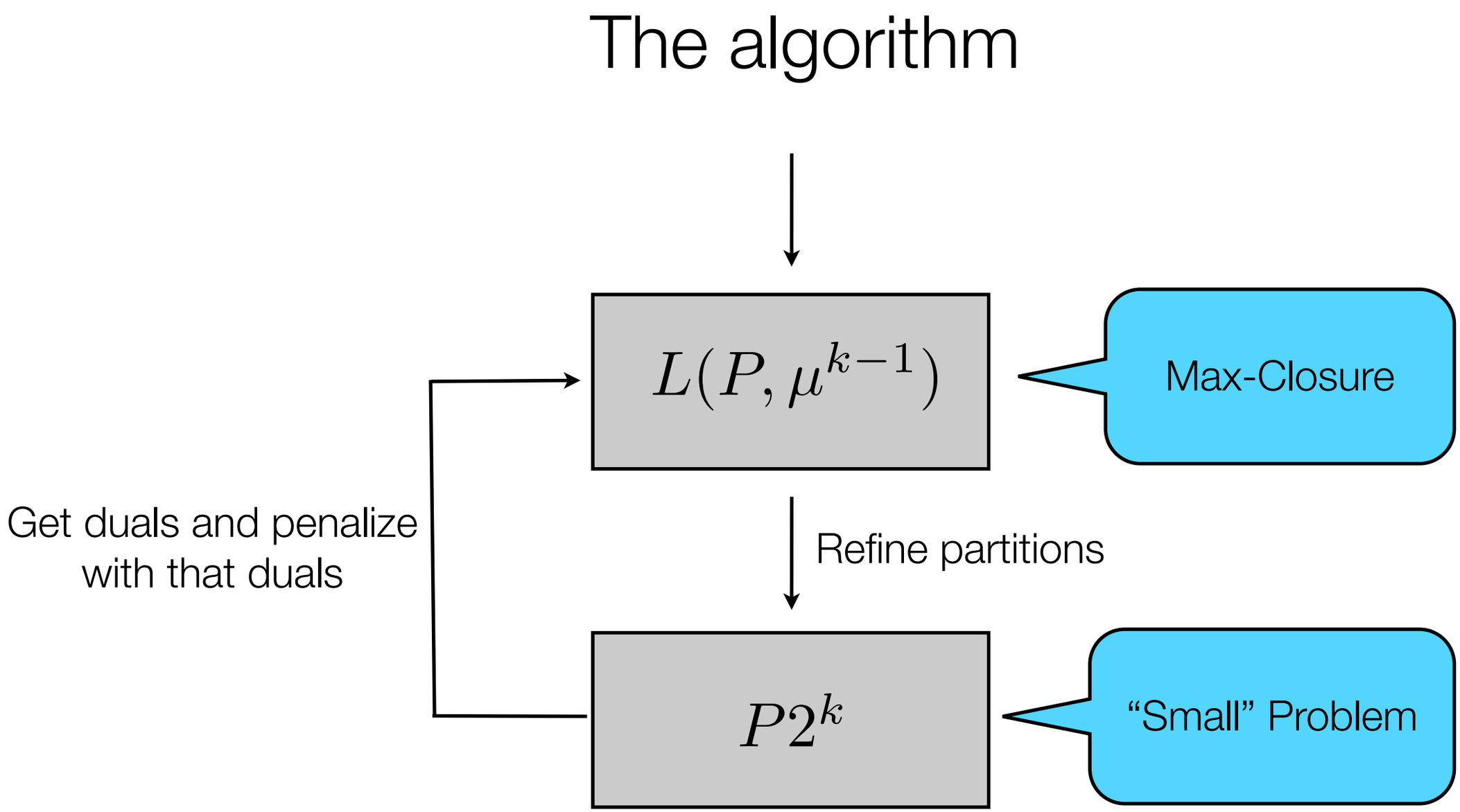
   If $k > 1$ and y^k satisfies $H^{k-1}x = 0$ stop.

3. Define C^k partition of $\mathcal{B}$ and impose the constraints $H^k x = 0$ consistent of $x_i = x_j$, $\forall i, j \in C_h^k$. Also the solution y^k must satisfy the constraints $H^k x = 0$.

4. Solve $P2^k$, obtained by adding the constraints $H^k x = 0$ to the problem P, and call the solution x^k with dual variables μ^k associated to the side-constraints. If $\mu^k = \mu^{k-1}$ stop.

5. Set $k \leftarrow k + 1$ and go to step 2.

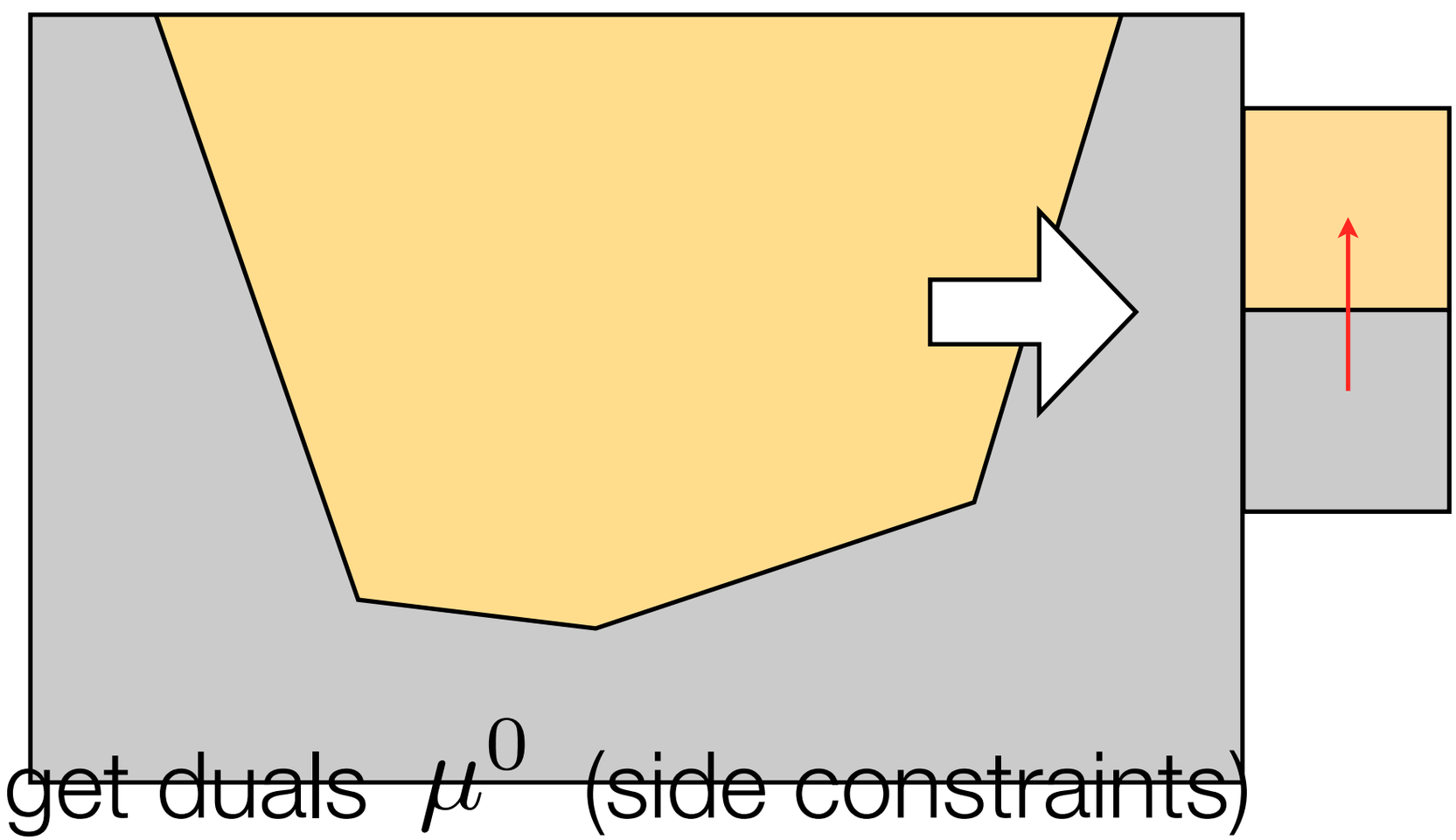
BZ algorithm



BZ algorithm

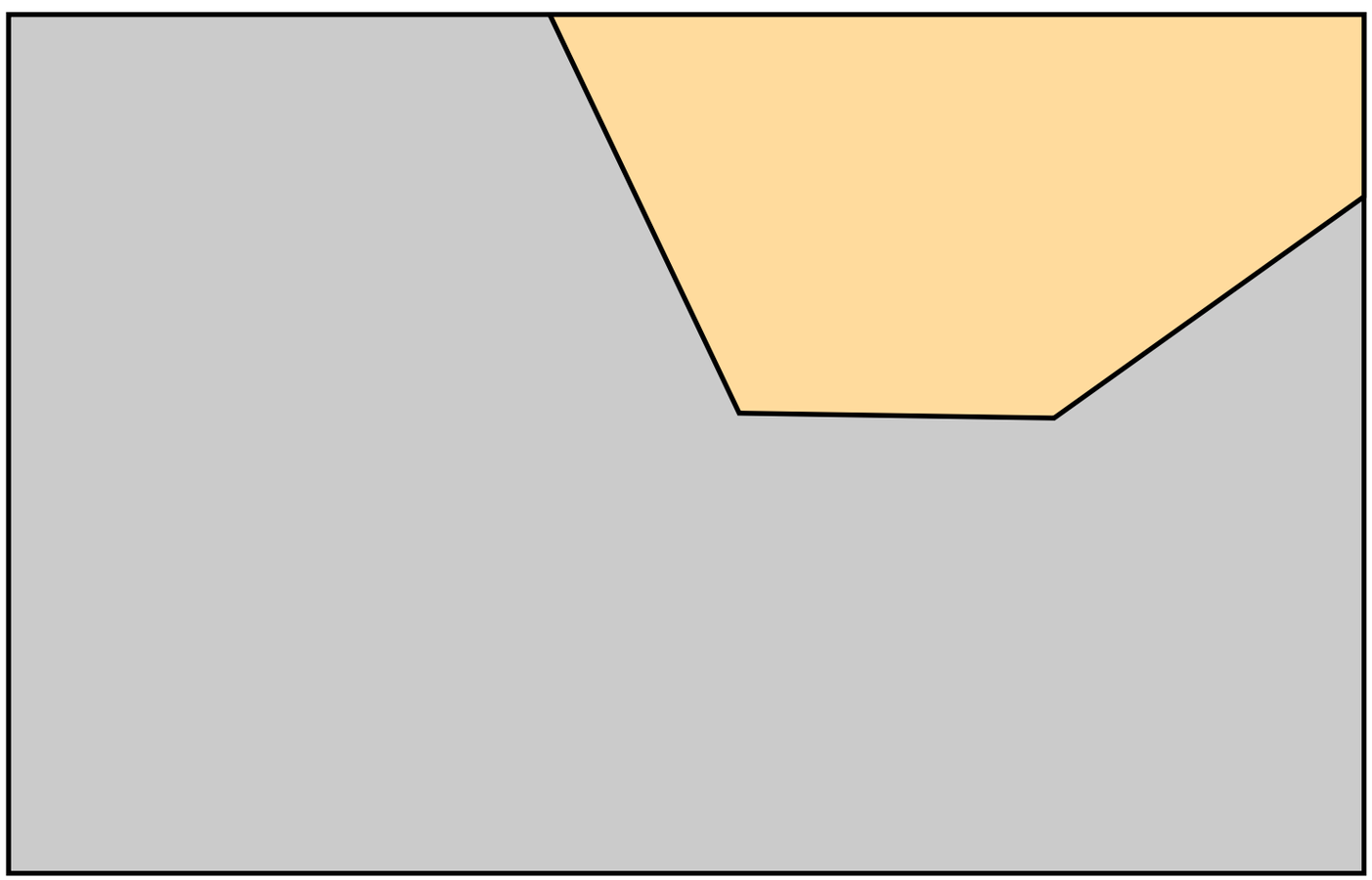
Graphically

- Define $LP(\mathcal{P}^0, \vec{a})$ and solve it



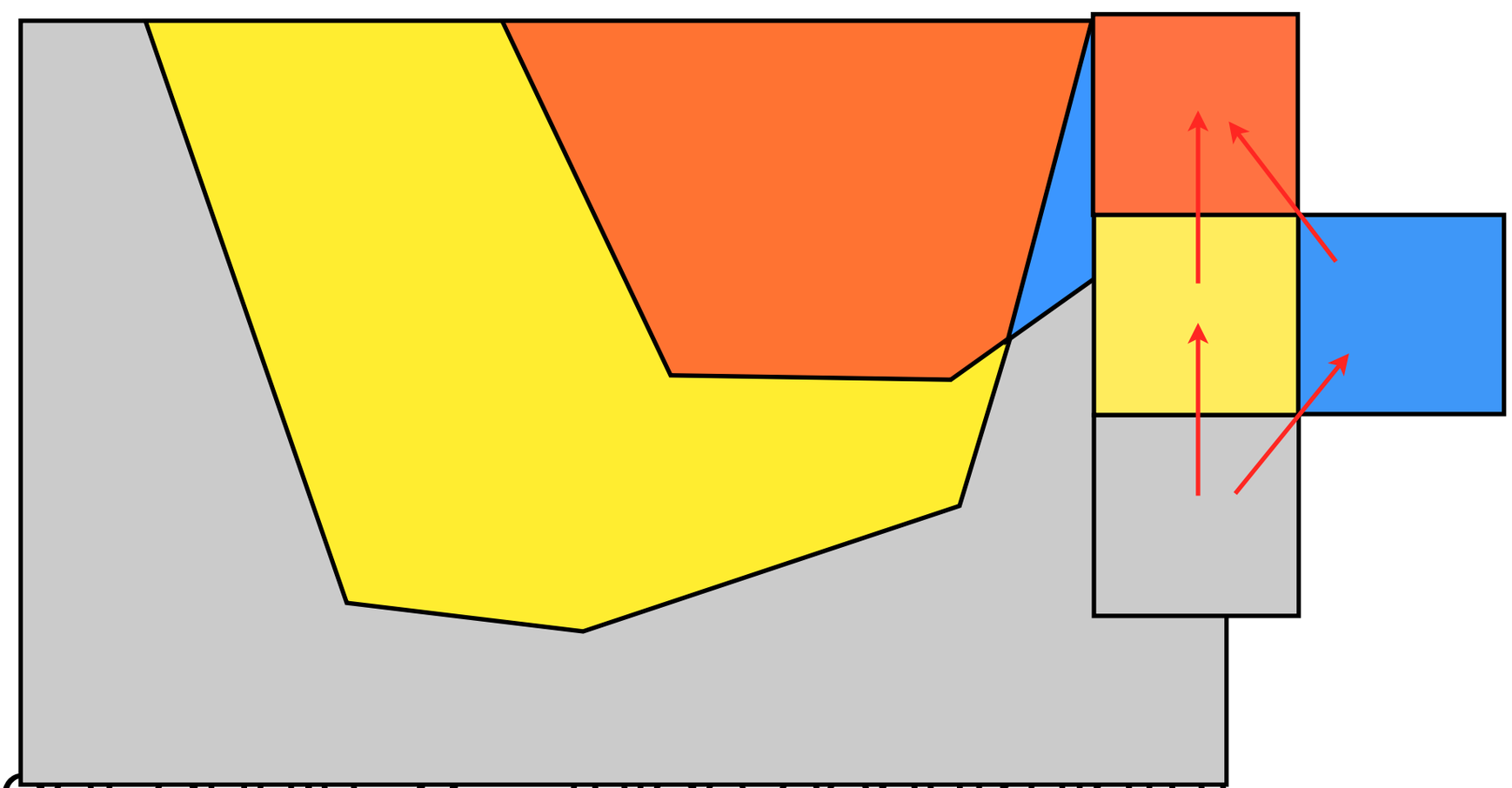
BZ algorithm

- Define $L(P, p^0)$
- Define the partition



BZ algorithm

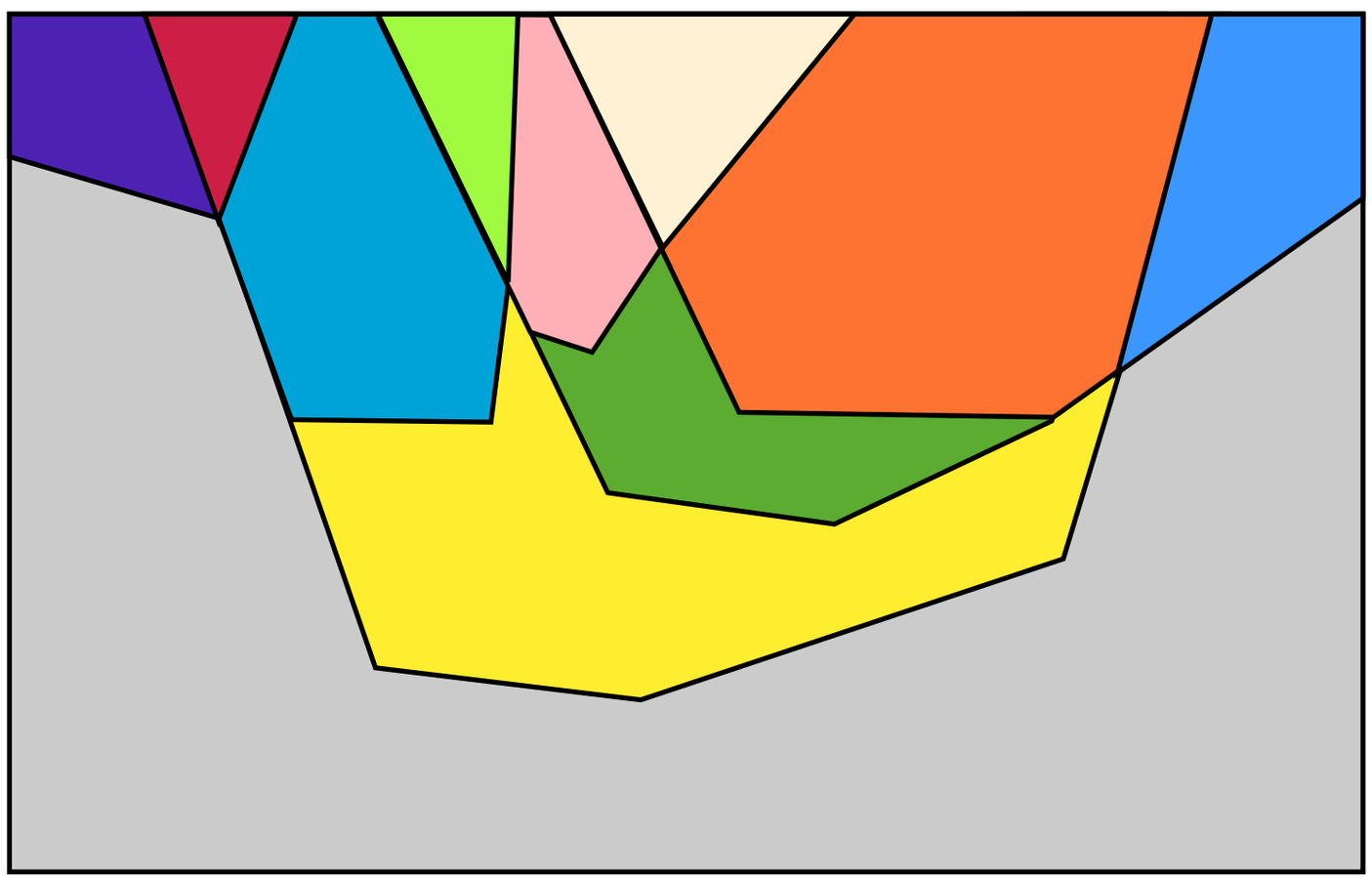
- Define the partition P_2^1 and solve it



get duals μ (side constraints)

BZ algorithm

And on...



BZ algorithm

BZ algorithm (+ a LOT of tricks)

Mine	Blocks	Prec.	Periods	CPX Time	BZ Time
Tinkerbell	1060	3922	6	8.44	0.09
PeterPan	9400	145640	20	327359.74	11.51
Pinochio	14153	219778	12	95631.97	8.19
Mowgli	16292	74260	20	29785.6	2.17
Hansel	29277	1271207	15	MEM!	67.53
Ariel	53271	656411	20	MEM!	7.89
SnowWhite	57958	1186665	21	MEM!	221.36
Grettel	96821	1053105	30	MEM!	497.34
Simba	99014	96642	30	5700.51	58.18
Cinderella-Limit	112687	3035483	15	MEM!	761.52
Dorothy	167937	5956121	51	MEM!	1369.07
Rapunzel	304977	15665274	81	MEM!	121675.2
Beauty	326428	1518326	80	MEM!	7500.09
Aladdin	329859	7066518	60	MEM!	16621.1
Alice	1101098	65185607	50	MEM!	7173.48
BigBadWolf	1663499	92440037	30	MEM!	MEM!
Cinderella	2140342	3035483	20	MEM!	278.73

General algorithm template

We consider the following problem

$$
\begin{array}{lll}
(P) & \max & cx \\
 & \text{s.t} & Ax \leq b \\
 & & Dx \leq d
\end{array}
$$

"Easy"

"Hard"

General algorithm template

Feasible solution z^0

$k = 1 \quad \mu^0 = \vec{0}$

$$\boxed{L(P, \mu^{k-1})}$$

General algorithm template

Feasible solution z^0

$k = 1 \quad \mu^0 = \vec{0}$

$$\boxed{L(P, \mu^{k-1})}$$

General algorithm template

Feasible solution z^0

$$k = 1 \quad \mu^0 = \vec{0}$$

$$\downarrow$$

$$\boxed{L(P, \mu^{k-1})} \longrightarrow w^k$$

General algorithm template

Feasible solution z^0

$k = 1 \quad \mu^0 = \vec{0}$

$\downarrow$

$L(P, \mu^{k-1})$

$\downarrow$

$P2^k$

General algorithm template

Feasible solution z^0

$k = 1 \quad \mu^0 = \vec{0}$

$L(P, \mu^{k-1})$

Get duals and penalize with that duals

$P2^k$

General algorithm template

Clearly $H^k x = h^k$ defines an affine subspace, but for simplicity let's assume $h^k = 0$ so the subspace is linear

General algorithm template

This way we can rewrite the $P2^k$ problem as

$$
\begin{aligned}
(P2^k) \quad \max \quad & c\left(\sum_i \lambda_i x^i\right) \\
\text{s.t} \quad & A\left(\sum_i \lambda_i x^i\right) \leq b \\
& D\left(\sum_i \lambda_i x^i\right) \leq d
\end{aligned}
$$

where $\{x^i\}_i$ is a generator of $\{x \mid H^k x = 0\}$

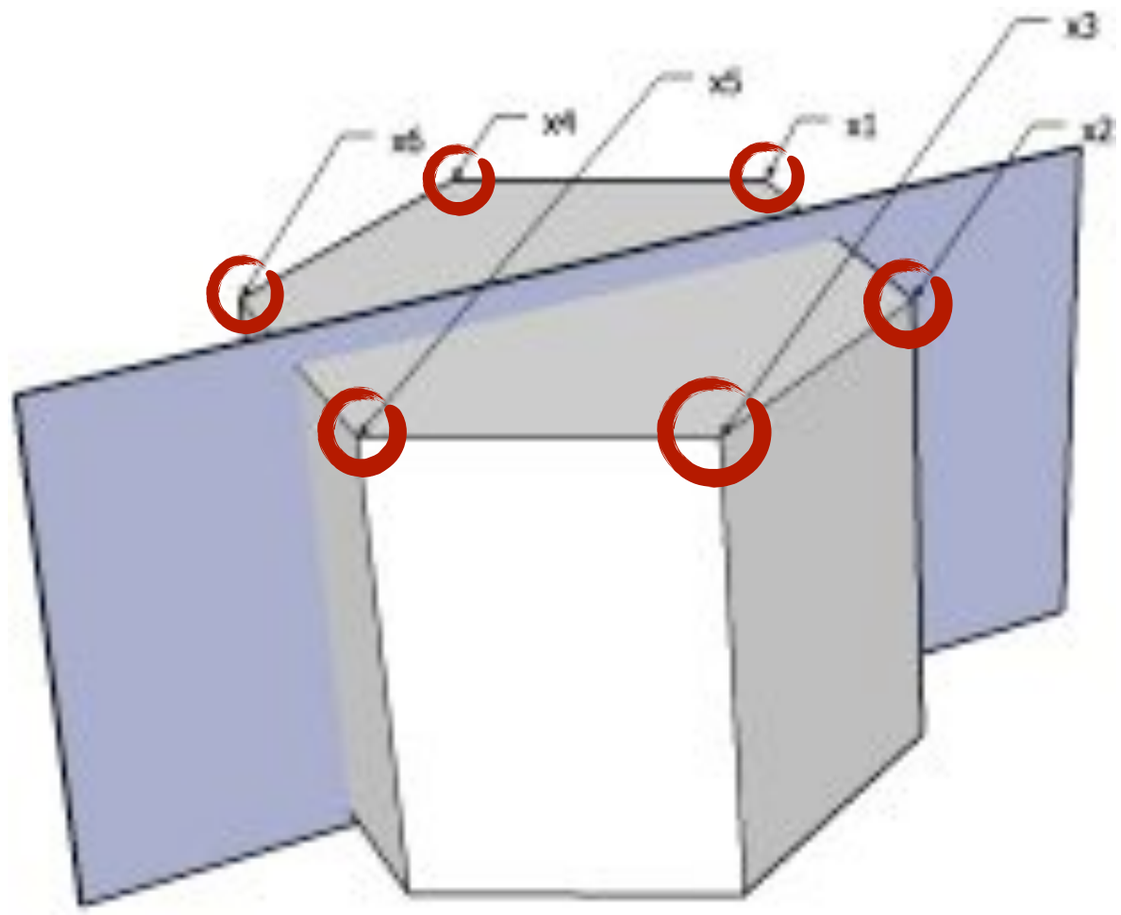
Differences?

BZ	Dantzig-Wolfe
In the master problem we move in an affine subspace that contains the generated point	In the master problem we move in the convex hull of the generated extreme points
We have the freedom of choosing the affine subspace	We add one column (extreme point) at a time

We solve the same “pricing” problem in both

Graphically

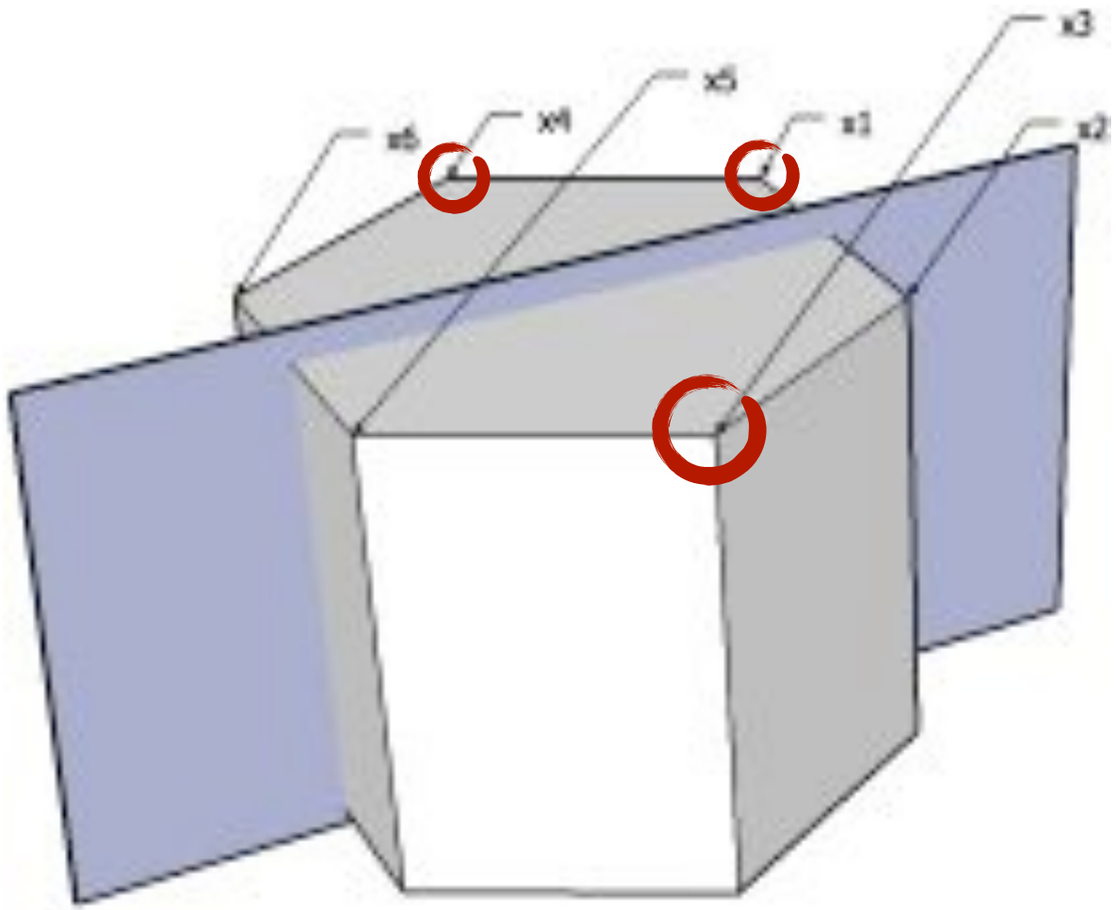
Maybe DW would need to generate...



before moving out of the upper lid

Graphically

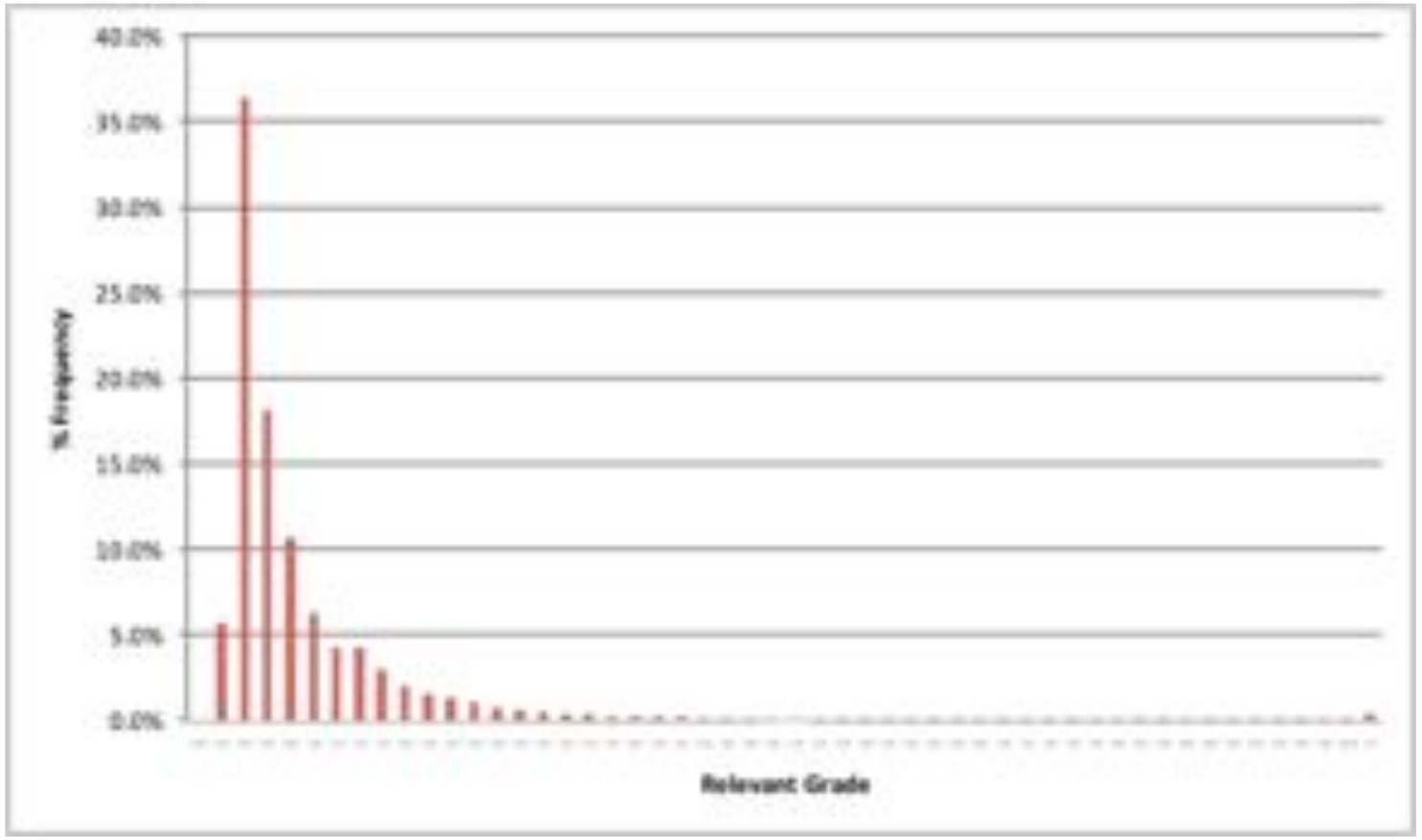
But BZ only needs...



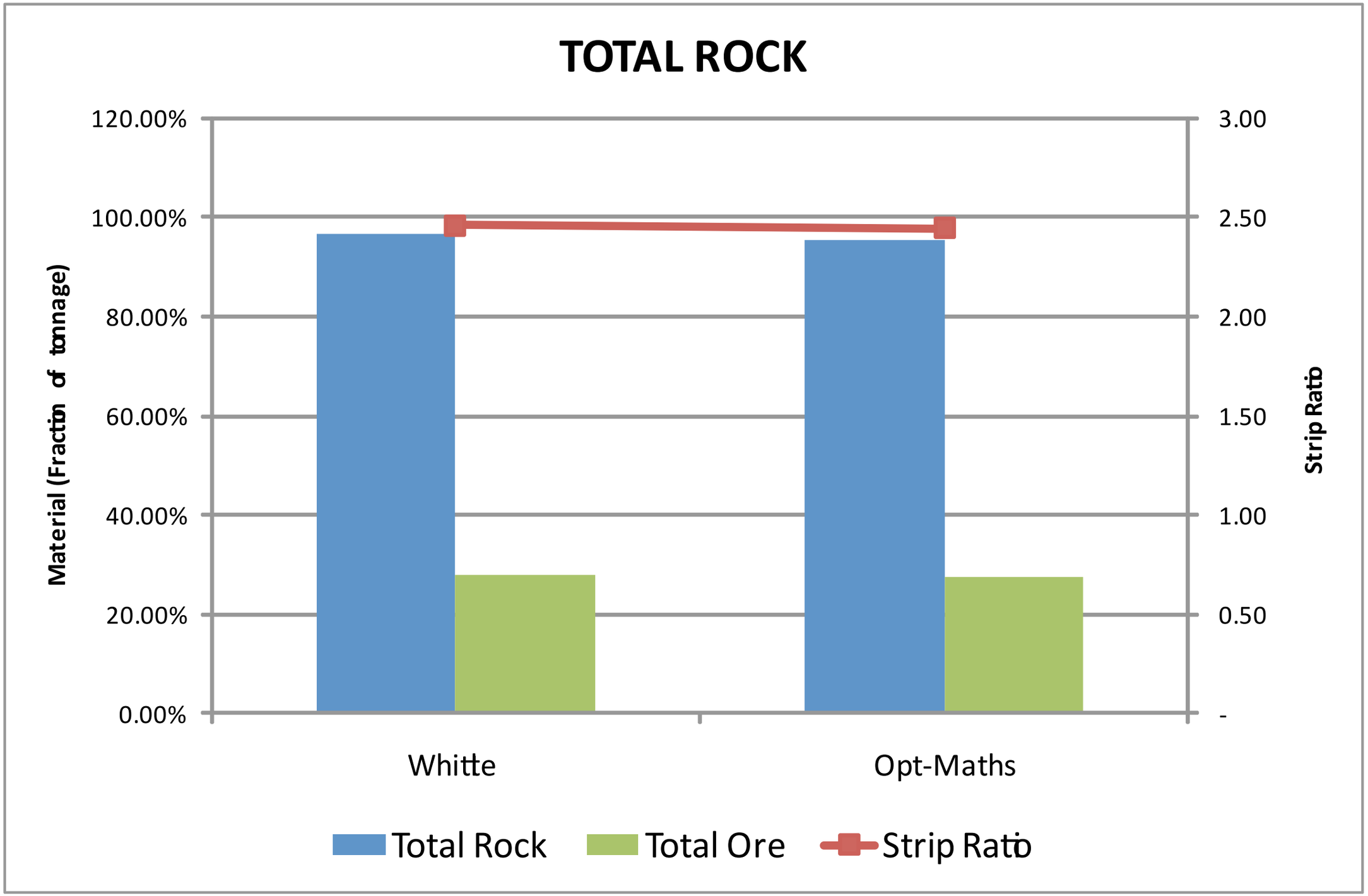
to describe the entire upper lid

Case Study

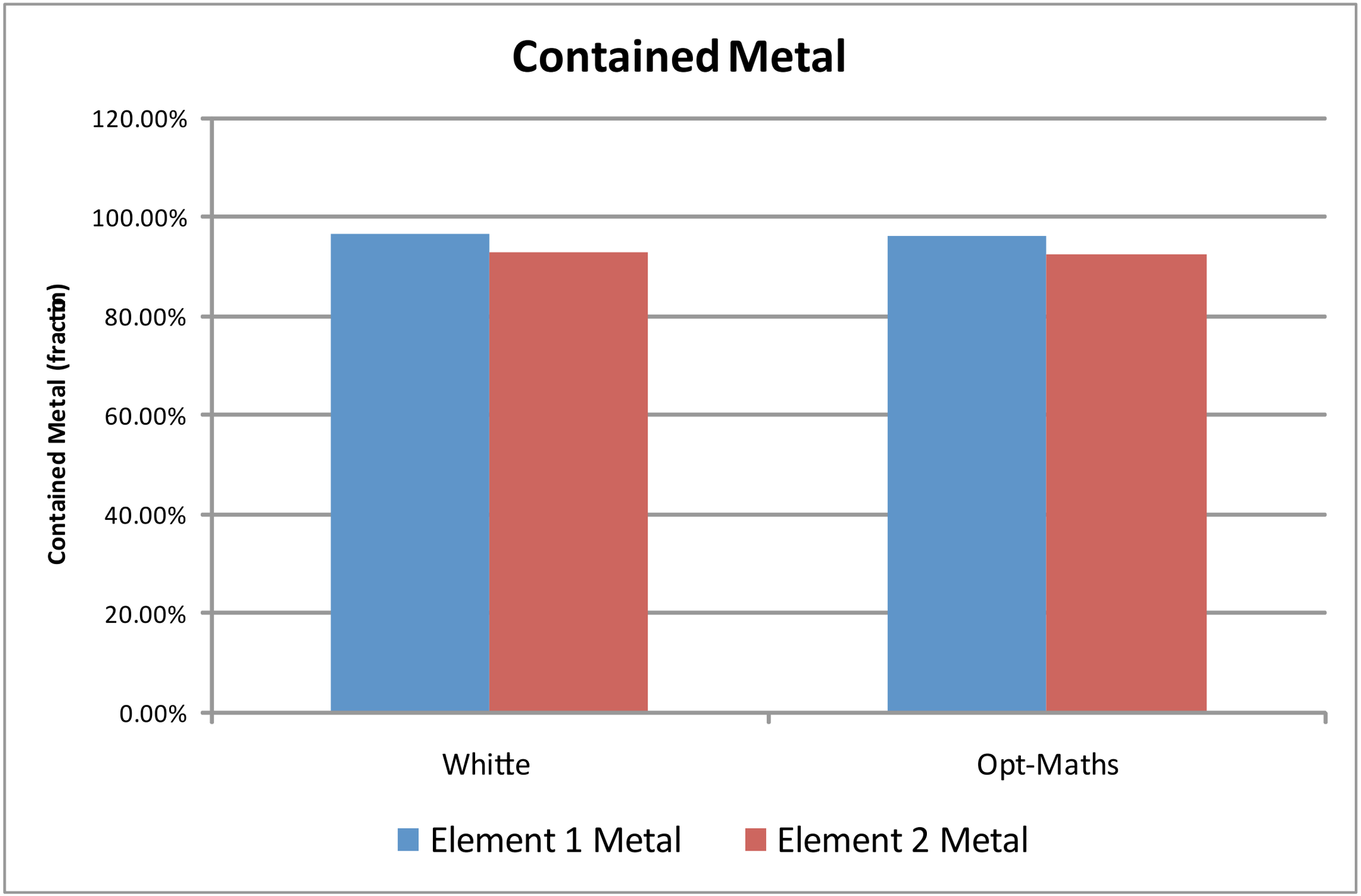
- Selective Mine, two metallurgical process
- Prices, Met. Recoveries, Mining, Process and Selling costs based on the usual numbers of the industry;
- Asymmetrical contribution of value by process: 10% of the reserves (Process A) give 45% of the cash flow of the project.
- Some comparison between both process (Process A / Process B):
- Process Cost 12:1



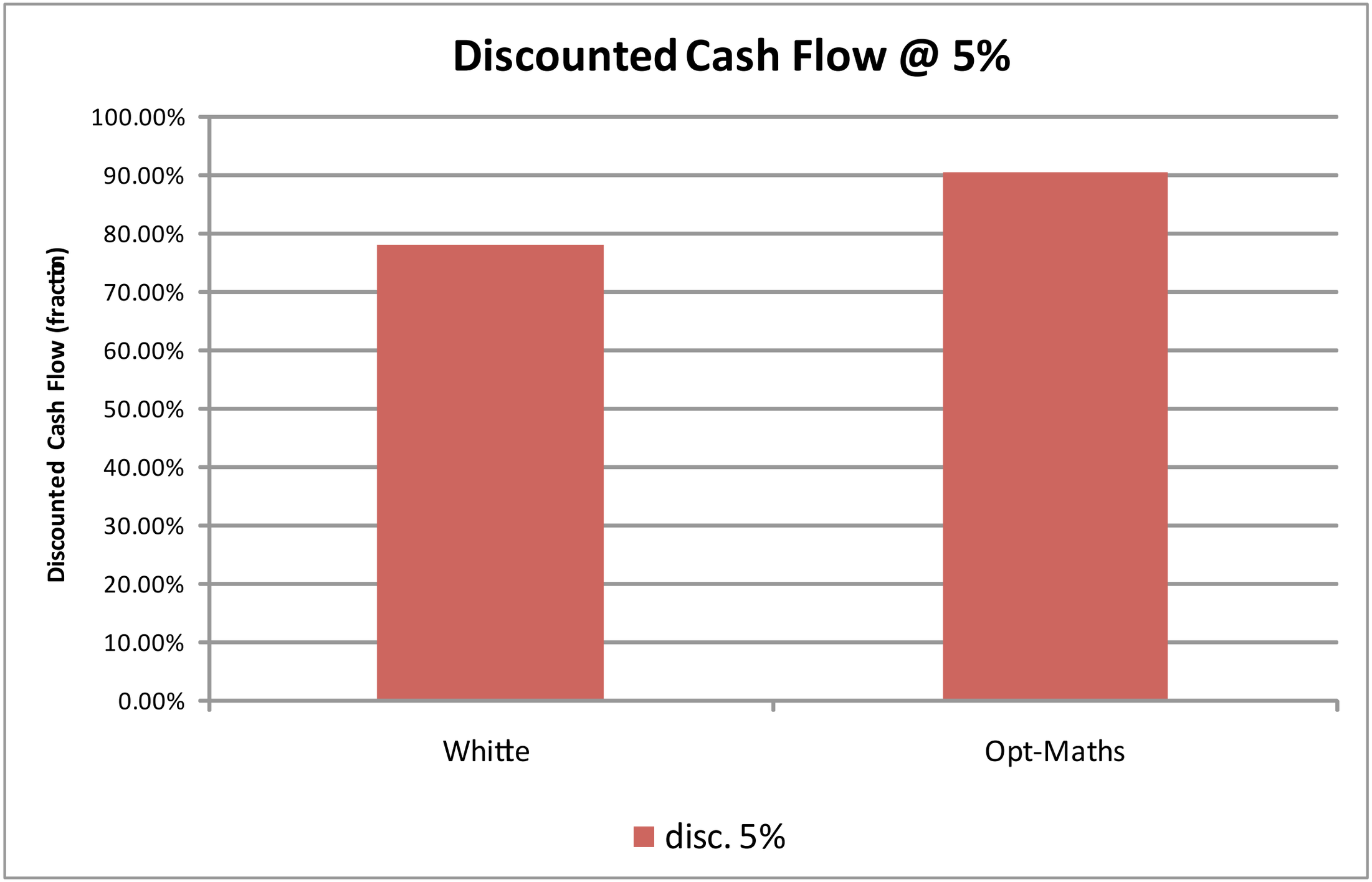
Inventory Reserves – Total Tonnes



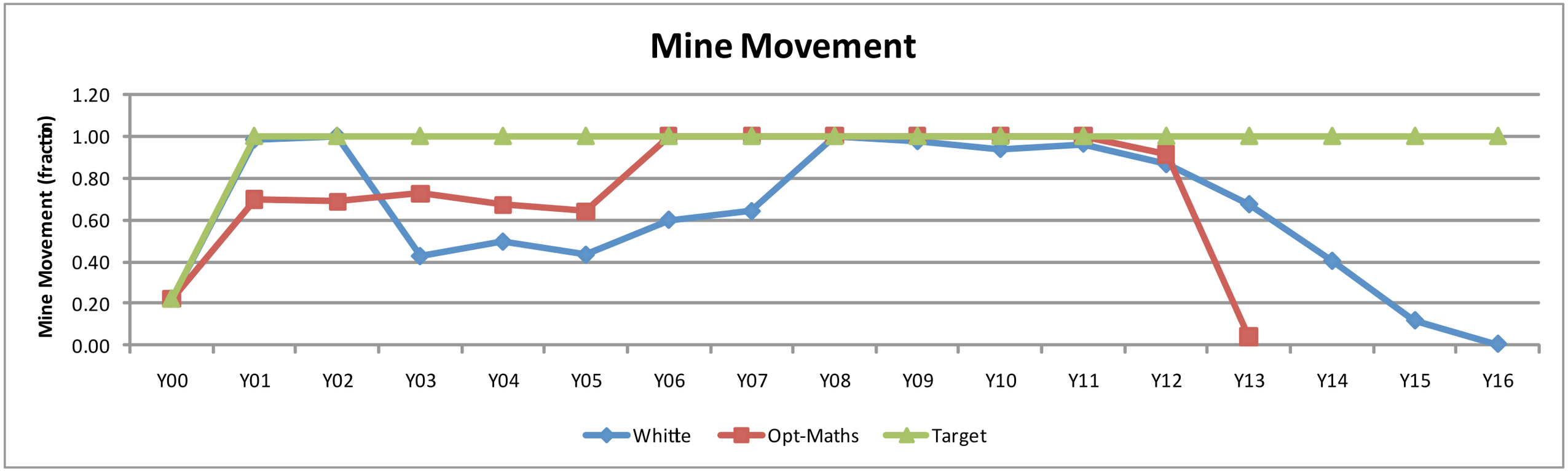
Inventory Reserves – Metal



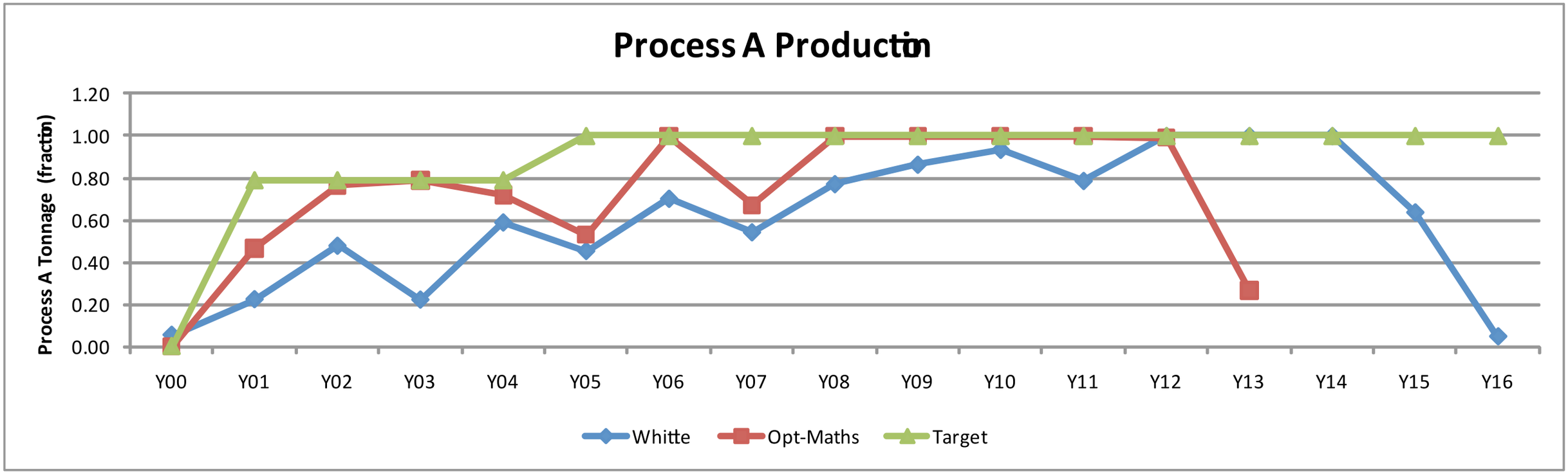
Inventory Reserves – Cash Flow

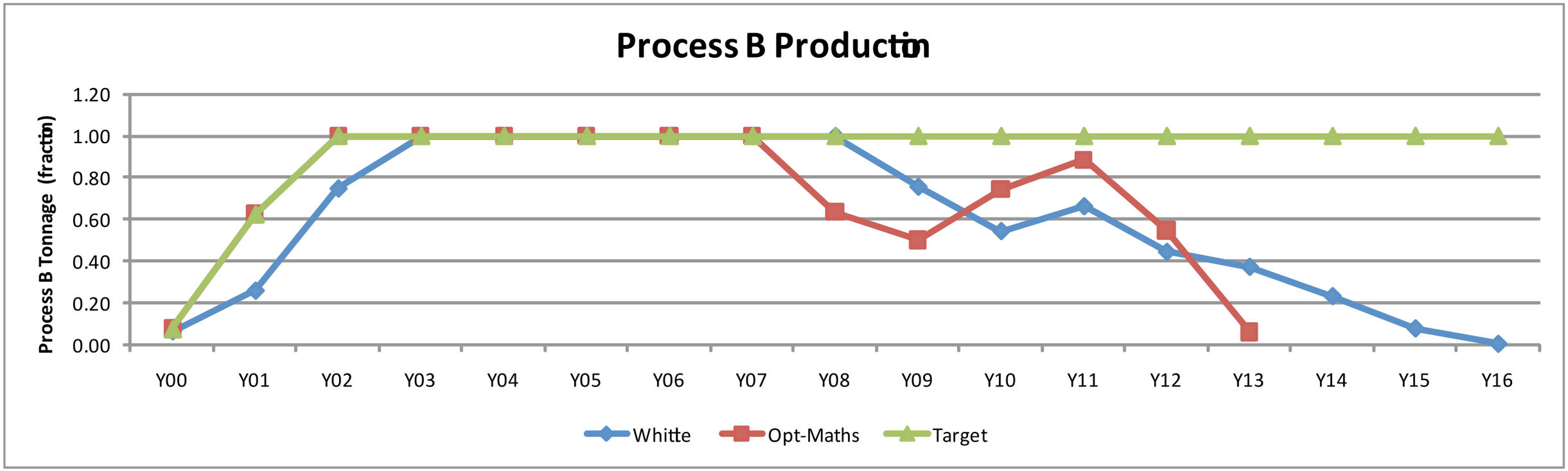


Mine Schedule - Tonnage

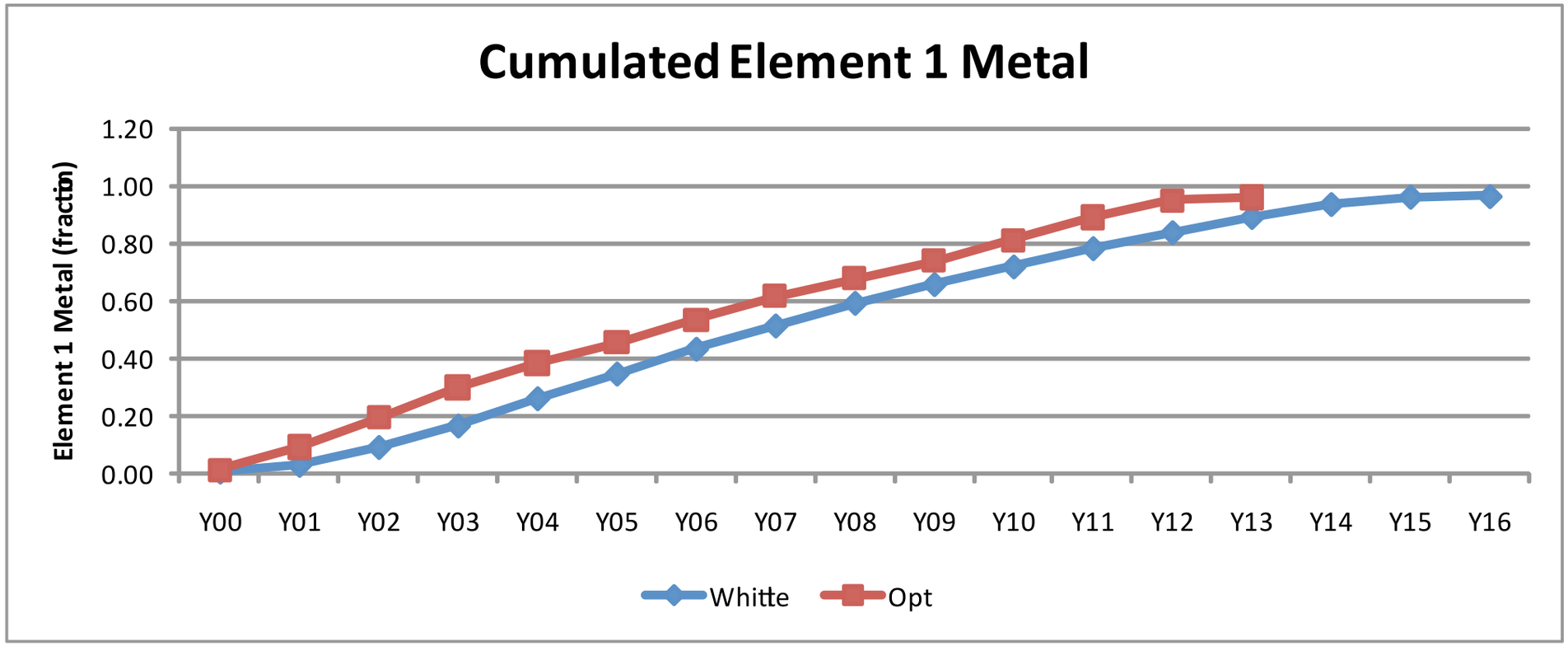


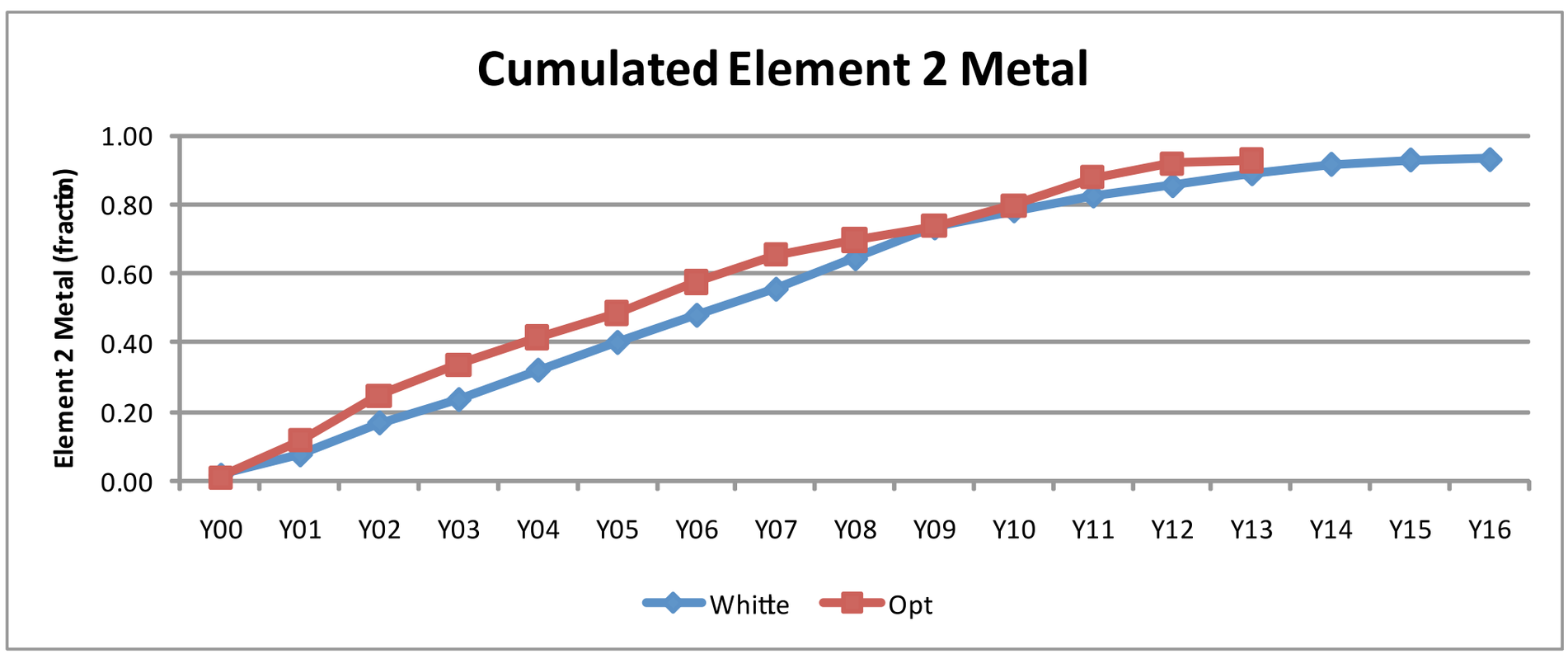
Mill Feed - Tonnage



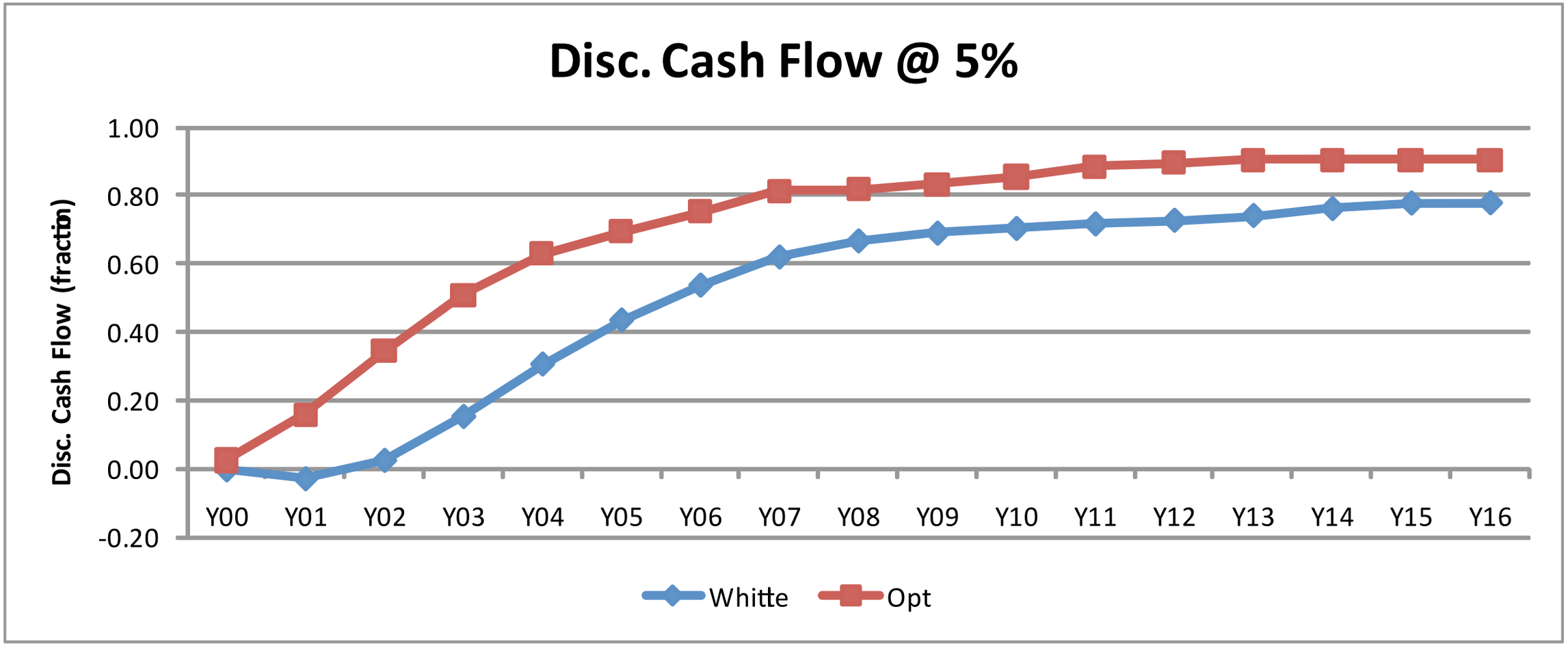


Mine Schedule – Cumulated Metal Production

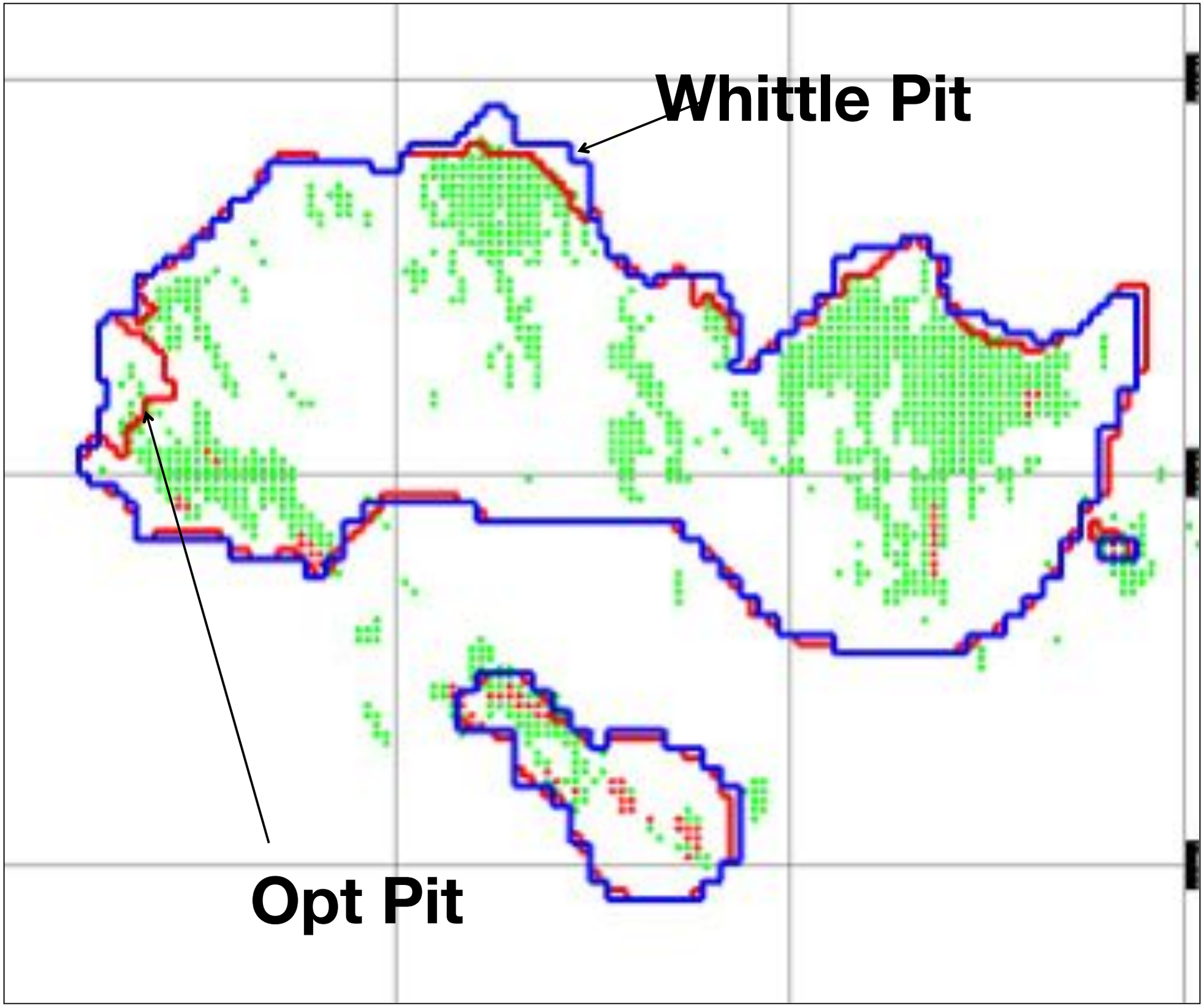




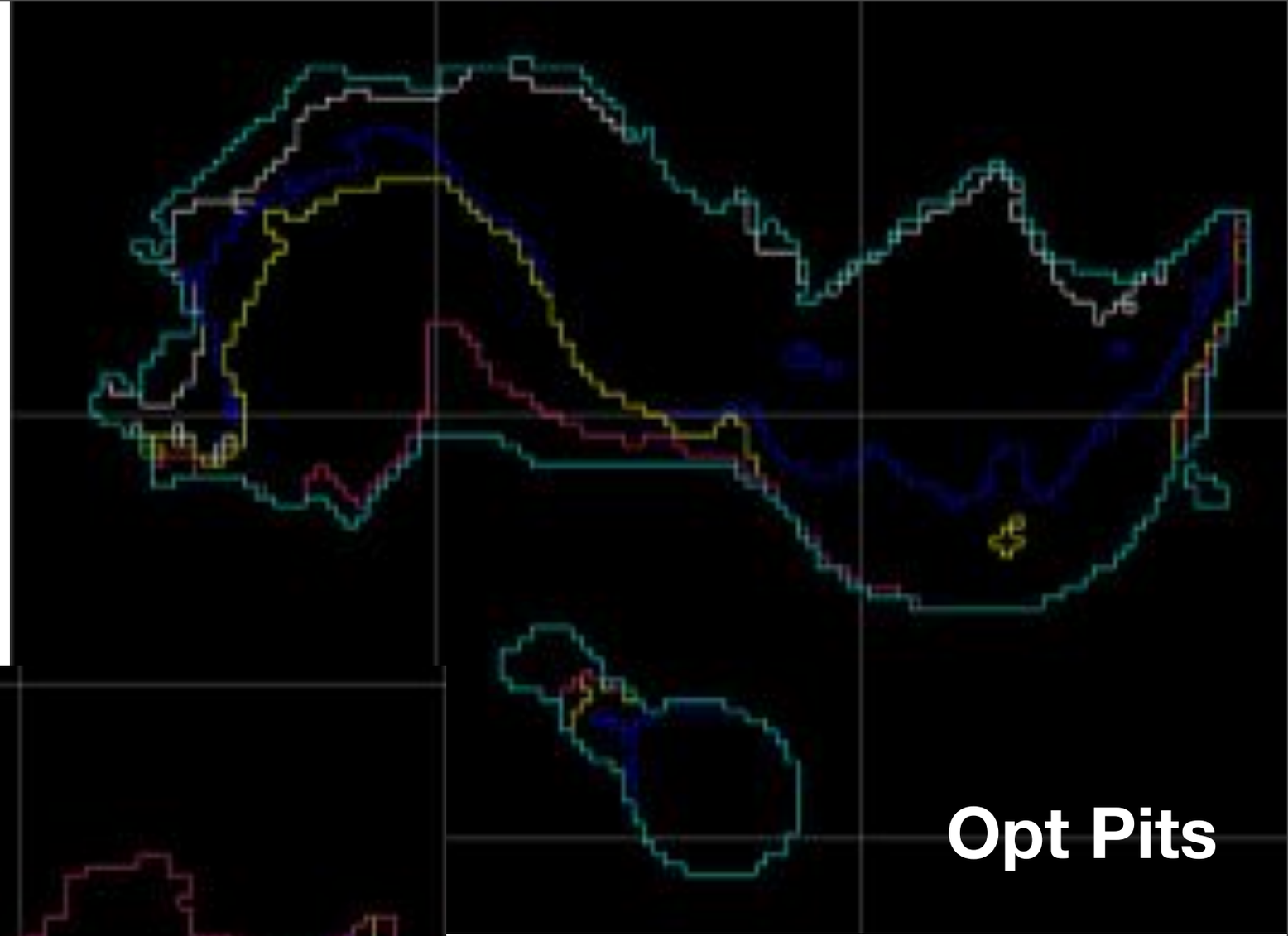
Mine Schedule – Cash Flow

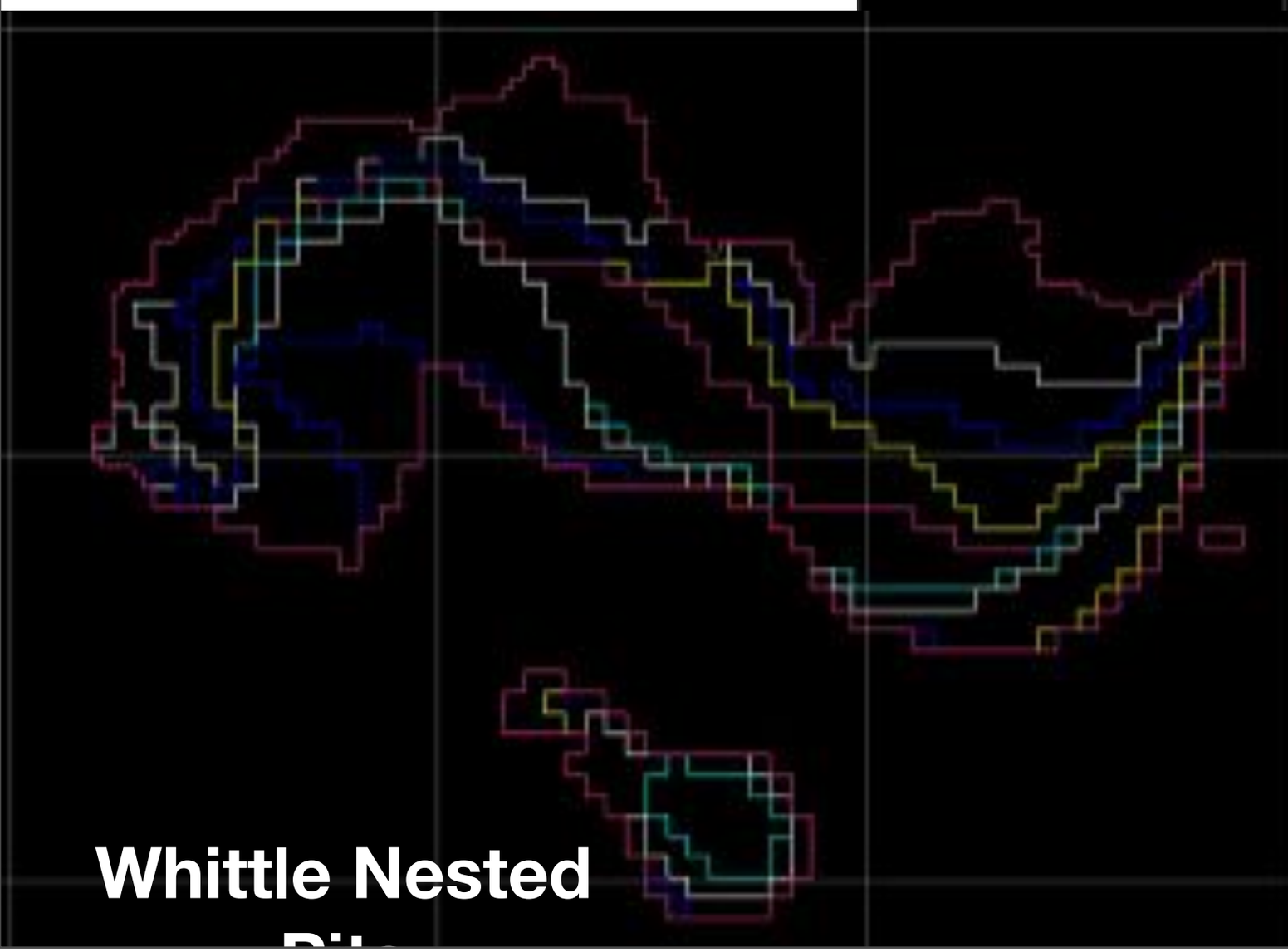


Ultimate Pit



Mining Sequence



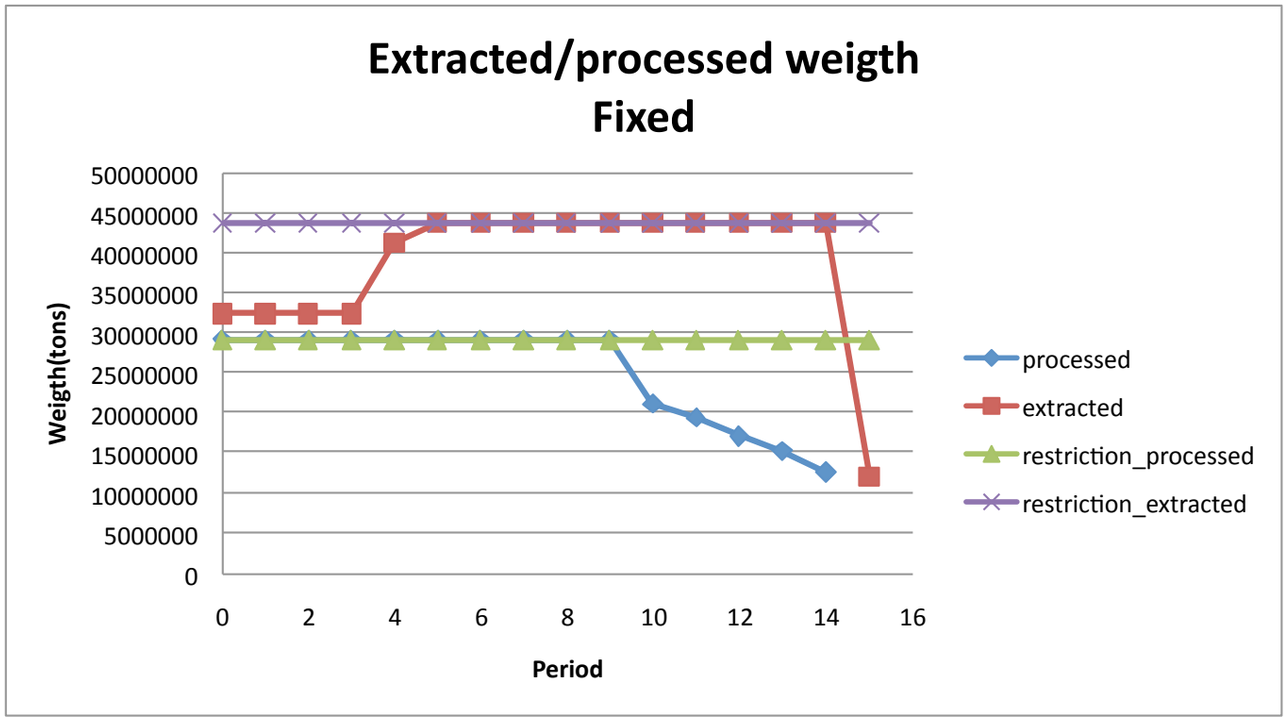


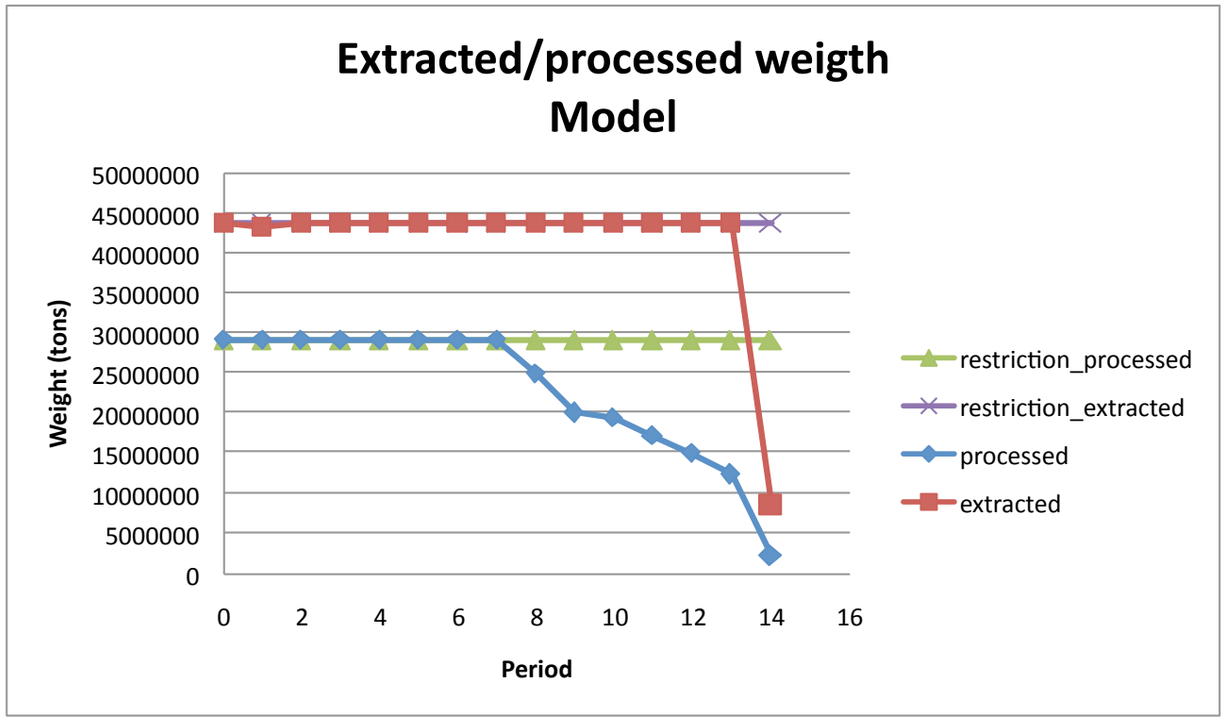
An important extension

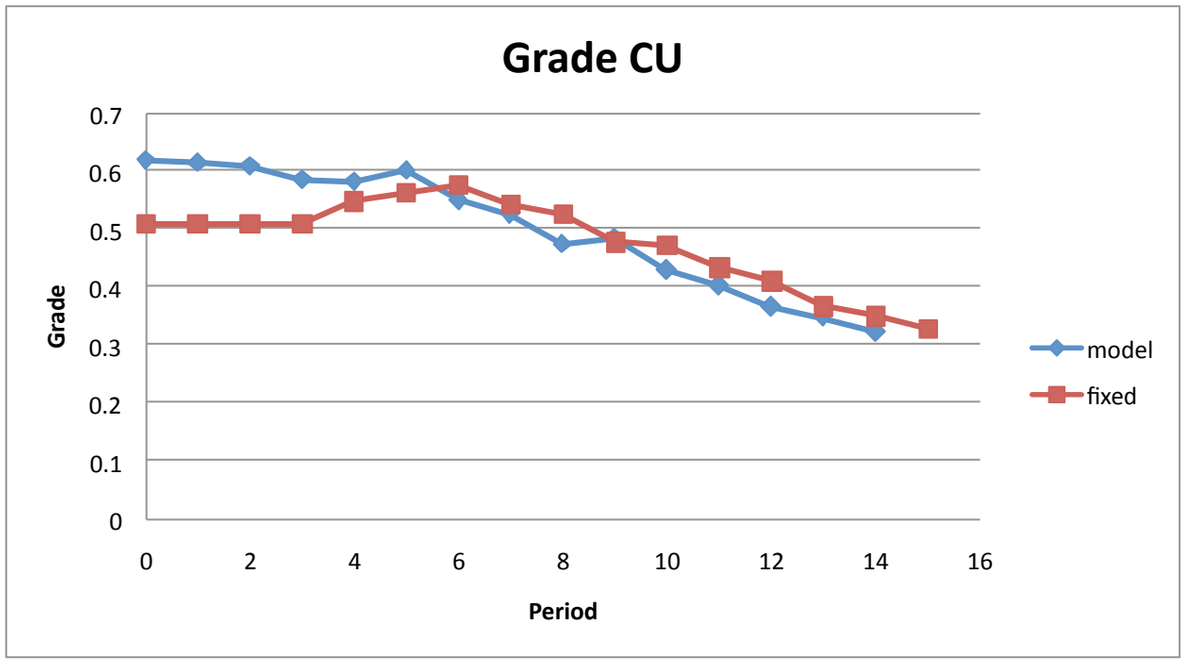
$$x_{b,d,t} = \begin{cases} 1 & \text{block } b \text{ is extracted in time } 1..t \text{ and destination } 1..d \\ 0 & \text{otherwise} \end{cases}$$

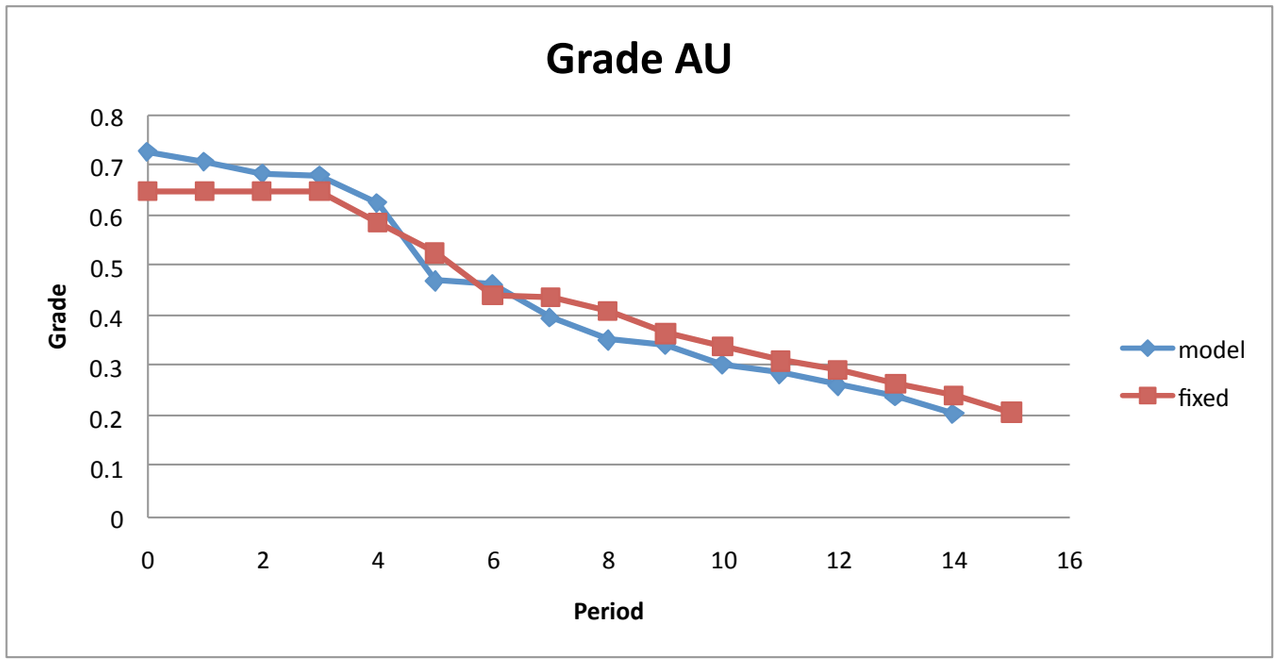
$$\begin{aligned} x_{b,0,0} &\geq 0 \\ x_{b,R-1,t} &\leq x_{b,0,t+1} \quad t < T-1 \\ x_{b,d,t} &\leq x_{b,d+1,t} \quad d < R-1 \\ x_{b,R-1,T-1} &\leq 1 \\ x_{a,R-1,t} &\leq x_{b,R-1,t} \quad \forall (a,b) \in \mathcal{A}, \forall t \end{aligned}$$

Marvin

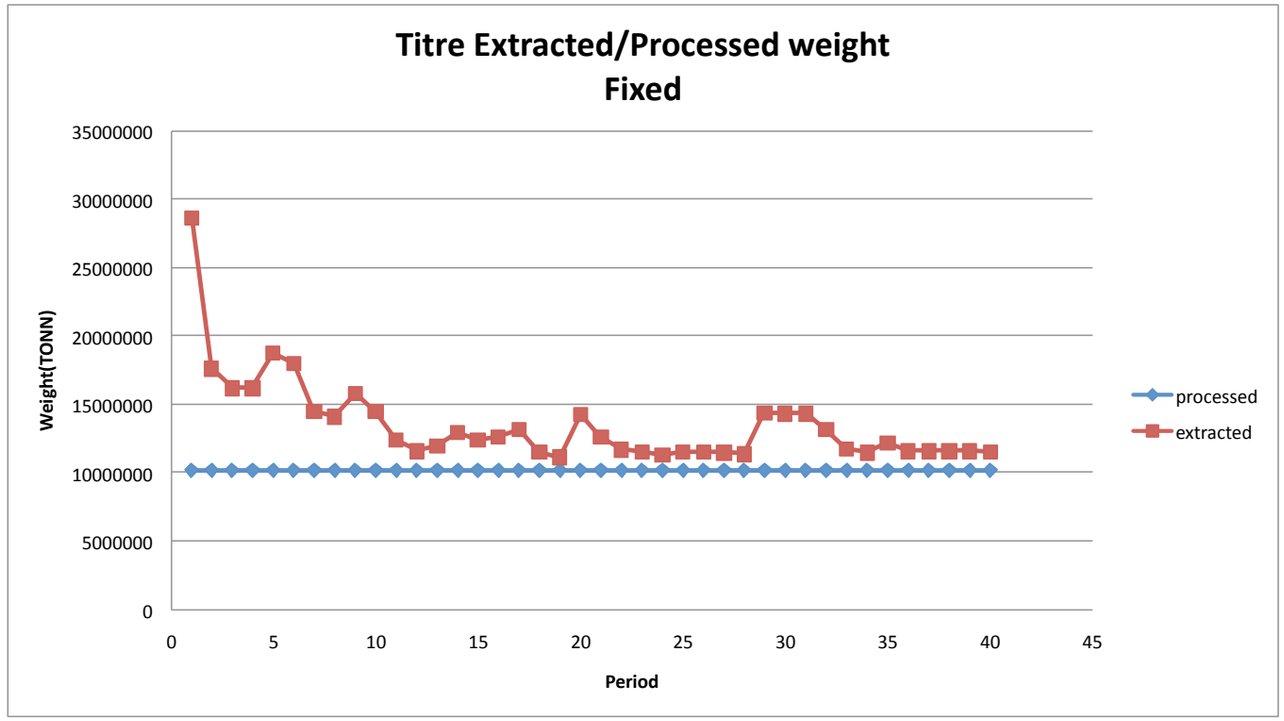

Extracted/processed weigth
Fixed
Weigth(tons)
50000000
45000000
40000000
35000000
30000000
25000000
20000000
15000000
10000000
5000000
0
0 2 4 6 8 10 12 14 16
Period
processed
extracted
restriction_processed
restriction_extracted

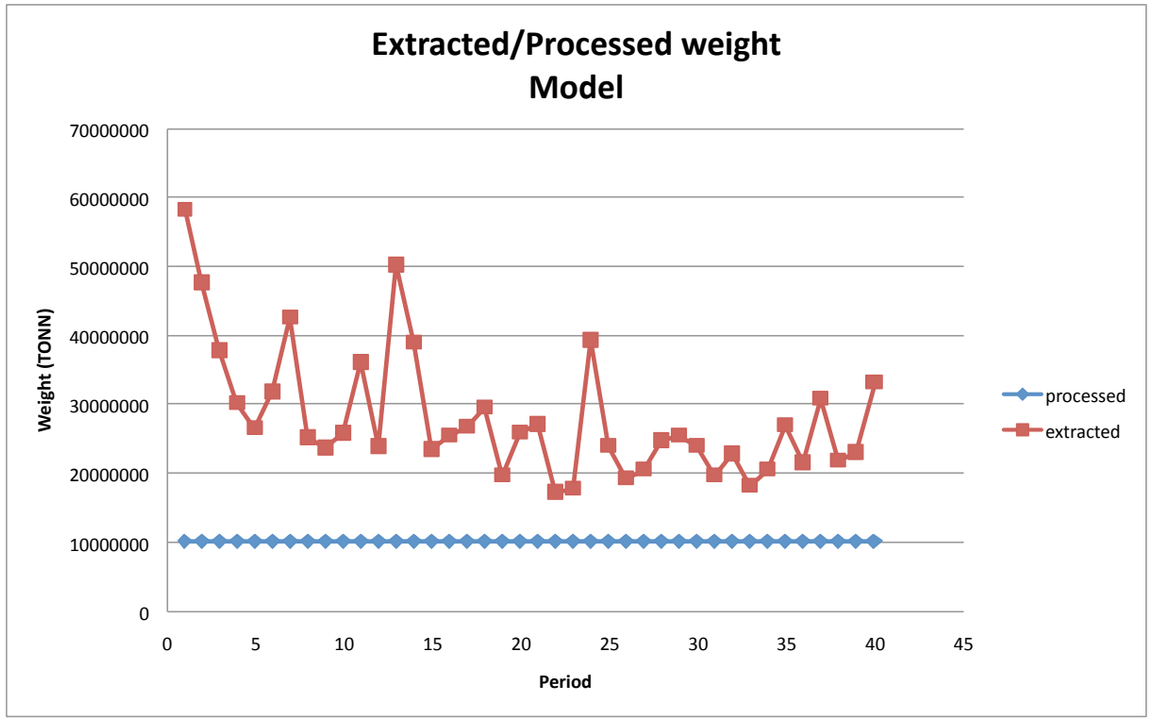

Extracted/processed weigth
Model
Weight (tons)
50000000
45000000
40000000
35000000
30000000
25000000
20000000
15000000
10000000
5000000
0
0 2 4 6 8 10 12 14 16
Period
restriction_processed
restriction_extracted
processed
extracted

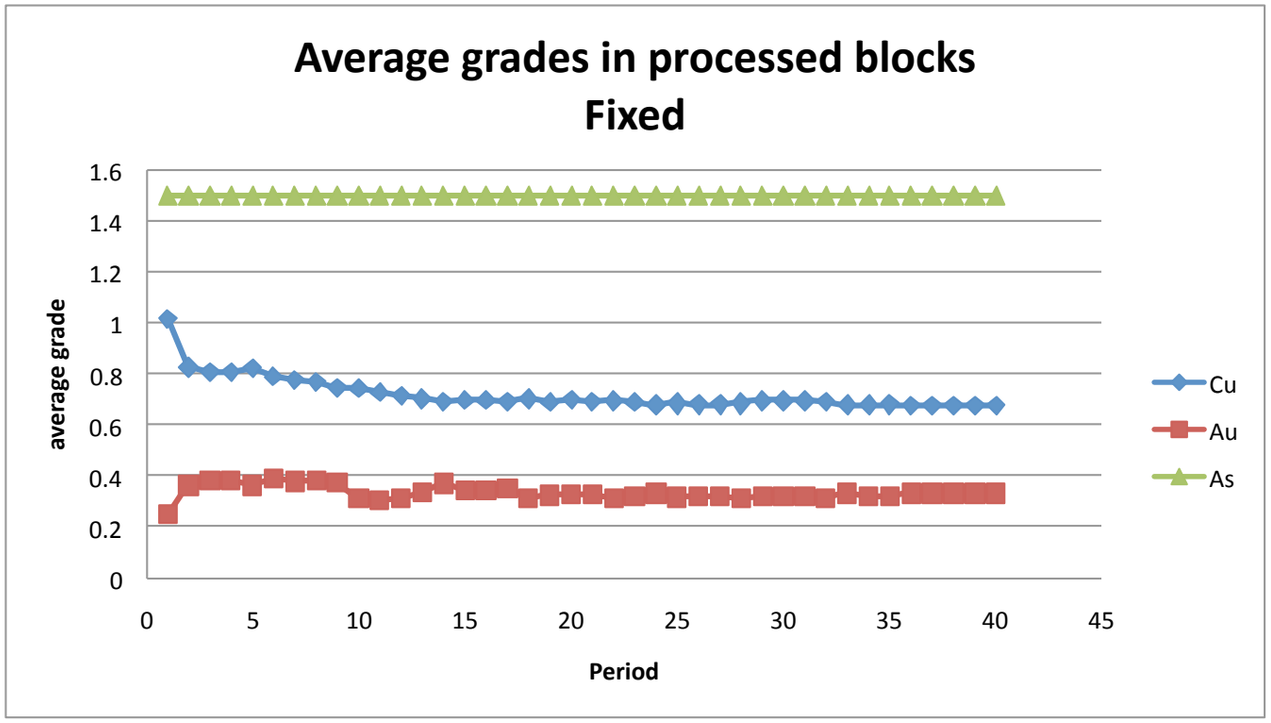

Grade CU
Grade
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 2 4 6 8 10 12 14 16
Period
model
fixed

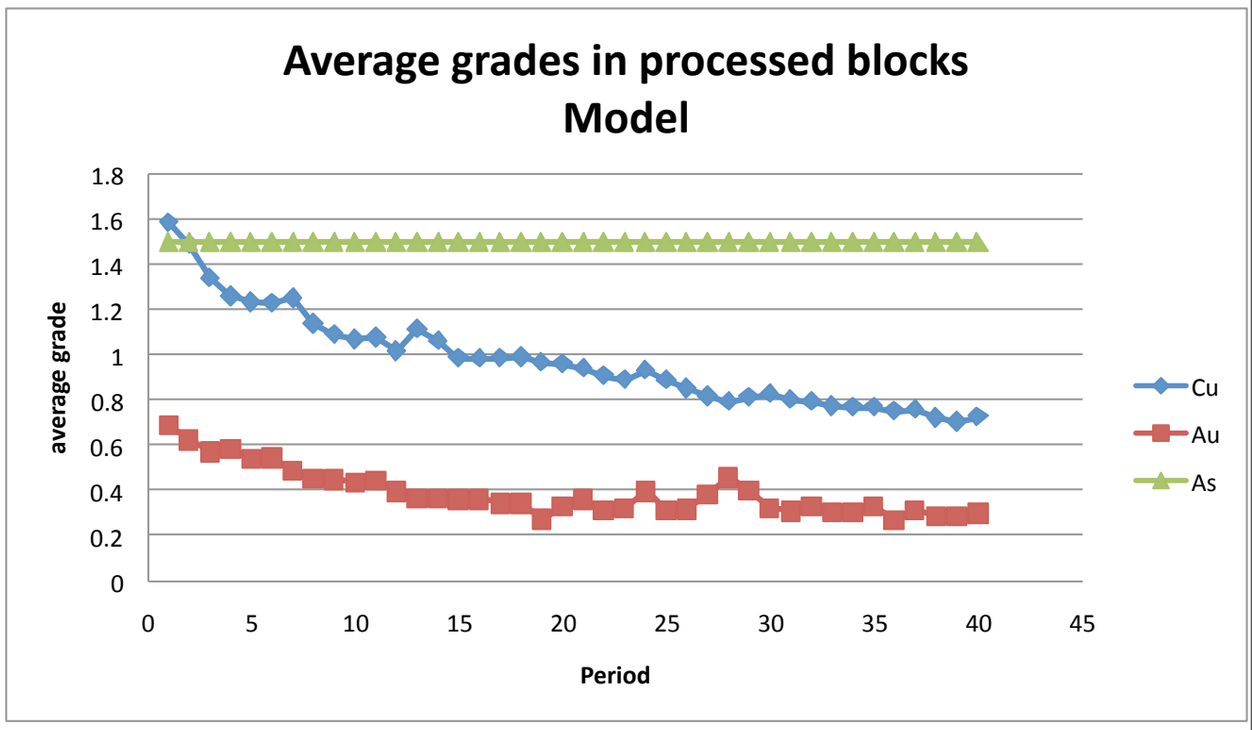

Grade AU
Grade
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 2 4 6 8 10 12 14 16
Period
model
fixed

Delphos

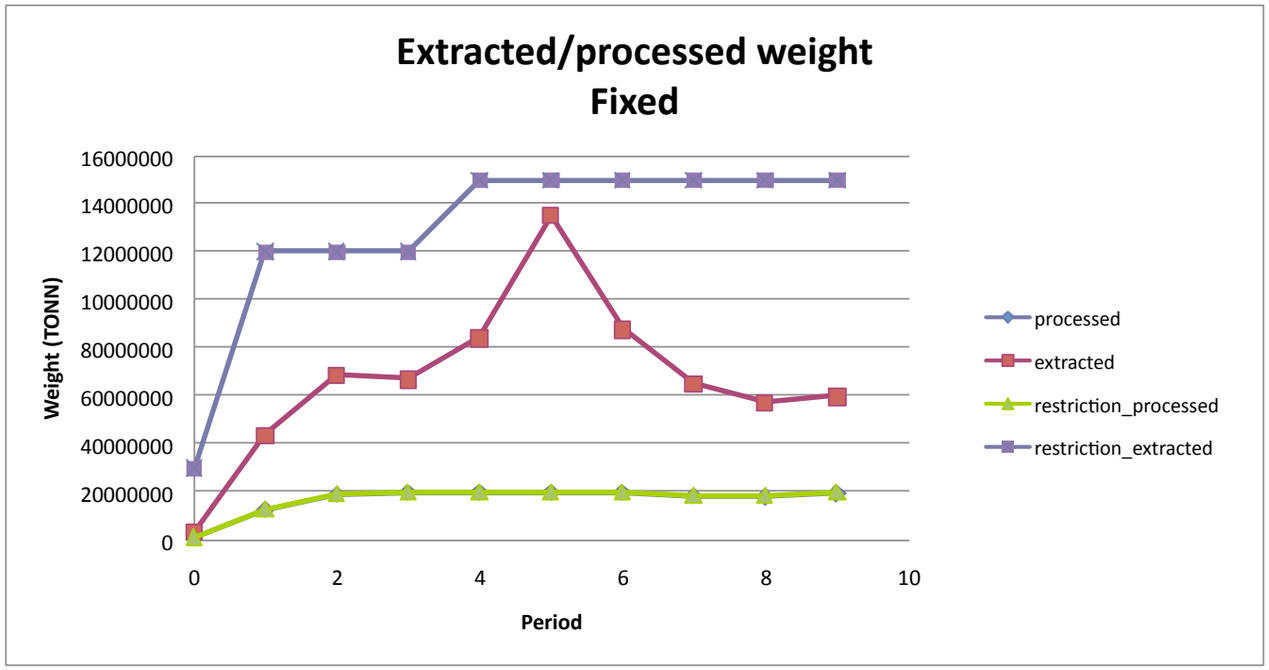

Titre Extracted/Processed weight
Fixed
Weight(TONN)
35000000
30000000
25000000
20000000
15000000
10000000
5000000
0
0 5 10 15 20 25 30 35 40 45
Period
processed
extracted

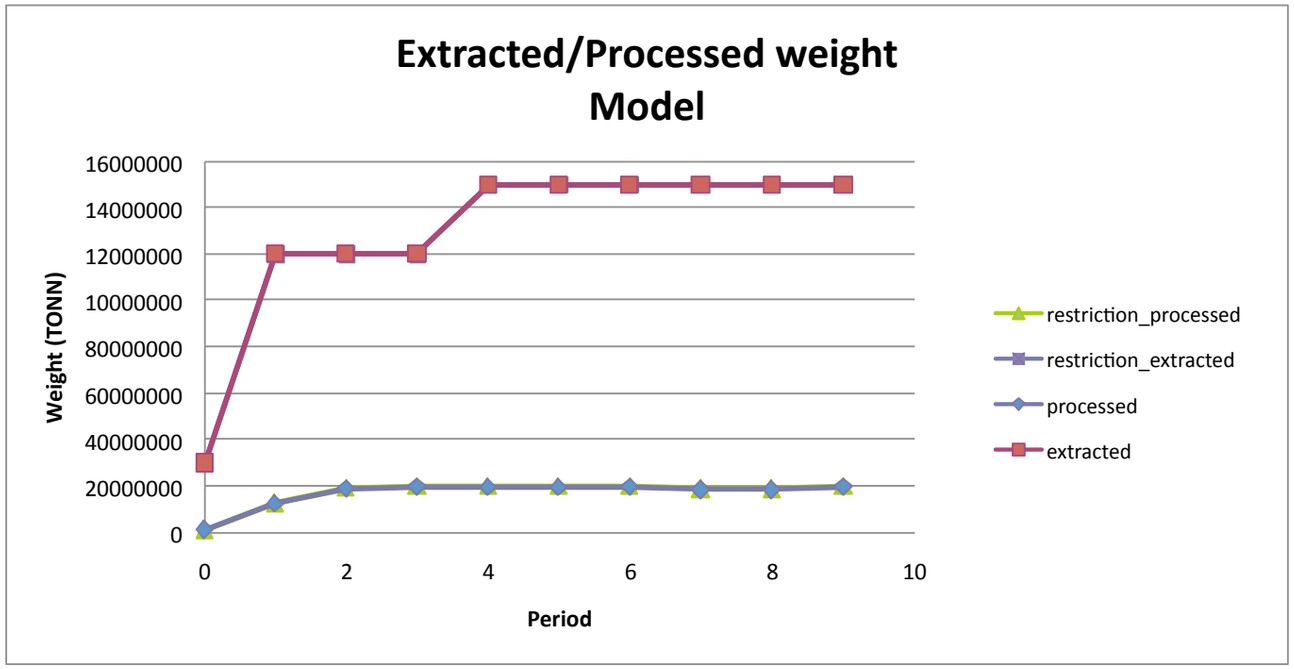

Extracted/Processed weight
Model
Weight (TONN)
70000000
60000000
50000000
40000000
30000000
20000000
10000000
0
0 5 10 15 20 25 30 35 40 45
Period
processed
extracted

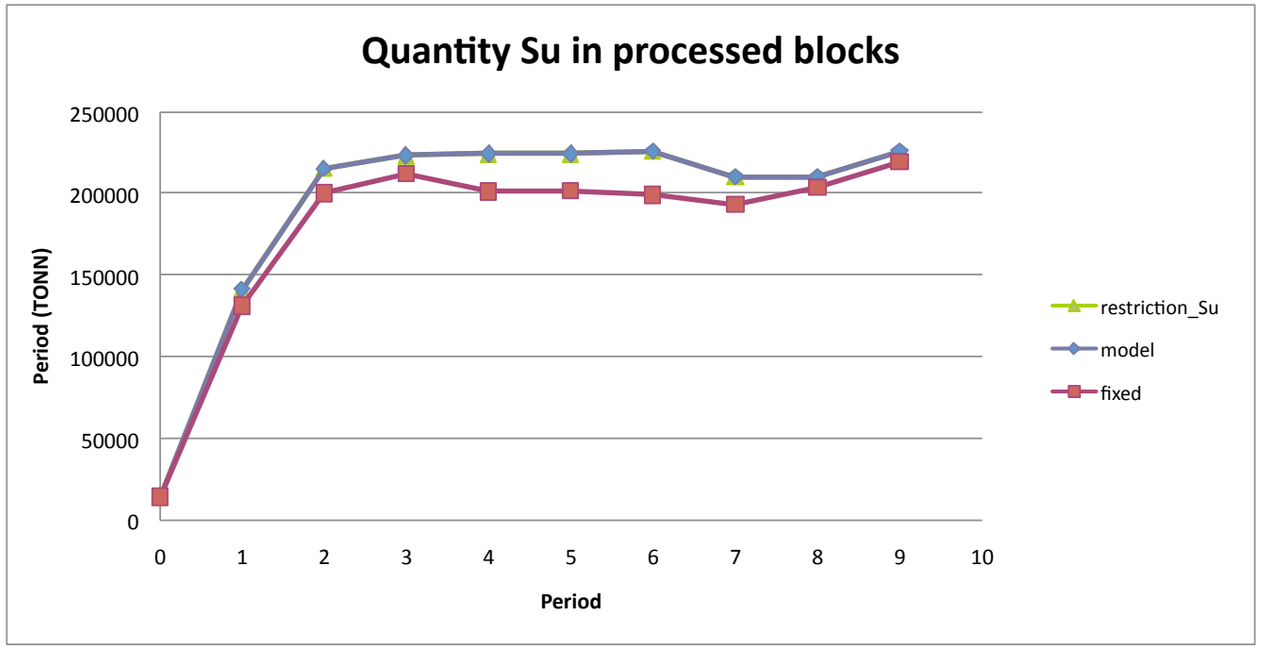

Average grades in processed blocks
Fixed
average grade
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0 5 10 15 20 25 30 35 40 45
Period
Cu
Au
As

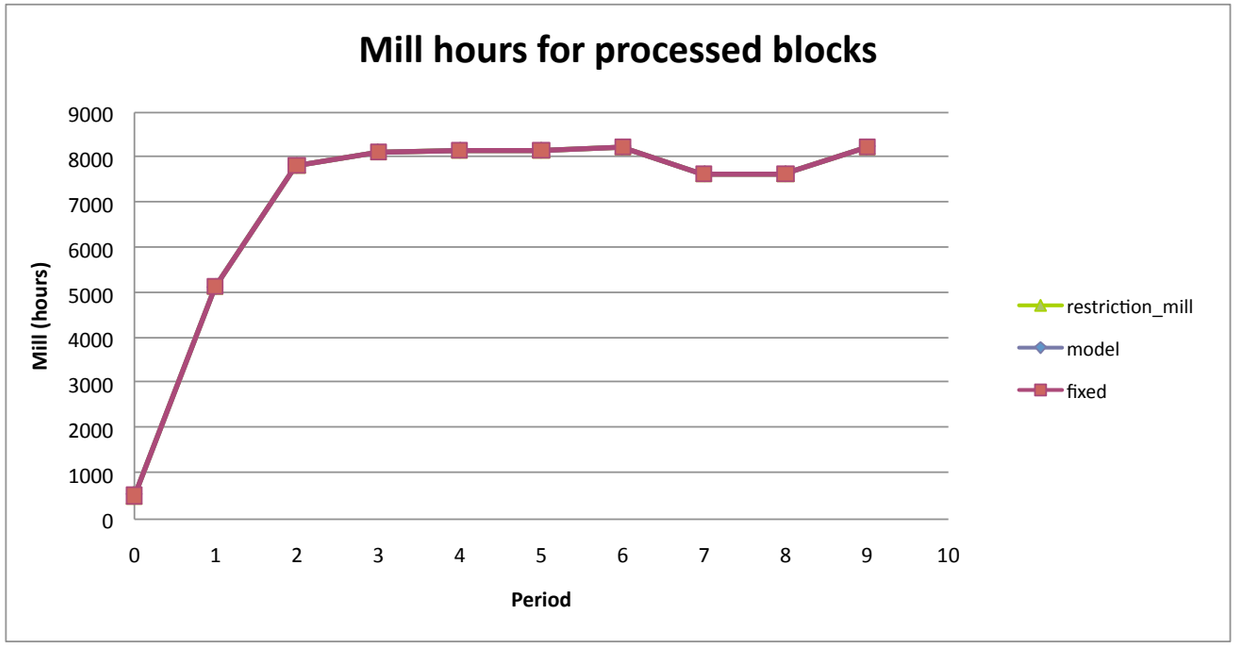

Average grades in processed blocks
Model
average grade
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0 5 10 15 20 25 30 35 40 45
Period
Cu
Au
As

NCL


Extracted/processed weight
Fixed
Weight (TONN)
16000000
14000000
12000000
10000000
80000000
60000000
40000000
20000000
0
0
2
4
6
8
10
Period
processed
extracted
restriction_processed
restriction_extracted


Extracted/Processed weight
Model
Weight (TONN)
16000000
14000000
12000000
10000000
80000000
60000000
40000000
20000000
0
0
2
4
6
8
10
Period
restriction_processed
restriction_extracted
processed
extracted


Quantity Su in processed blocks
Period (TONN)
250000
200000
150000
100000
50000
0
0
1
2
3
4
5
6
7
8
9
10
Period
restriction_Su
model
fixed


Mill hours for processed blocks
Mill (hours)
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
1
2
3
4
5
6
7
8
9
10
Period
restriction_mill
model
fixed

Uncertainty in Open-pit Mine Planning

- Grade Uncertainty
- Ore Prices

$$x_b^e = \begin{cases} 1 & \text{if } b \text{ is extracted} \\ 0 & \text{if not} \end{cases}$$

$$x_b^p = \begin{cases} 1 & \text{if } b \text{ is processed} \\ 0 & \text{if not} \end{cases}$$

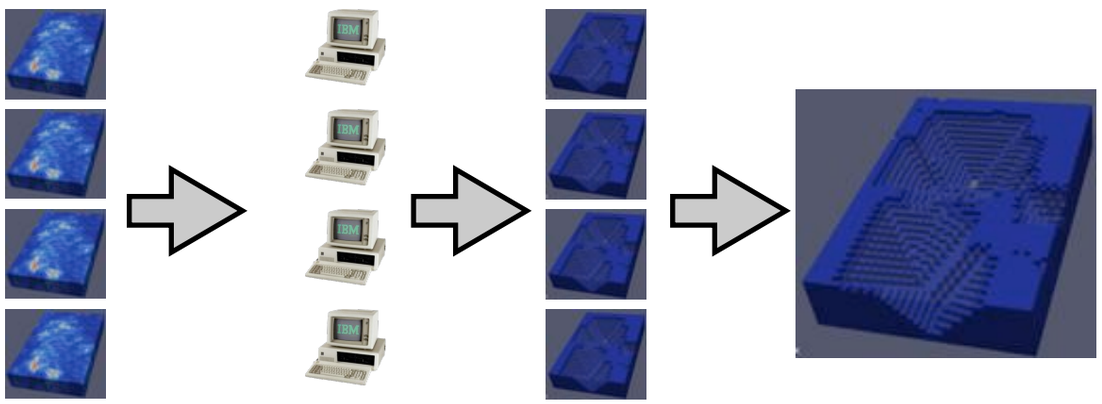
$$x_b^p \leq x_b^e \ \forall b$$

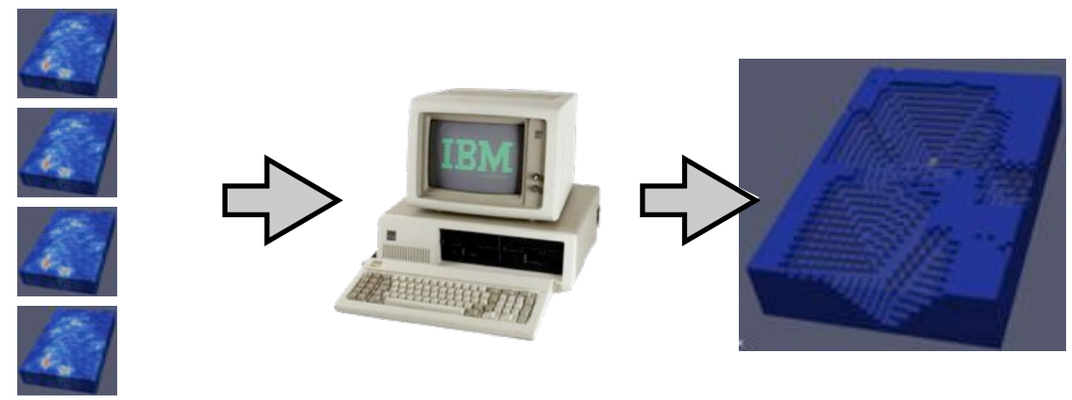
Objective Function:

$$\Pi_\rho(x^e, x^p) = (p^p \cdot \rho - c^p)x^p - c^e x^e$$

$$(\rho_b := \text{grade of ore in block } b)$$

Scenario approaches

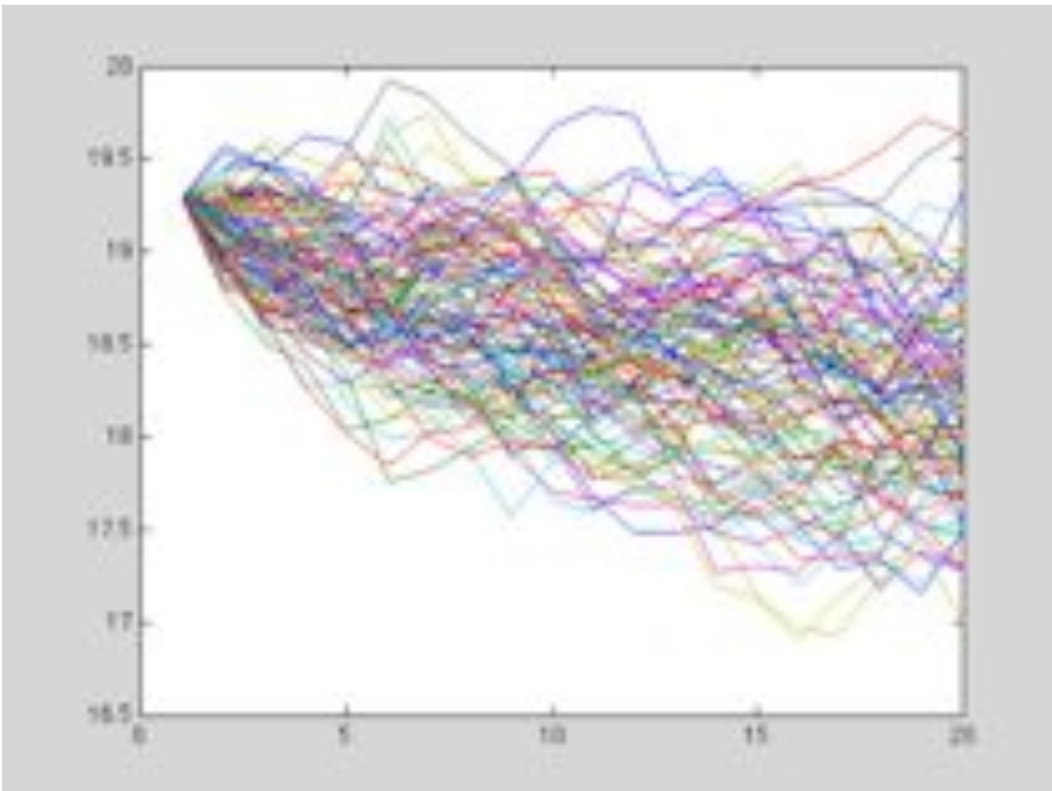




- Most common in mining literature
- Dimitrakopoulos et al.

- PH on ore prices
- DP on grade uncertainty
- Risk minimization
- Robust programming

Our approach for ore prices uncertainty



mean reversion process
of ore prices

(Ornstein–Uhlenbeck process)

$$p^p \in \mathcal{U} = \{p^p | \exists u, u'u \leq 1, p^p = \bar{p} + \Sigma u\}$$

A second-order conic constraint

$$\begin{array}{lll} (P) & \max & cx \\ & \text{s.t} & Ax \leq b \\ & & Dx \leq d \end{array}$$

"Easy"

"Hard"

Other uses of BZ Algorithm

- Value-at-Risk

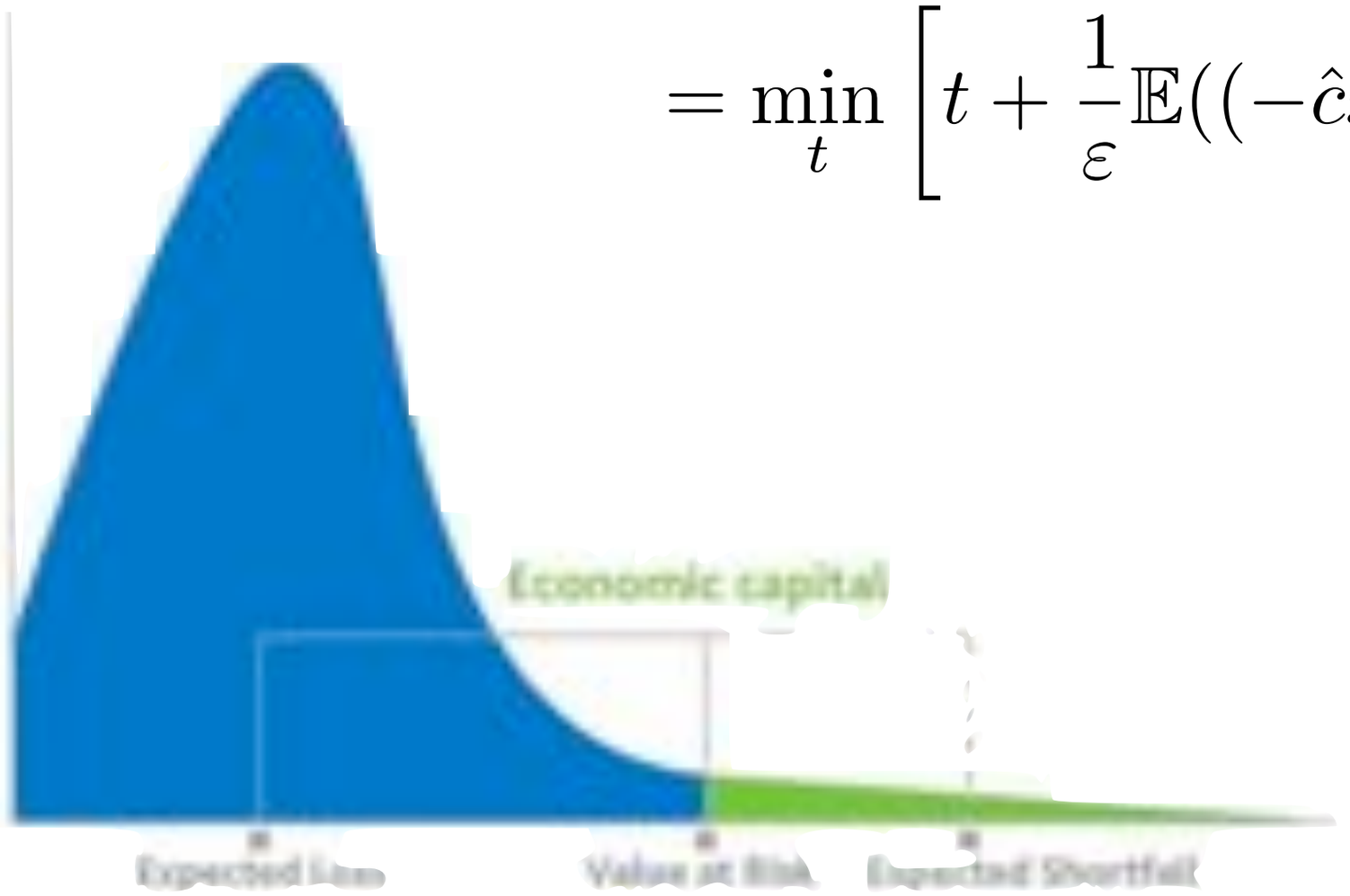
$$\mathrm{VaR}_\varepsilon(-\hat{c}x) = \min_t \{t : F_{-\hat{c}x}(t) \geq 1 - \varepsilon\}$$

- Conditional Value-at-Risk (expected shortfall)

$$\mathrm{CVaR}_\varepsilon(-\hat{c}x) = \mathbb{E}(-\hat{c}x | -\hat{c}x \geq \mathrm{VaR}_\varepsilon(-\hat{c}x))$$

$$= \min_t \left[t + \frac{1}{\varepsilon}\mathbb{E}((-\hat{c}x - t)^+) \right]$$

Convex function!



CVaR formulation

$$\min \ \mathrm{CVaR}_\varepsilon(-\hat{c}x)$$
$$s.t.\ Ax = b$$
$$x \geq 0$$

SAA

probability of scenario i

$$\min t + \frac{1}{\varepsilon}\sum_{i=1}^{N} p_i \eta_i$$
$$s.t.\ Ax = b$$
$$\eta_i \geq -c^i x - t \quad \forall i$$
$$x, \eta \geq 0$$

scenario i

$$\mathrm{CVaR}_\varepsilon(-\hat{c}x) = \min_t \left[t + \frac{1}{\varepsilon}\mathbb{E}((-\hat{c}x - t)^+) \right]$$

Our idea

$$\min t + \frac{1}{\varepsilon}\sum_{i=1}^{N} p_i \eta_i$$
$$s.t.\ Ax = b$$
$$c^i x + t + \eta_i \geq 0 \quad \forall i$$
$$x, \eta \geq 0$$

$\tilde{x} : A\tilde{x} = b$

Upper bound

$$\min t + \frac{1}{\varepsilon}\sum_{i=1}^{N} p_i \eta_i$$
$$s.t.\ c^i \tilde{x} + t + \eta_i \geq 0 \quad \forall i$$
$$\eta \geq 0$$

Easy to solve!

$\tilde{x}$

$\{P^j\}$

$\mathcal{N} = \{P^j\}_{j=1}^{k}$

Lower bound

$$\hat{p}_j = \sum_{i \in P^j} p_i \qquad \bar{c}^j = \frac{1}{\hat{p}_j}\sum_{i \in P^j} p_i c^i$$

$$\min t + \frac{1}{\varepsilon}\sum_{j=1}^{k} \hat{p}_j \hat{\eta}_j$$
$$s.t.\ Ax = b$$
$$\bar{c}^j x + t + \hat{\eta}_j \geq 0 \quad \forall j \in \{1, \ldots, k\}$$
$$x, \hat{\eta} \geq 0$$

Smaller problem!

Dual problems

$$\tilde{x} : A\tilde{x} = b$$

~~Upper~~ Lower bound

$$\max -\frac{1}{\varepsilon}\sum_{i=1}^{N} p_i \lambda_i c^i \tilde{x}$$

$$\text{s.t.} \sum_{i=1}^{N} p_i \lambda_i = \varepsilon$$

$$0 \le \lambda_i \le 1 \, \forall i$$

$$\lambda^*_{(i)} = 1 \; i \le i^*$$

$$\lambda^*_{(i+1)} = \gamma$$

$$\lambda^*_{(i)} = 0 \; i > i^* + 1$$

$$\tilde{x}$$

$$\max y^t b$$

$$s.t.\, y^t A + \frac{1}{\varepsilon}\sum_{j=1}^{k} \hat{p}_j \hat{\lambda}_j \bar{c}^j \le 0$$

$$\sum_{j=1}^{k} \hat{p}_j \hat{\lambda}_j = \varepsilon$$

$$0 \le \hat{\lambda}_j \le 1 \quad \forall j \in \{1 \ldots K\}$$

$$\max y^t b$$

$$s.t.\, y^t A + \frac{1}{\varepsilon}\sum_{i=1}^{N} p_i \lambda_i c^i \le 0$$

$$\sum_{i=1}^{N} p_i \lambda_i = \varepsilon$$

$$0 \le \lambda_i \le 1 \quad \forall i$$

$$\mathcal{N} = \{P^j\}_{j=1}^{k}$$

~~Lower~~ Upper bound

Additional constraints:

$$\lambda_i = \lambda_{i'} \; \forall i, i' \in P^j$$

Computational results for large N

	$N = 100,000$			$N = 1,000,000$			$N = 10,000,000$		
Name	time	Iter.	$\lvert\mathcal{N}\rvert$	time	Iter.	$\lvert\mathcal{N}\rvert$	time	Iter.	$\lvert\mathcal{N}\rvert$
adlittle	85	17.3	5232.8	2080	19.8	32763.5	56684	22.3	218034.0
afiro	2	2.0	2.0	22	2.0	2.0	210	2.0	2.0
agg	45	13.8	924.8	568	15.8	4272.3	6270	17.5	19766.3
bandm	57	14.8	1353.8	720	17.0	7072.8	7854	18.8	29525.3
beaconfd	21	7.8	44.5	260	8.8	79.3	2462	8.5	91.0
boeing2	54	15.0	1579.5	653	16.8	8781.8	7318	18.3	37770.0
bore3d	5	2.0	2.0	54	2.3	2.5	488	2.3	2.5
brandy	2	2.5	3.0	23	2.5	3.0	225	2.5	3.0
capri	31	14.8	2517.0	399	16.3	6927.0	4674	17.8	15675.3
e226	76	16.5	3576.3	1013	18.8	23025.3	13197	21.5	149284.5
etamacro	10	4.8	12.0	135	5.5	20.5	1040	4.8	15.5
fffff800	14	10.5	118.5	158	10.8	252.8	1499	10.3	354.0
ganges	37	12.8	815.0	471	14.8	4581.5	5380	17.3	30458.5
grow15	8	4.8	87.3	99	5.3	412.0	831	5.5	1740.5
grow22	21	9.5	135.8	218	9.5	493.8	1999	9.5	2329.3
grow7	16	9.5	146.5	164	9.3	199.5	1283	8.3	284.8
israel	38	13.8	544.5	440	14.5	2331.0	4745	16.3	9594.3
kb2	3	3.3	5.0	47	4.3	9.0	435	4.5	12.5
lotfi	0	0.0	0.0	50	5.0	14.8	535	5.5	17.8
perold	3	2.5	3.0	30	2.8	3.5	268	2.8	4.0
pilot.ja	4	2.8	3.5	32	2.8	3.5	300	2.8	3.5

grow7	16	9.5	146.5	164	9.3	199.5	1283	8.3	284.8
israel	38	13.8	544.5	440	14.5	2331.0	4745	16.3	9594.3
kb2	3	3.3	5.0	47	4.3	9.0	435	4.5	12.5
lotfi	0	0.0	0.0	50	5.0	14.8	535	5.5	17.8
perold	3	2.5	3.0	30	2.8	3.5	268	2.8	4.0
pilot.ja	4	2.8	3.5	32	2.8	3.5	300	2.8	3.5
pilot	45	14.3	1488.5	467	16.5	8316.5	5257	18.5	47483.0
pilot4	3	3.8	6.0	37	3.5	5.5	342	3.5	5.8
pilotnov	47	13.3	2589.8	539	14.8	12104.3	5199	16.3	19192.8
recipe	5	2.0	2.0	50	2.0	2.0	454	2.0	2.0
sc105	1	2.0	2.0	13	2.0	2.0	130	2.0	2.0
sc205	1	2.0	2.0	13	2.0	2.0	130	2.0	2.0
sc50a	1	2.0	2.0	14	2.0	2.0	130	2.0	2.0
sc50b	1	2.0	2.0	13	2.0	2.0	129	2.0	2.0
scagr7	20	6.8	29.8	231	7.5	78.8	2225	7.8	276.0
scfxm1	5	3.0	4.0	52	3.0	4.0	466	3.0	4.0
scfxm2	6	3.0	4.0	63	3.0	4.0	557	3.0	4.0
scfxm3	8	3.5	6.3	81	3.5	7.0	700	3.3	5.0
share1b	16	8.3	21.8	189	8.5	27.3	1167	5.8	22.0
share2b	16	9.0	36.3	194	9.8	67.8	1073	5.8	28.8
stair	1	2.0	2.0	14	2.0	2.0	129	2.0	2.0
standata	3	3.0	4.0	38	3.0	4.0	255	2.5	3.0
standgub	3	3.0	4.0	38	3.0	4.0	257	2.5	3.0
standmps	2	2.0	2.0	23	2.0	2.0	209	2.0	2.0
stocfor1	19	10.5	78.3	206	10.8	178.0	1305	7.5	139.5
tuff	2	2.0	2.0	19	2.0	2.0	172	2.0	2.0
vtp.base	3	3.0	4.0	36	3.0	4.3	213	2.3	2.5
woodw	5	4.5	12.0	49	4.3	12.3	287	3.0	4.5

Thanks...