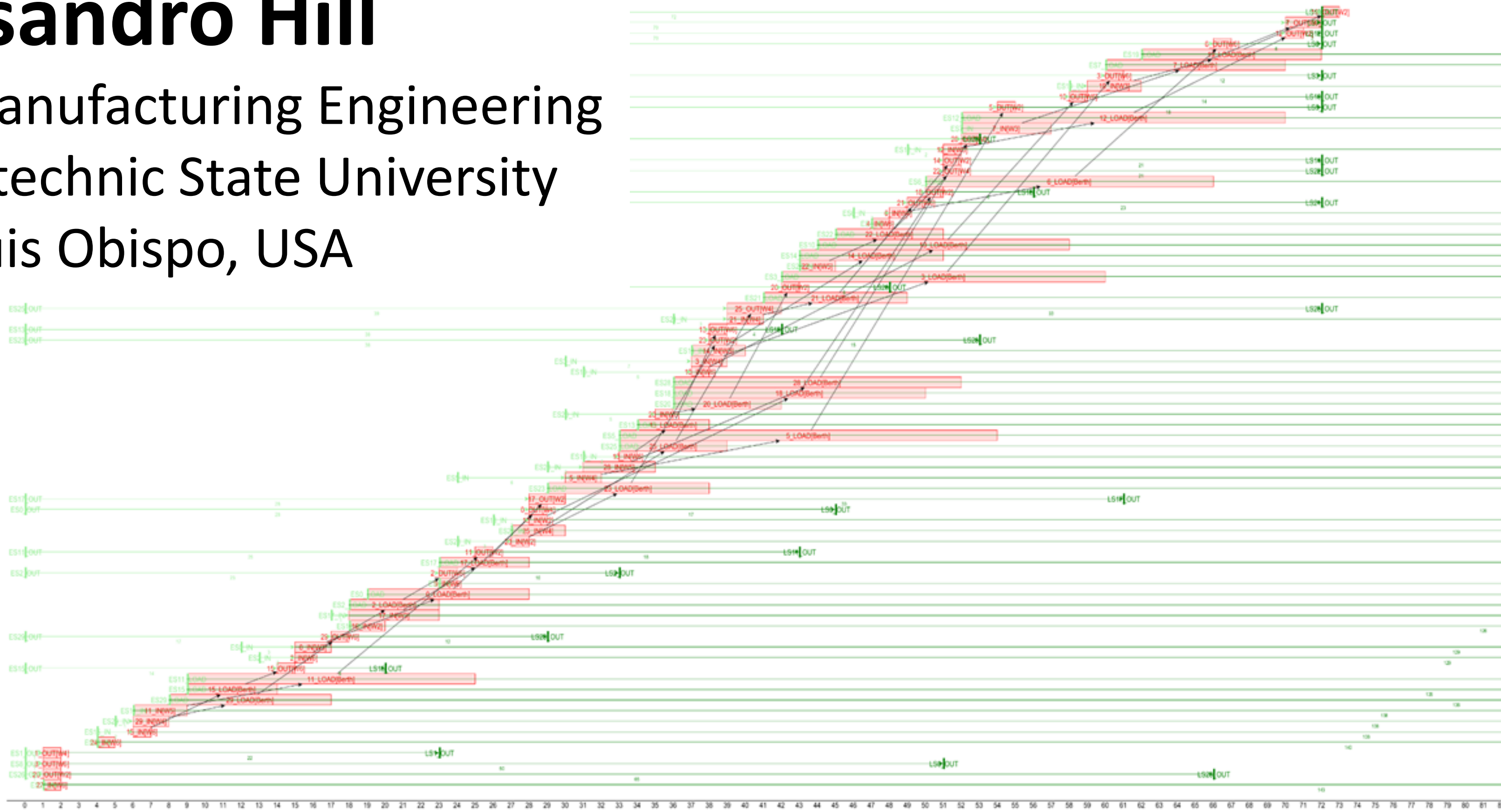
Two Successful Applications of Resource-Constrained Project Scheduling: Ship Waterway Traversals at Maritime ports and Strategic Underground Mining

Alessandro Hill

Industrial and Manufacturing Engineering

California Polytechnic State University

San Luis Obispo, USA



Università della Calabria, 20.11.2023

OUTLINE

0. Resource-Constrained Project Scheduling Problems (RCPSPs)
1. Ship Waterway Traversals at Maritime Ports

   a. Ship Scheduling and RCPSPs
   b. Reformulation and Integer Programming Approach
   c. Computational Results
2. Strategic Underground Mining

   a. Mine Planning and RCPSPs
   b. Hybrid Mathematical Programming and Constraint Programming Algorithms
   c. Computational Results
3. Summary and Outlook

OUTLINE

0. Resource-Constrained Project Scheduling Problems (RCPSPs)

1. Ship Waterway Traversals at Maritime Ports

   a. Ship Scheduling and RCPSPs
   b. Reformulation and Integer Programming Approach
   c. Computational Results

2. Strategic Underground Mining

   a. Mine Planning and RCPSPs
   b. Hybrid Mathematical Programming and Constraint Programming Algorithms
   c. Computational Results

3. Summary and Outlook

The Resource-Constrained Project Scheduling Problem (RCPSP)

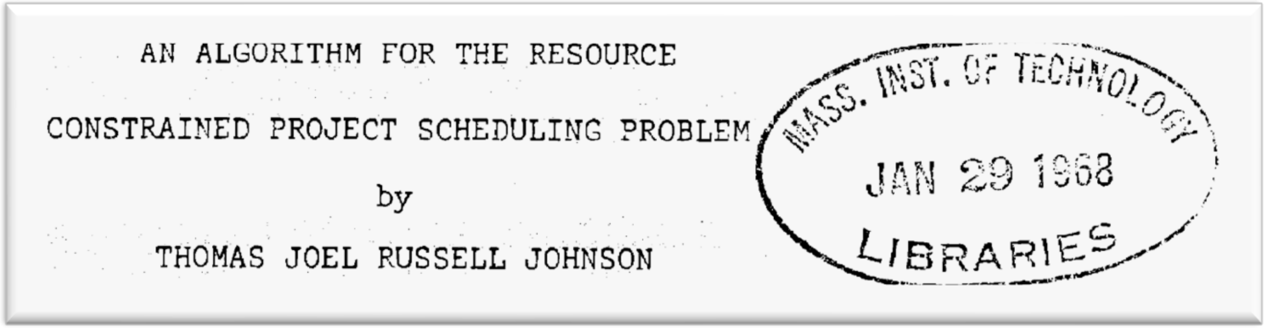
Given:

- Discrete time horizon $T = \{0, \dots, h\}$
- Renewable resources R
  - Per-period availability q_r $(\forall r \in R)$
- Jobs $J = \{1, \dots, n\}$
  - Duration d_j $(\forall j \in J)$
  - Per-period resource utilization $u_{j,r}$ $(\forall j \in J, r \in R)$
- Precedence relation $A \subseteq J \times J$

RCPSP:
Find a schedule S for jobs in J in T such that

- j does not start before i ends $(\forall (i,j) \in A)$, and
- capacity utilization in period $[t-1, t]$ does not exceed q_r $(\forall r \in R, t \in T \setminus \{0\})$

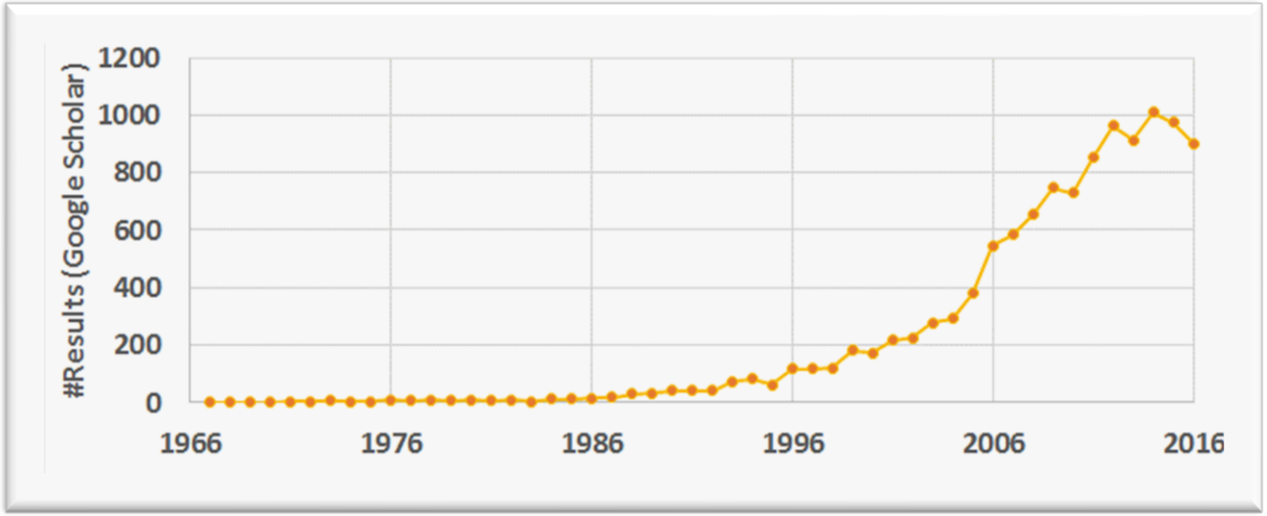
minimizing the project makespan (=duration).


AN ALGORITHM FOR THE RESOURCE

CONSTRAINED PROJECT SCHEDULING PROBLEM

by

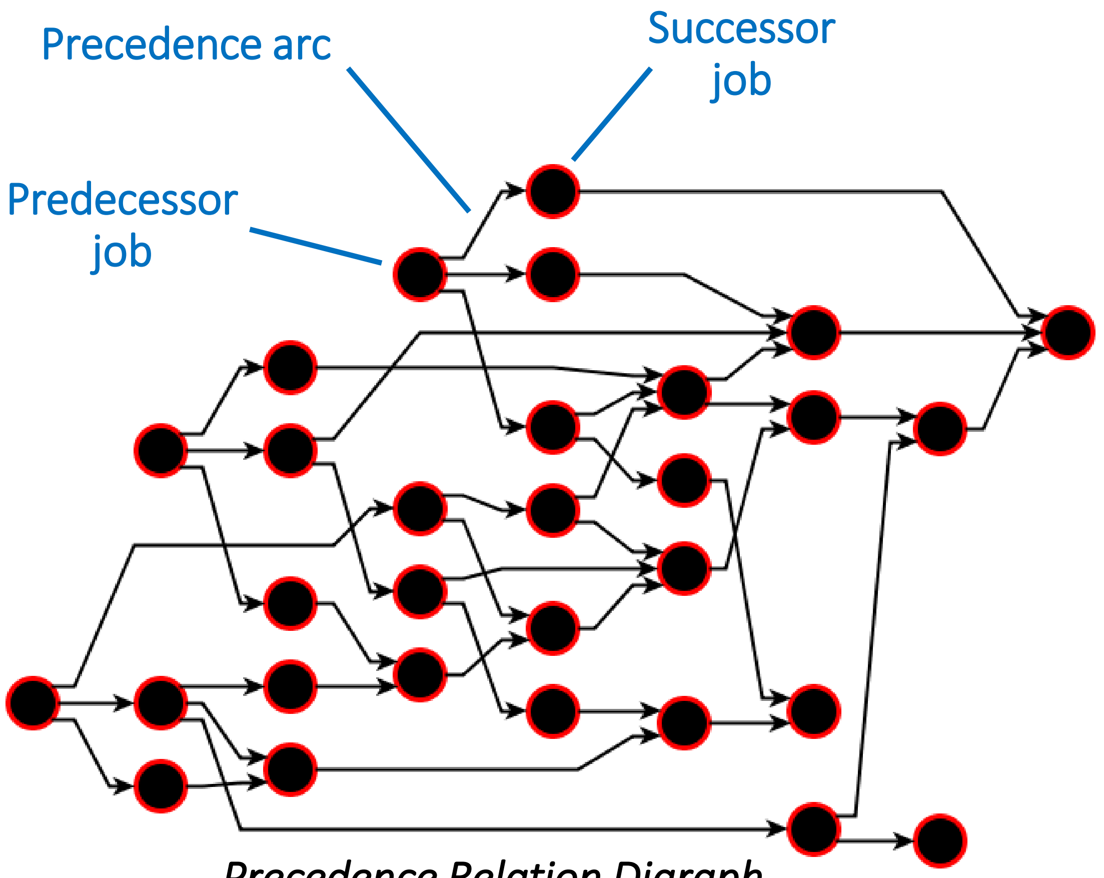
THOMAS JOEL RUSSELL JOHNSON



***Note:** RCPSP extends the classic project scheduling problem by considering resources.*

RCPSP

30 jobs

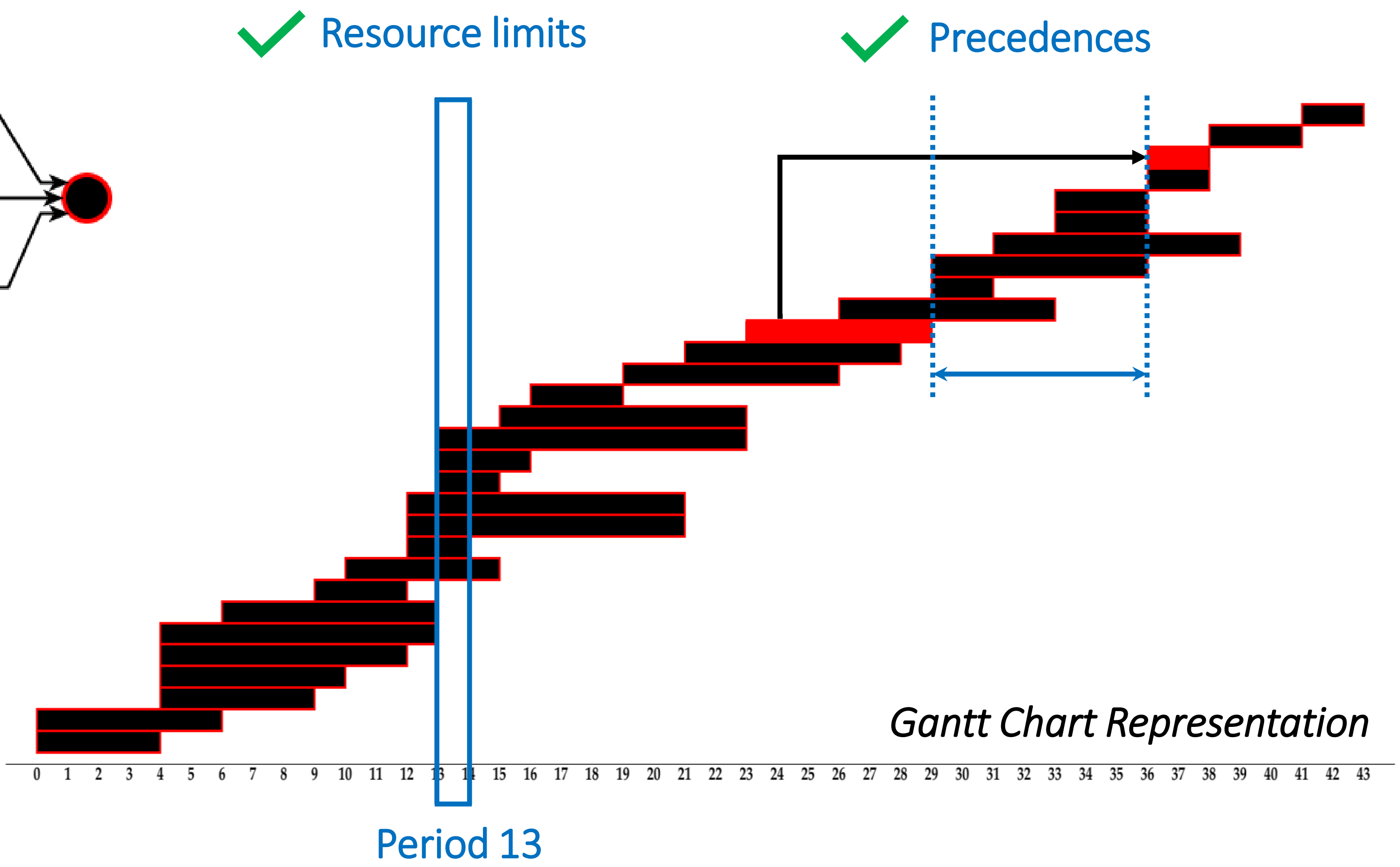


Precedence Relation Digraph

Assumption: *Digraph* (J, A) *is acyclic!*

Solution Schedule

43 periods



Gantt Chart Representation

OUTLINE

0. Resource-Constrained Project Scheduling Problems (RCPSPs)

1. Ship Waterway Traversals at Maritime Ports

   a. Ship Scheduling and RCPSPs
   b. Reformulation and Integer Programming Approach
   c. Computational Results

2. Strategic Underground Mining

   a. Mine Planning and RCPSPs
   b. Hybrid Mathematical Programming and Constraint Programming Algorithms
   c. Computational Results

3. Summary and Outlook

A Multi-Mode Resource-Constrained Project Scheduling Reformulation for the Waterway Ship Scheduling Problem

CAL POLY

Alessandro Hill

Industrial and Manufacturing Engineering
California Polytechnic State University
San Luis Obispo, USA

UNIVERSIDAD ADOLFO IBÁÑEZ

Marcos Goycoolea

School of Business
Universidad Adolfo Ibáñez
Santiago, Chile

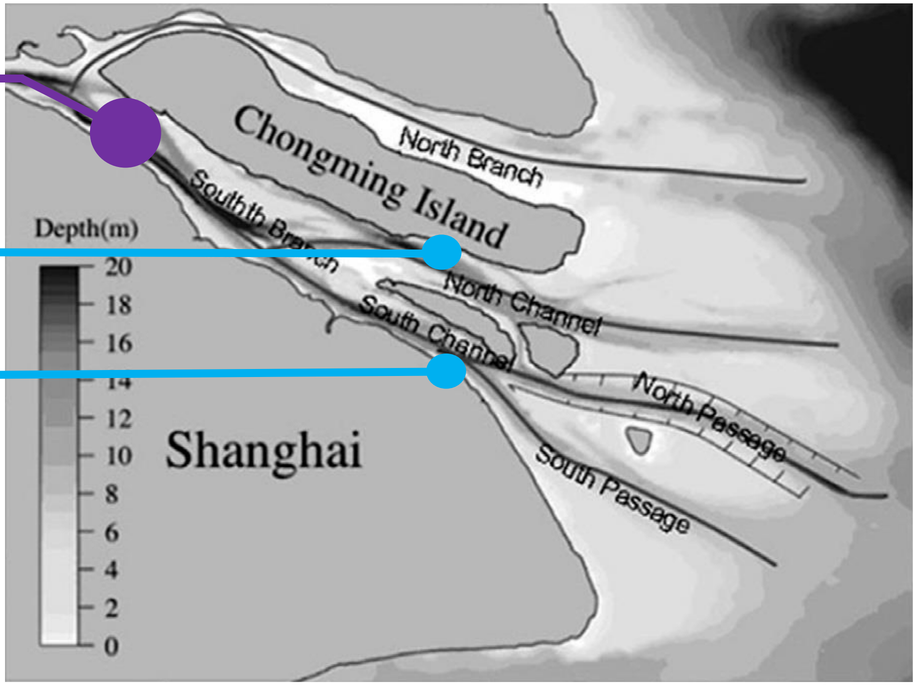
UHH Universität Hamburg

Eduardo Lalla-Ruiz and Stefan Voβ

Institute of Information Systems (IWI)
University of Hamburg
Hamburg, Germany

Waterway Ship Scheduling

- **Vessels arriving at and departing from container terminals**
- **Multiple waterways can be traversed**
- **Short-term slot assignment by port authorities:** 24 hours



Yangtze Estuary

Goals

- → **Reduce waiting times, traffic congestions, emissions**
- → **Smoothen port operations and ship routing**
- → **Improve port accessibility**

Restrictions

Geographical: width, tide-dependent depth, etc.

Operational: opposing traffic, clearance distance, etc.

Related problems in the literature

- **Canal ship traffic control** (e.g., Panama, Kiel)
  Lübbecke (2015)
- **Lock scheduling** (e.g., Mississippi, Danube)
  Campbell et al. (2007)
- **Bi-direction path traffic** Disser et al. (2015)

Waterway Ship Scheduling Problem (WSSP)*

Given

- **Waterways:** width, time-dependent depth
- **Arriving ships:** ETA
- **Departing ships:** ETD

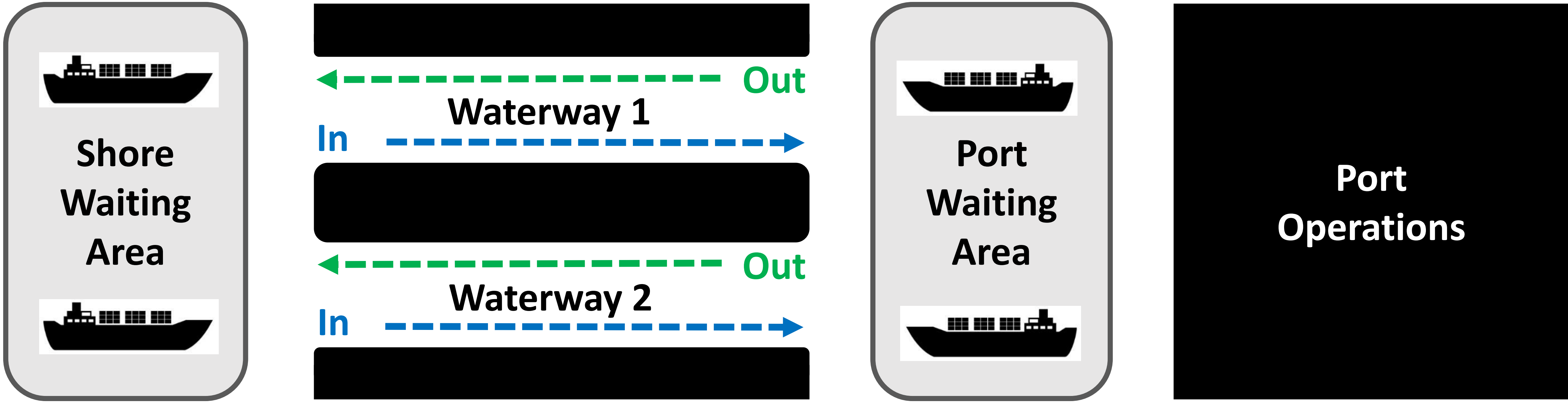
Width, draft, speed

→ **Minimize the total vessel turnaround time**

Existing approaches*

- **Mixed-Integer-Programming:** Based on a compact Vehicle Routing Problem formulation
- **Simulated Annealing**

⟹ **NP-hard**



*Lalla-Ruiz et al. (2016) "The Waterway Ship Scheduling Problem", *Transportation Research Part D*

The Multi-Mode Resource-Constrained Project Scheduling Problem (MM-RCPSP)

Given

- **Discrete time horizon**
- **Resources:** period-dependent availability
- **Jobs**
- **Modes for jobs $\boldsymbol{M}$:** mode dependent duration d_m and resource consumption $u_{m,r}$
- **Precedences:** "Job A has to finish before job B starts"
- **Earliest job start time $\boldsymbol{EST_j}$ and latest job end time $\boldsymbol{LET_j}$**

Objective

Find a schedule that

→ schedules every job in exactly one mode

→ is resource, time-window and precedence feasible

→ **minimizes** the sum of job end times minus ESTs

Integer Programming Formulation (MM-RCPSP)

Binary **job-mode-end-variables** $x_{j,m,t} \in \{0,1\} \quad \forall\, j \in J,\ m \in M_j,\ t \in T.$

$x_{j,m,t} = 1$ If job j ends at time t in mode m; 0 otherwise.

$$min \quad \sum_{j\in J} \sum_{m\in M_j} \sum_{t\in T} (t - EST_j) x_{j,m,t}$$

Tournaround Time Objective

$$subject\ to \quad \sum_{m\in M_j} \sum_{t\in T} x_{j,m,t} \;=\; 1 \quad \forall\, j \in J,$$

Unique Job Mode

$$\sum_{j\in J} \sum_{m\in M_j} \sum_{t<t'\leq t+d_m} u_{m,r} x_{j,m,t'} \;\leq\; q_{r,t} \quad \forall\, r \in R,\ t \in T,$$

Resource Capacities

$$EST_j \;\leq\; \sum_{m\in M_j} \sum_{t\in T} (t - d_m) x_{j,m,t} \;\leq\; LST_j \quad \forall\, j \in J,$$

Earliest Start / Latest End Times

Zhu and Bard (2006) “A Branch-and-Cut Procedure for the Multimode Resource-Constrained Project-Scheduling Problem”, *INFORMS Journal on Computing*

Waterway Ship Scheduling using MM-RCPSPs

Reformulation

MM-RCPSP		WSSP
Jobs	⟷	Incoming, outgoing ships
Modes	⟷	Waterways
Resources	⟷	Waterway width, depth, parallel traffic

Method

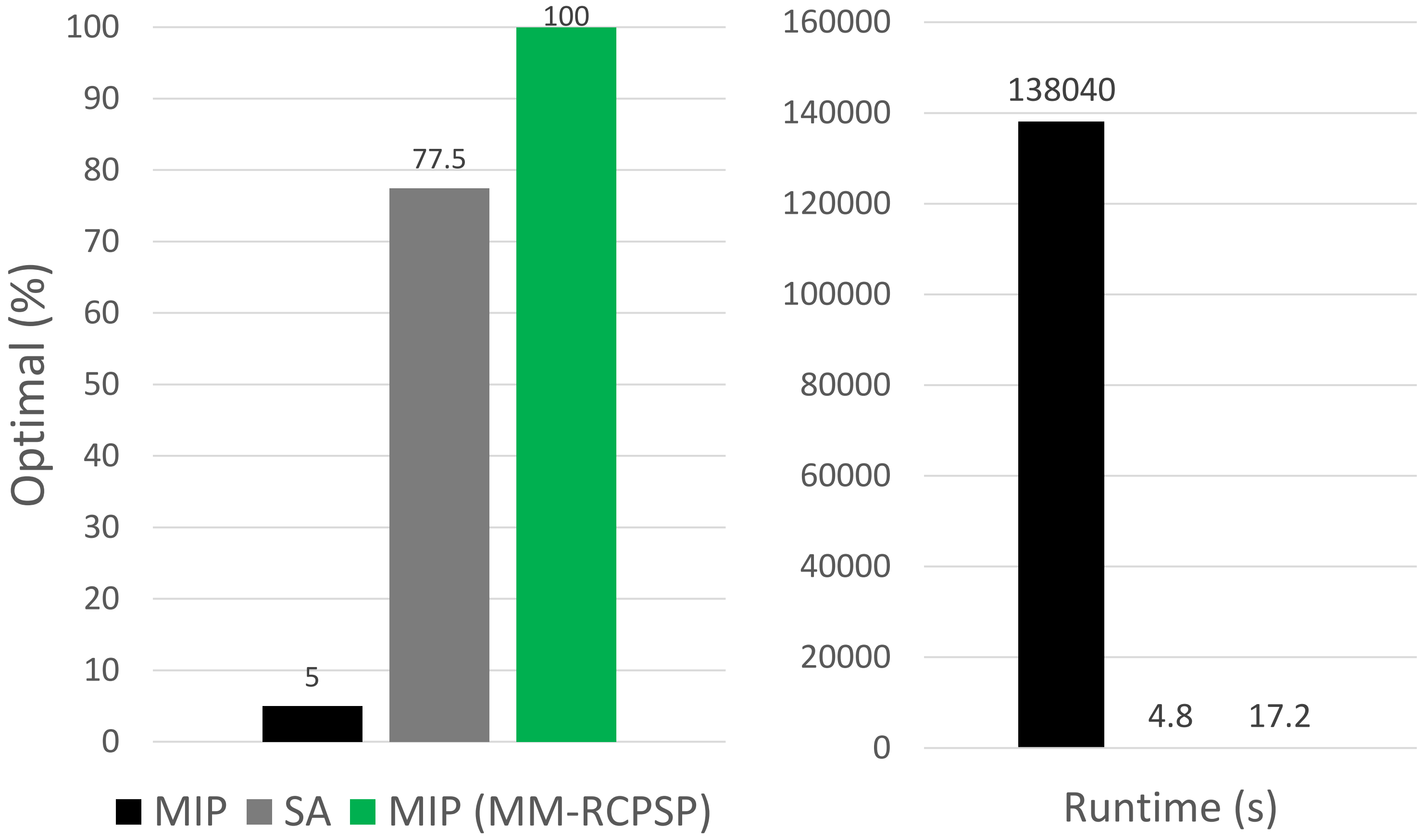
Integer Programming

- **MM-RCPSP formulation**
- **ILOG CPLEX 12.71**

Computational Results (WSSP)

- 40 instances* (Shanghai-based)
- Time discretization: 30 minutes
- CPU Intel i7 2.0 GHz

#	Ships (IN)	Ships (OUT)	Waterways
5	15	15	2
5	15	15	4
5	20	20	2
5	20	20	4
5	25	25	2
5	25	25	4
5	30	30	2
5	30	30	4



→ **Solved all instances to optimality**

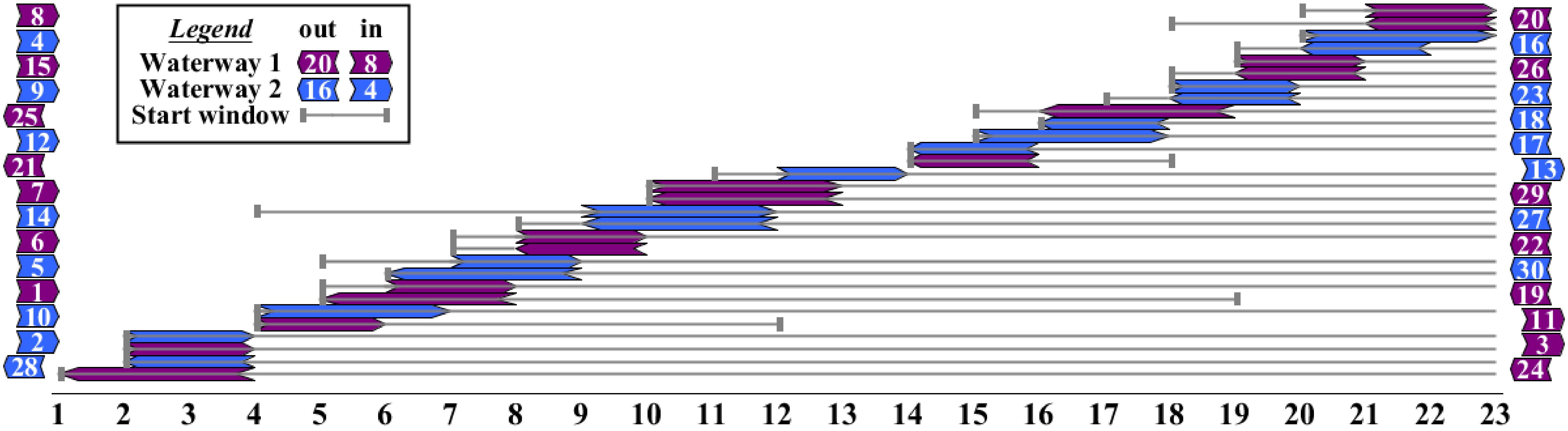
→ **82% solved at the root node**

→ **< 5 branching nodes**

*Lalla-Ruiz et al. (2016) "The Waterway Ship Scheduling Problem", *Transportation Research Part D*

Solution Schedule (WSSP)

30 ships, 2 waterways



Summary

- Waterway ship scheduling can be modelled using MM-RCPSPs
- Mathematical programming can solve realistic WSSPs efficiently to optimality

Future work:

- Incorporated berthing operations
- Develop cutting planes for mathematical programming and heuristics
- Apply constraint programming

OUTLINE

0. Resource-Constrained Project Scheduling Problems (RCPSPs)
1. Ship Waterway Traversals at Maritime Ports

   a. Ship Scheduling and RCPSPs
   b. Reformulation and Integer Programming Approach
   c. Computational Results
2. Strategic Underground Mining

   a. Mine Planning and RCPSPs
   b. Hybrid Mathematical Programming and Constraint Programming Algorithms
   c. Computational Results
3. Summary and Outlook

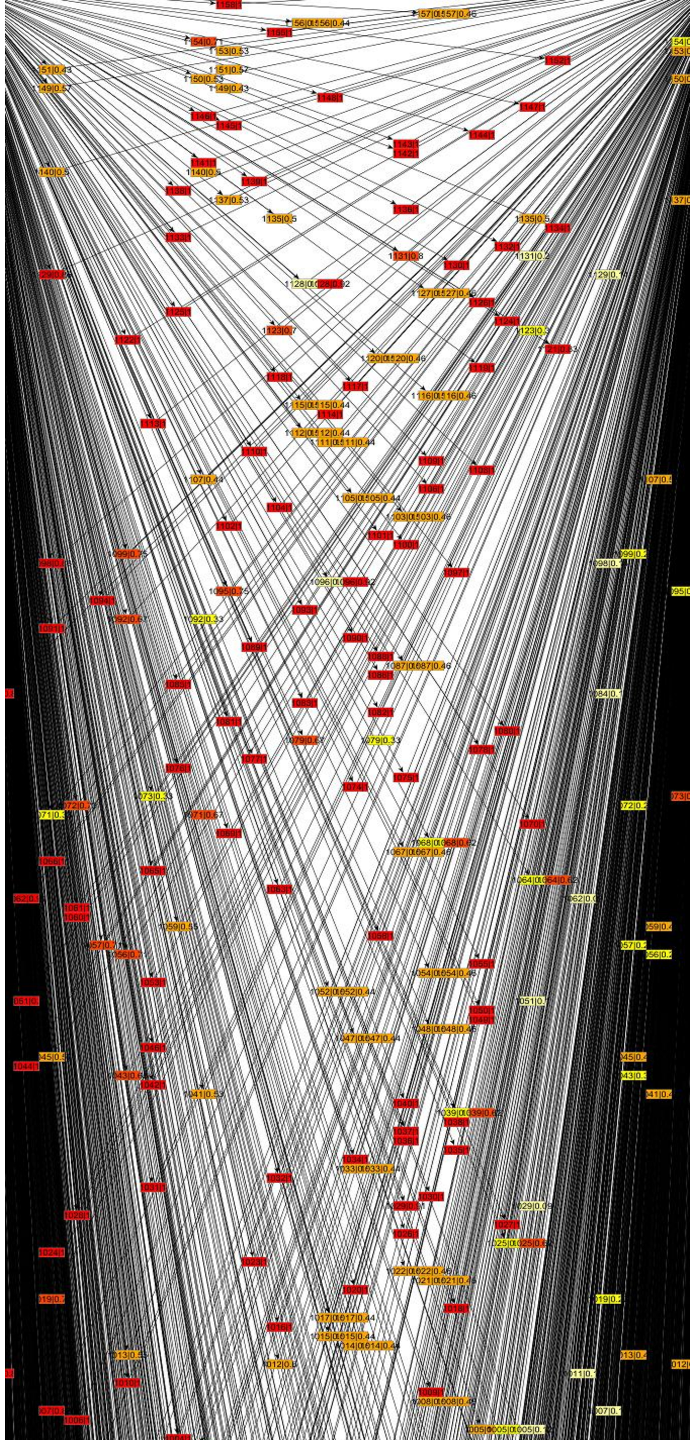
Optimization Strategies for Resource-Constrained Project Scheduling Problems in Underground Mining

Alessandro Hill
Industrial and Manufacturing Engineering
California Polytechnic State University, USA

Andrea J. Brickey
Mining Engineering and Management
South Dakota School of Mines, USA

Alexandra Newman
Mechanical Engineering
Colorado School of Mines, USA

Italo Cipriano & Marcos Goycoolea
School of Business
Universidad Adolfo Ibáñez, Chile



Strategic Mine Planning

Given:
- **Block model** of the ore body
- **Mining method** (Open pit, Open Stoping, Block Caving, Room & Pillar, . . .)

Goals

⟹ **Find long-term operational schedule**
Activities: Excavation, hauling, backfilling . . .

⟹ **Respect operational requirements**
Mill capacity, stability, safety, ventilation, . . .

⟹ **Maximize mine lifetime profit**

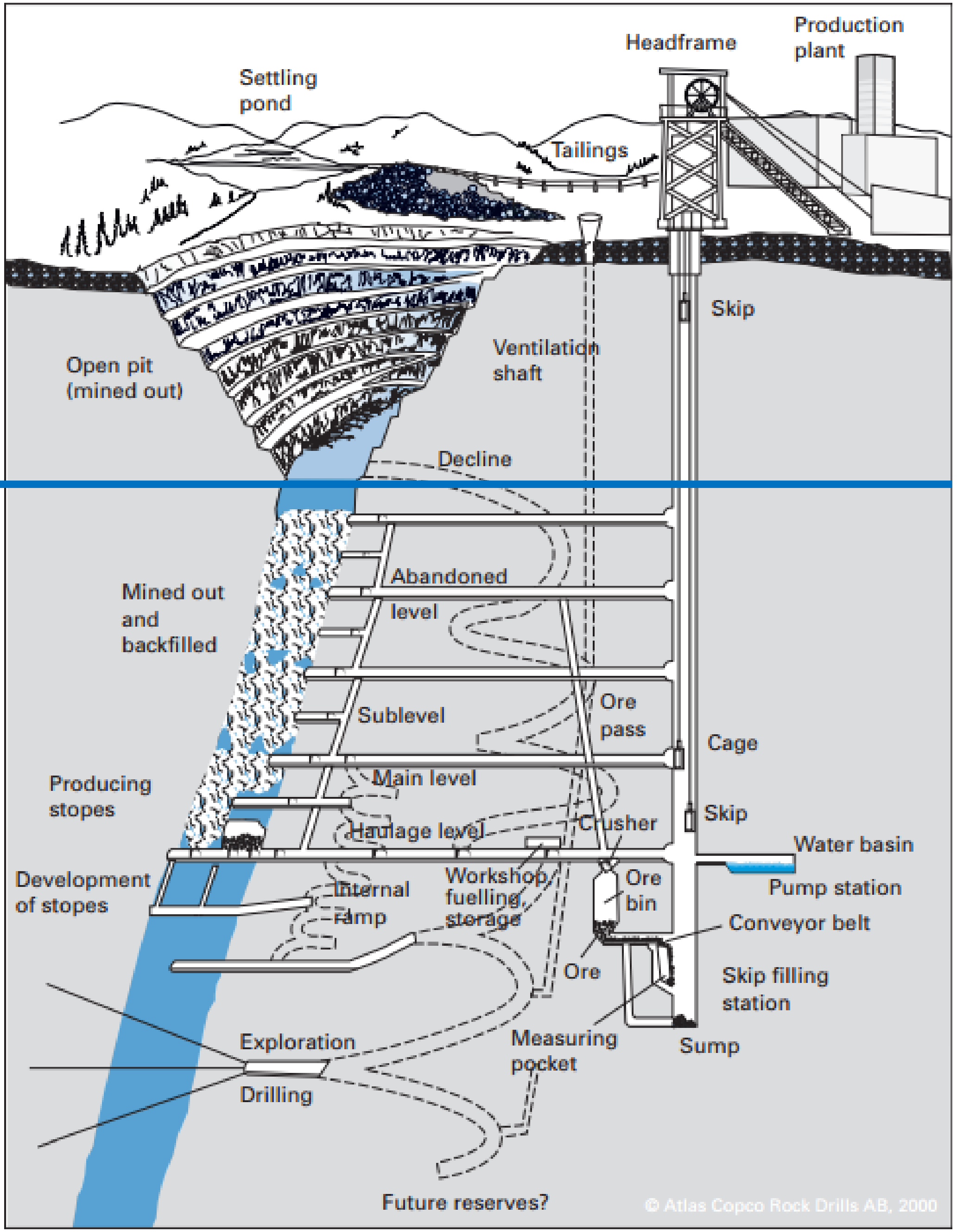
Challenges

➡ **Very-large models**
Up to 30,000 activities
Up to 50 years planning horizon

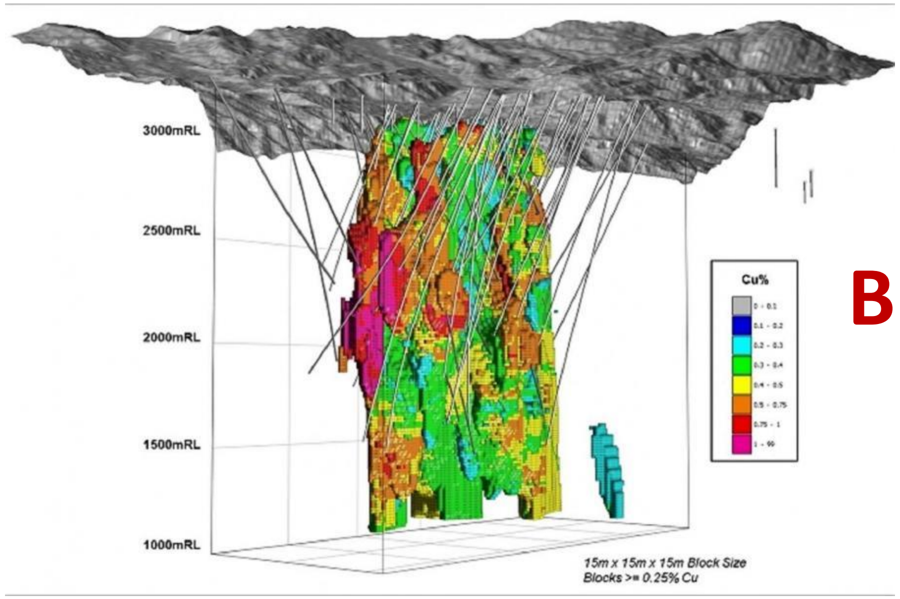
➡ **Complex production constraints**

Open Pit

Underground

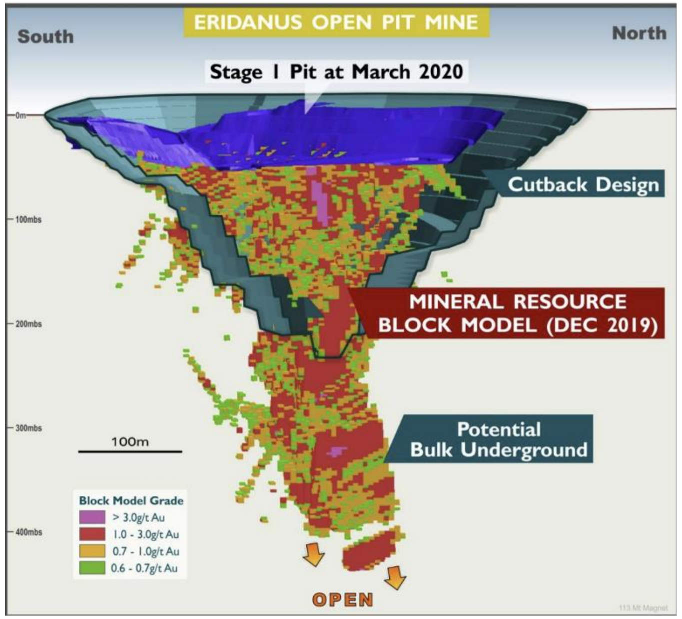


Block Models

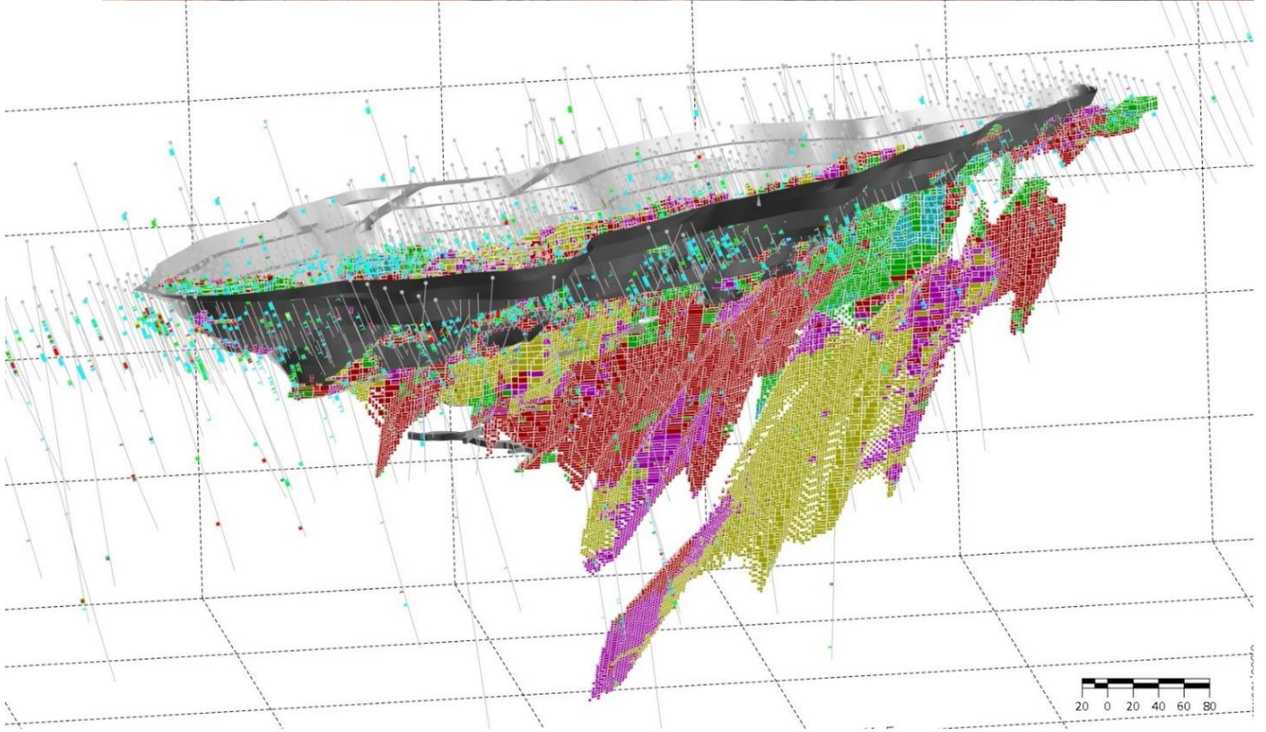


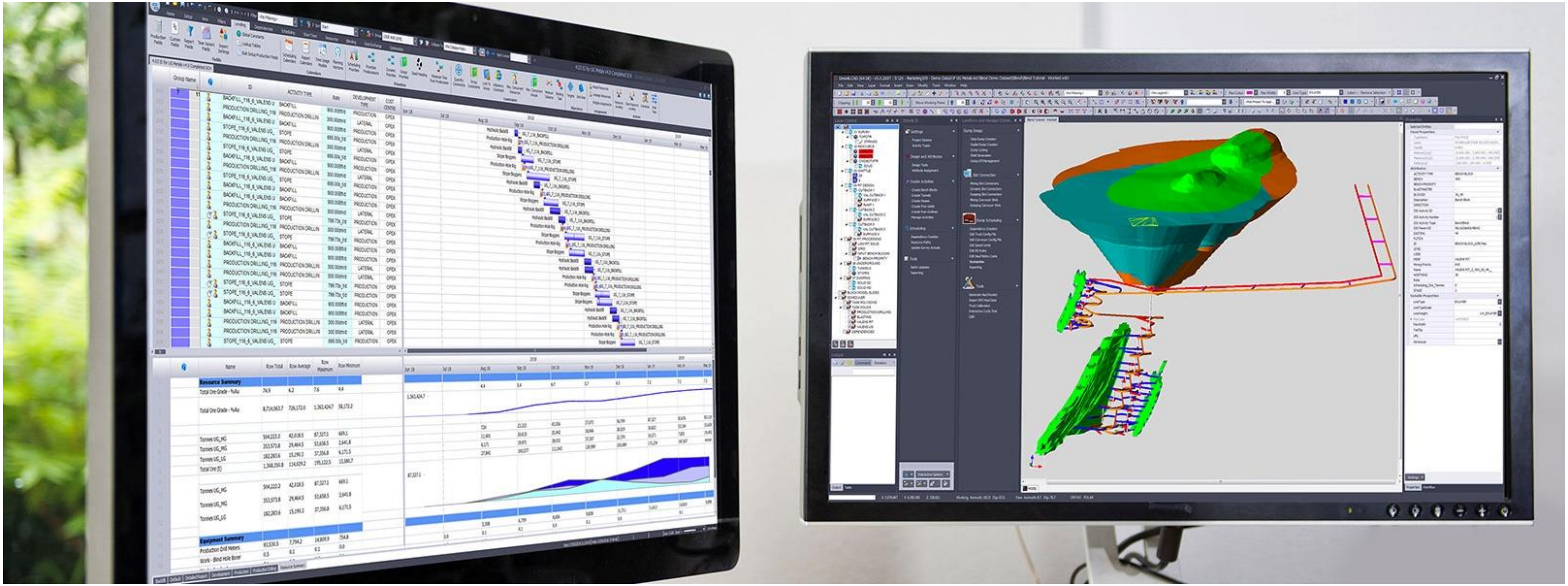
https://www.businessnews.com.au/article/K2fly-rolls-out-new-mining-software-solutions

https://horizongold.com.au/gumcreekproject/kingfisher/



https://announcements.asx.com.au/asxpdf/20200928/pdf/44n1rdhl9j90sn.pdf





Deswik Interactive Scheduler, a professional project managing software for underground mine planning

Imagen de deswik.com

The RCPSP with Discounted Cash Flows (RCPSP+DC)

RCPSP **+DC**

Given:

- Discrete time horizon $T = \{0, \ldots, h\}$
- Renewable resources R
  - Per-period availability q_r ($\forall r \in R$)
- Jobs $J = \{1, \ldots, n\}$
  - Duration d_j ($\forall j \in J$)
  - Resource utilization $u_{j,r}$ ($\forall j \in J, r \in R$)
- Precedence relation $A \subseteq J \times J$

+

- **Job profit** p_j

Find a schedule S for jobs in J in T such that

- j does not start before i ends $\forall (i, j) \in A$, and
- capacity utilization in period $[t-1, t]$ does not exceed q_r $\forall r \in R, t \in T \setminus \{0\}$

minimizing the project makespan (=duration).

+

- **maximize the net present value:**

  maximize the sum of discounted profits!

RCPSP-DC Variant for Underground Mine Planning

- **Optional jobs:** If a job is scheduled, then all its predecessors need to be scheduled!
- Positive precedence **lags**.
- Large number of jobs (>10K), precedences (>100K) and time periods (>3K)!

Problem: Extremely difficult to optimize!

1. Linear relaxation (LP) cannot be solved efficiently.
2. Constraint Programming (CP) does not
   - load model data efficiently.
   - find good schedules.
3. “No heuristics available”.

Maximising the Net Present Value of Large Resource-Constrained Projects

Hanyu Gu[1], Peter J. Stuckey[2], and Mark G. Wallace[3]

[1] National ICT Australia, Victoria Research Laboratory,
The University of Melbourne, Victoria 3010, Australia
Hanyu.Gu@nicta.com.au
[2] National ICT Australia, Department of Computing and Information Systems,
The University of Melbourne, Victoria 3010, Australia
pstuckey@unimelb.edu.au
[3] Monash University, Victoria 3100, Australia
mark.wallace@monash.edu

In 8-16 hours: 20 - 40% optimality gaps.

Turquoise Ridge Mine, Nevada, USA

14,000 activities

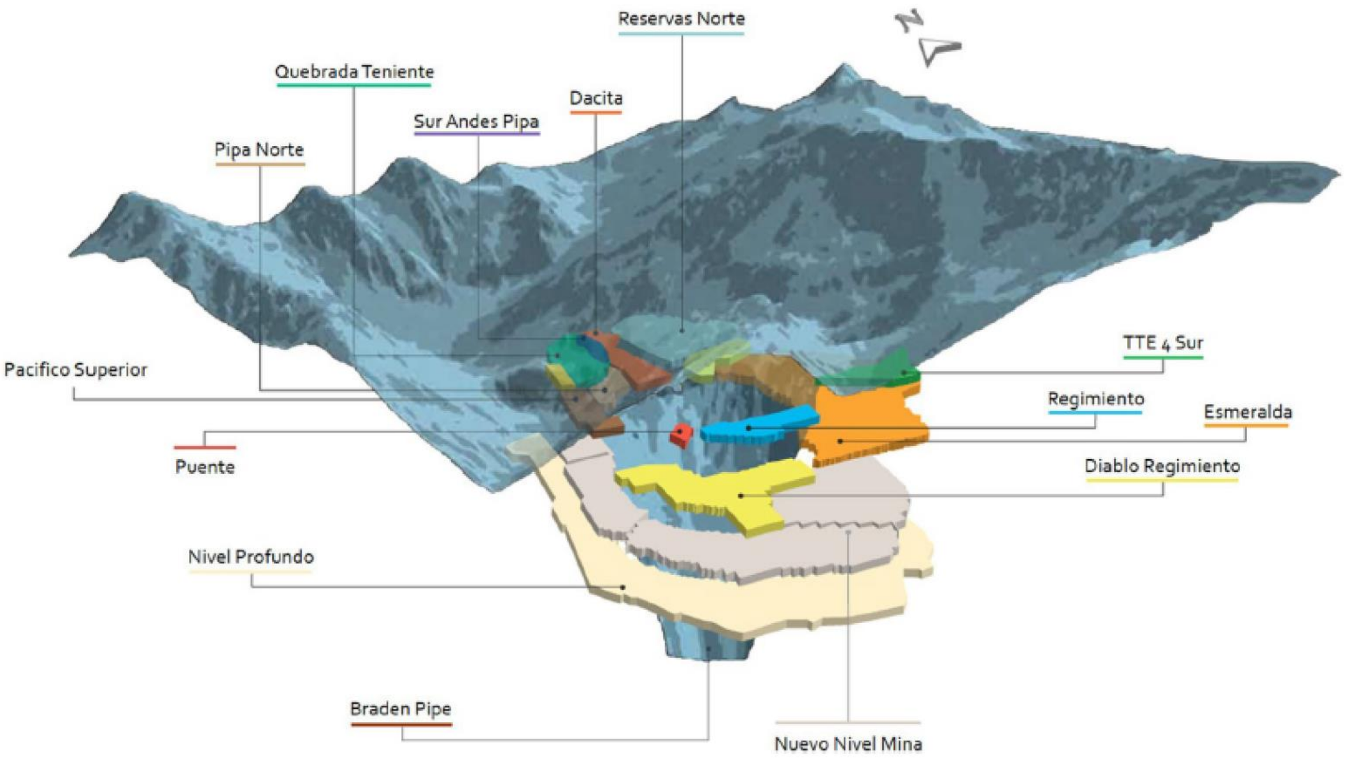
3,600 time periods

5 types of resources

maximize NPV

Image from Brickey et al. 2019

El Teniente, Chile

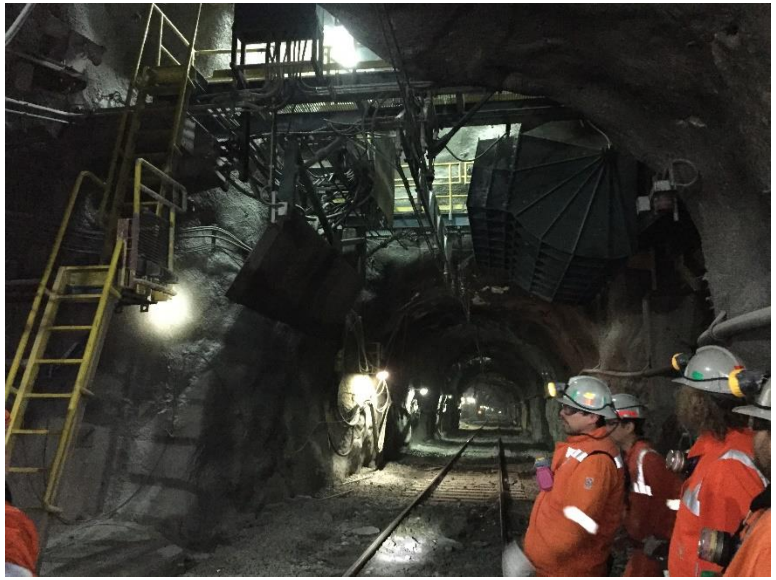


Over 3,000 km of underground tunnels — about the distance from Las Vegas to New York — and close to 1,500km of underground roads.

https://www.youtube.com/watch?v=kXvjrEonXEU

https://www.mining.com/worlds-largest-underground-mine-looks-like-38052/

El Teniente, Chile



RCPSP+DCs and Related Work

- Vast literature considering makespan minimization
  - Schwindt and Zimmermann (2015) “Handbook of Project Management and Scheduling”
  - Hartmann and Briskorn (2010) “A survey of variants and extensions of the resource-constrained project scheduling problem”
    European Journal of Operational Research

- Extremely hard optimization problems (PSPLIB*: <60 job instances unsolved)
  - Kolisch and Sprecher (1996) “PSPLIB - A project scheduling problem library”
    European Journal of Operational Research

- Methods for the NPV case: Heuristics and Constraint Programming
  - Vanhoucke (2001) “On maximizing the net present value of a project under renewable resource constraints”
    Management Science
  - Vanhoucke et al. (2010) “A scatter search heuristic for maximising the net present value of a resource-constrained project with fixed activity cash flows”
    International Journal of Production Research
  - Schutt et Al. (2012) “Maximising the net present value for resource-constrained project scheduling”
    CPAIOR 2012

Strategic Mine Planning using RCPSPs

RCPSP		Mine Planning
Jobs	⟷	Activities (Extraction, hauling, backfilling, etc.)
Resources	⟷	Mill capacity per period, machine/personnel availability, etc.
Precedence restrictions	⟷	Development of mine structure over time

Special RCPSP+DC Properties

- **Tasks can be unprofitable** (e.g., infrastructure development)
- **Not all tasks have to be scheduled**
- **Generalized precedence constraints** (time lags $l_{i,j}$)
- **All task predecessors need to be scheduled**

State-Of-The-Art Underground Mine Planning

Not much research has been done (as for open pit)
No powerful software systems available (as for open pit)

Algorithm* (Brickey, 2015):

1. Build mathematical formulation.
2. Solve the LP-relaxed problem with the Bienstock-Zuckerberg algorithm**.
3. Derive a feasible schedule via a priority-based heuristic.

Efficient implementation: OMP***

*Brickey (2015) "Underground production scheduling optimization with ventilation constraints"

**Bienstock, Zuckerberg (2010) "Solving LP Relaxations of Large-Scale Precedence Constrained Problems"

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

Hybrid Algorithm

I. Apply Problem Reduction Techniques
Effectively reduce real-world instance size

II. Solve Linear Relaxation
Bienstock-Zuckerberg Algorithm

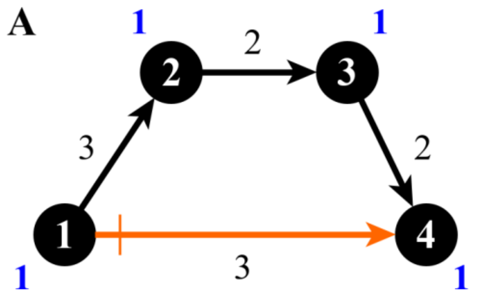
III. Derive Schedule(s) using List Scheduling
Fast LP-rounding-based heuristics

IV. Feed LP bound and Schedule into Constraint Programming
Optimize using IBM ILOG CP Optimizer

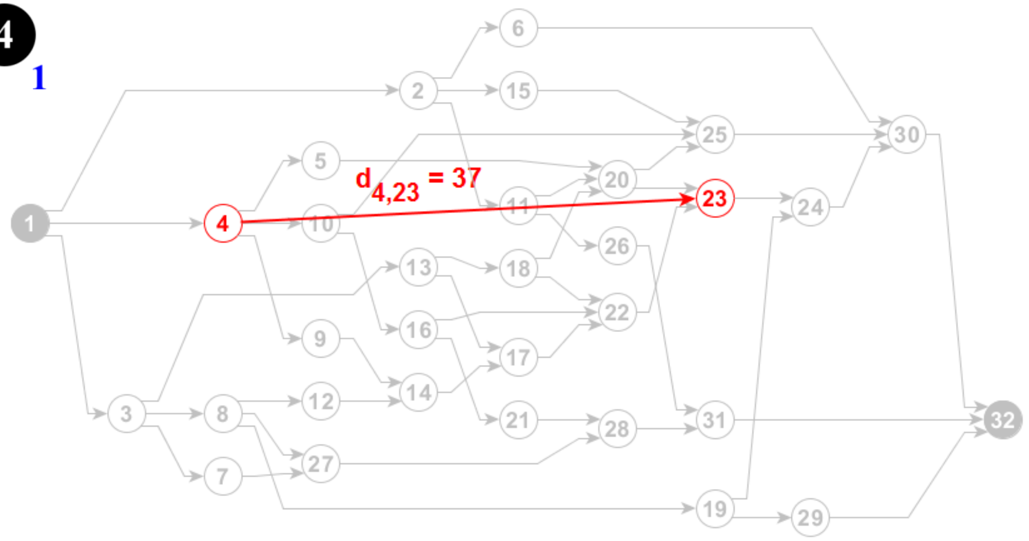
Reduction Techniques

Variable (and Constraint) Reduction

- **Weighted Transitive Reduction**

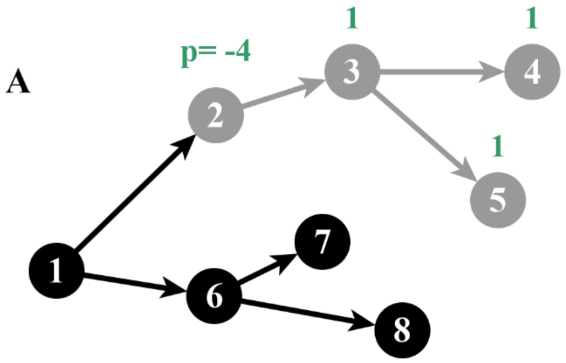


- **Earliest Activity Start Times:** Critical paths for all jobs



- **Sub-Makespan Precedences:**
  Find lower bounds on sub-project durations and add precedence lags (CP)

- **Unprofitable Branches:**
  Eliminate unprofitable successor trees



- **Trivial Job Elimination**
  (E.g., duration zero, zero resource utilization)

Time Period Aggregation

Solving the Linear Relaxation

Weak time-indexed IP formulation

$$x_{a,t} \in \{0,1\}$$

start activity “a” in time “t”

$$\max \sum_{t=1}^{T} \sum_{a \in A} e^{-\delta t} p_a x_{a,t}$$

Maximize project NPV

$$\text{s.t.:} \quad x_{b,t} \leq \sum_{s=1}^{t-l(a,b)} x_{a,s}$$

“b” is scheduled → “a” must start <u>at least</u> “l(a,b)” periods before “b”

$$\sum_{a \in A} q_a \sum_{s=t-d_a+1}^{t} x_{a,s} \leq R_t$$

limited resource availability

→ **Upper Bound**

(for optimal mine profit)

Bienstock-Zuckerberg Algorithm
(Lagrangian decomposition method)

Comput Optim Appl (2018) 69:501–534
https://doi.org/10.1007/s10589-017-9946-1

CrossMark

A study of the Bienstock–Zuckerberg algorithm: applications in mining and resource constrained project scheduling

Gonzalo Muñoz[1] · Daniel Espinoza[2] · Marcos Goycoolea[3] · Eduardo Moreno[4] · Maurice Queyranne[5] · Orlando Rivera Letelier[6]

Received: 28 June 2016 / Published online: 3 October 2017

Abstract We study a Lagrangian decomposition algorithm recently proposed by Dan Bienstock and Mark Zuckerberg for solving the LP relaxation of a class of open pit mine project scheduling problems. In this study we show that the Bienstock–Zuckerberg (BZ) algorithm can be used to solve LP relaxations corresponding to a much broader class of scheduling problems, including the well-known Resource Constrained Project Scheduling Problem (RCPSP), and multi-modal variants of the RCPSP that consider batch processing of jobs. We present a new, intuitive proof of correctness for the BZ algorithm that works by casting the BZ algorithm as a column generation algorithm.

Artigues (2017) “On the strength of time-indexed formulations for the resource-constrained project scheduling problem”

Bienstock et al. (2010) “Solving LP Relaxations of Large-Scale Precedence Constrained Problems”

List Scheduling Heuristics

1. Assign priorities to jobs → Job sequence (list)
2. Schedule jobs in sequence respecting constraints

LP-based priority metrics:

- **Expected Job Start Time**

$$\sum_{t=1}^{T} t x_{a,t}^{*} + (T+1)\left(1 - \sum_{t=1}^{T} x_{a,t}^{*}\right)$$

- **Alpha Points**

$$min\left\{t \in 1, \dots, T : \sum_{s=1}^{t} x_{a,s}^{*} \geq \alpha\right\}$$

- **Beta Points**

$$argmin\left\{\sum_{s=1}^{T} x_{a,s}^{*} |t-s|^{\beta} : t \in 1, \dots, T\right\}$$

→ Feasible Schedule (Lower Bound)

Note: *Use suboptimal LP solutions rather than just the final LP solution..*

Constraint Programming Formulations

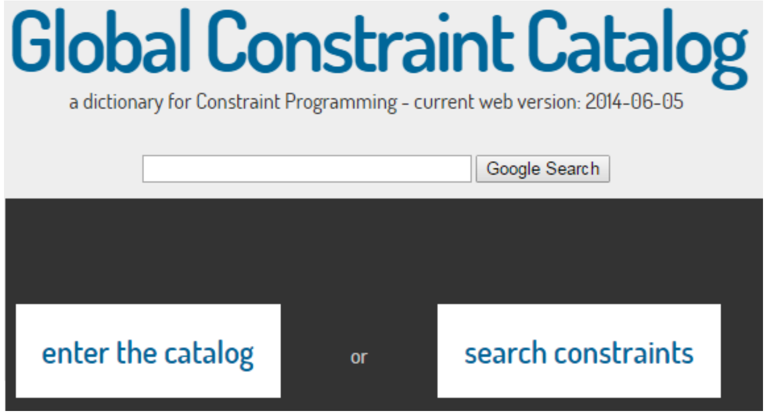
What is Constraint Programming (CP)?

1. **Decision variables**
   - Represented by <u>domains</u>
   - Continuous, discrete, Boolean

2. **Constraints**
   - Customized (e.g., $y_1^{y_2+3}$)
   - Pre-defined (e.g., $\mathtt{ALL\_DIFFERENT}(y_1, y_2, y_3)$)
   - Propagators identify implications on domains (re-applied after domain change)

3. **Objective function(s)**

Non-linear!



Special cases: SAT Solving, CP over finite domains

→ **Similarities with Mathematical Programming!**

Constraint Programming Formulations

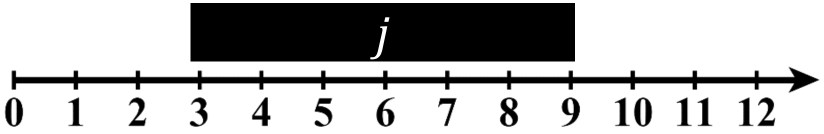
Constraint Programming solvers?

- **State-of-the-art solvers:**
  - Commercial (IBM ILOG CP Optimizer, etc.)
  - Free (GECODE, CHUFFED, etc.)

- **Key ingredients:**
  - Branch & bound
  - Automated parameter tuning
  - Finite domains are internally represented by intervals (e.g., x=[1,5]={1,2,3,4,5})
    (Scheduling models are not time-sensitive!)
  - Impact-based branching rules ("No-good learning")
  - Linear Programming
  - Neighborhood search
  - Machine learning
  - Conflict analysis
  - Probing

Constraint Programming Formulations

Job <u>interval variables</u> y_j $(j \in J)$



$\texttt{start}(y_j) \in T$

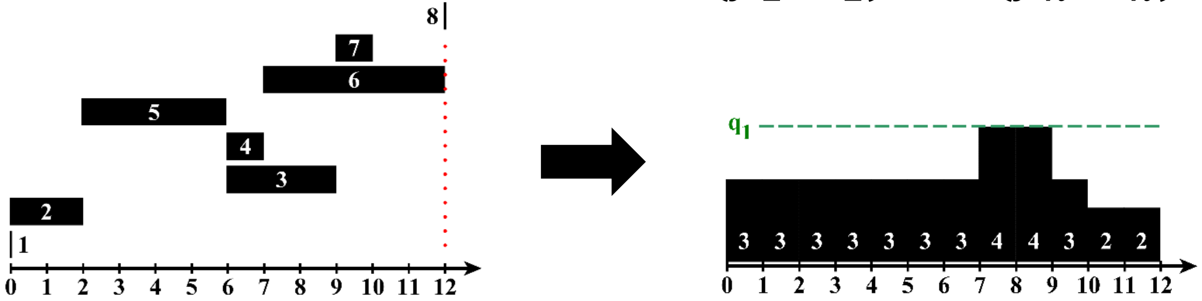
$\texttt{end}(y_j) \in T \qquad \geq \texttt{start}(y_j)$

$\texttt{length}(y_j) \in T \qquad = \texttt{end}(y_j) - \texttt{start}(y_j)$

$\texttt{presence\_of}(y_j) \in \{true, false\}$

$\texttt{cumulative\_function}((y_1, u_1), \ldots, (y_k, u_k))$

*Dynamically maintains a **resource profile** for a (partial) schedule with jobs 1,…,k.*



$\texttt{end\_before\_start}(y_i, y_j)$

*Constraint that ensures that j **does not start before** i **ends** (includes efficient propagator).*

Formulation - Constraint Programming

Max $\left(\sum_{j \in J} \mathrm{e}^{-r * \mathrm{start}(y_j)} p_j\right)$ ***NPV Objective***

Subject to: $\texttt{end\_before\_start}(y_i, y_j, l_{i,j})$ $((i,j) \in A)$ ***Precedences***

$\texttt{cumulative\_function}\ ((y_1, u_{1,r}), \ldots, (y_n, u_{n,r})) \leq\ q_r\ (r \in R)$ ***Resource Availability***

$\texttt{optional interval variable}\ y_j$ on T $(j \in J)$ ***Job Variables***

$\texttt{length}(y_j)\ =\ d_j$ $(j \in J)$ ***Job Duration***

$\texttt{presence\_of}(y_j)\ \Rightarrow\ \texttt{presence\_of}(y_i)$ $((i,j) \in A)$ ***Force Predecessor***

→ **Feasible Schedule and weak Upper Bound**

Note: *Based on well-known efficient RCPSP formulation (IBM ILOG CP Optimizer).*

Test Setup - Computations

- **16 academic* and real-world underground mining instances**
  1000 - 10000 tasks
  2000 - 110000 precedences
  60 - 3600 time periods
  2-8 resources

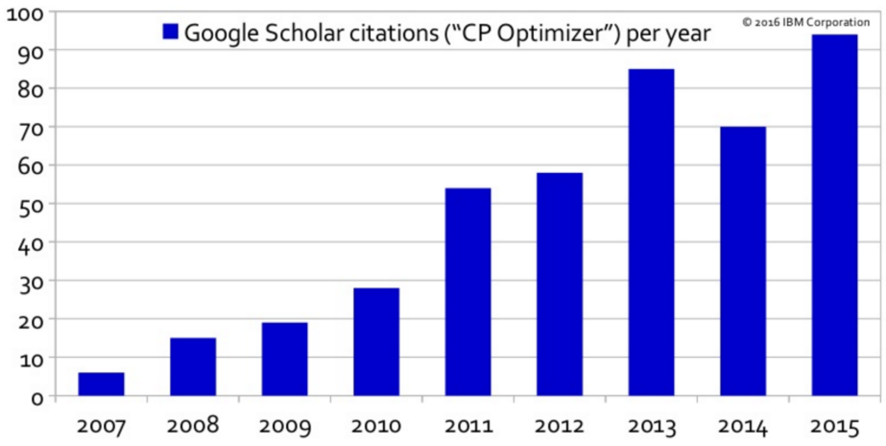
- **ILOG CP Optimizer 12.80**

- **Intel i5-2320, 3GHz, 4 cores, 8 GB RAM**

* Espinoza et al. (2013) "MineLib: a library of open pit mining problems"

ILOG CP Optimizer

- **State-Of-The-Art CP Solver**
- **Commercial software** (academic research license)
- **Rich scheduling-oriented interfaces** (C/C++, Java, Python, OPL)
- **Automated tuning**



CP Optimizer model for RCPSP:

```
dvar interval a[i in Tasks] size i.pt;

cumulFunction usage[r in Resources] =
  sum (i in Tasks: i.qty[r]>0) pulse(a[i], i.qty[r]);

minimize max(i in Tasks) endOf(a[i]);
subject to { forall (r in Resources)
               usage[r] <= Capacity[r];
             forall (i in Tasks, j in i.succs)
               endBeforeStart(a[i], a[<j>]);}
```

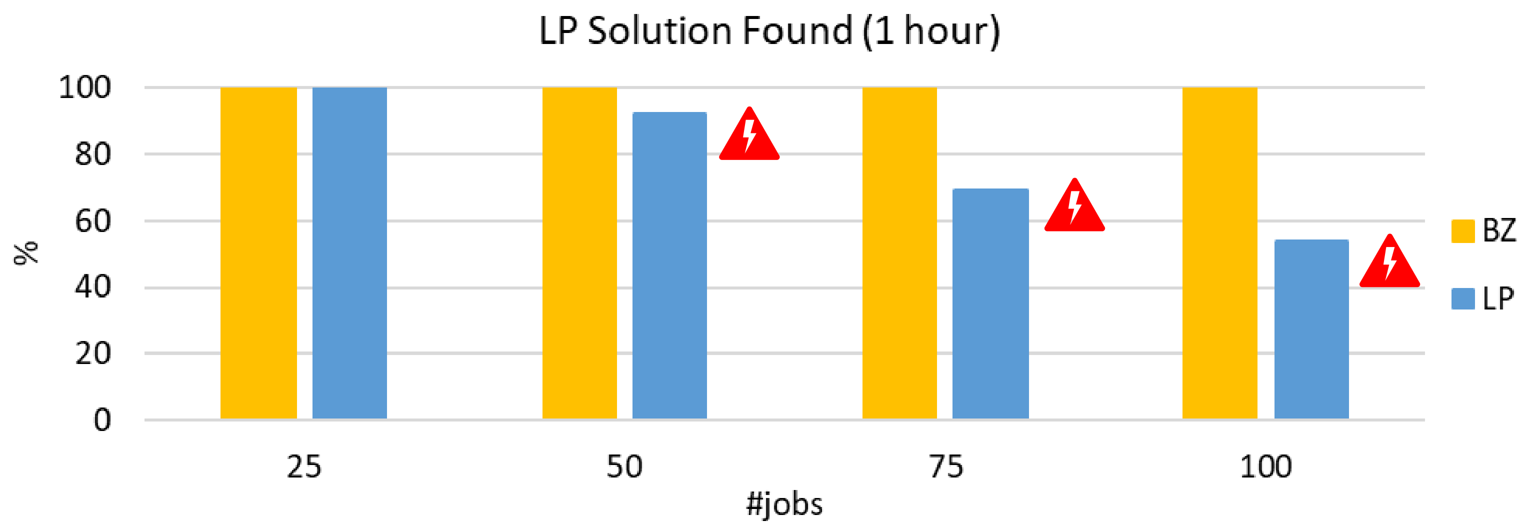
CP Performance (RCPSP-DC Literature Instances*)

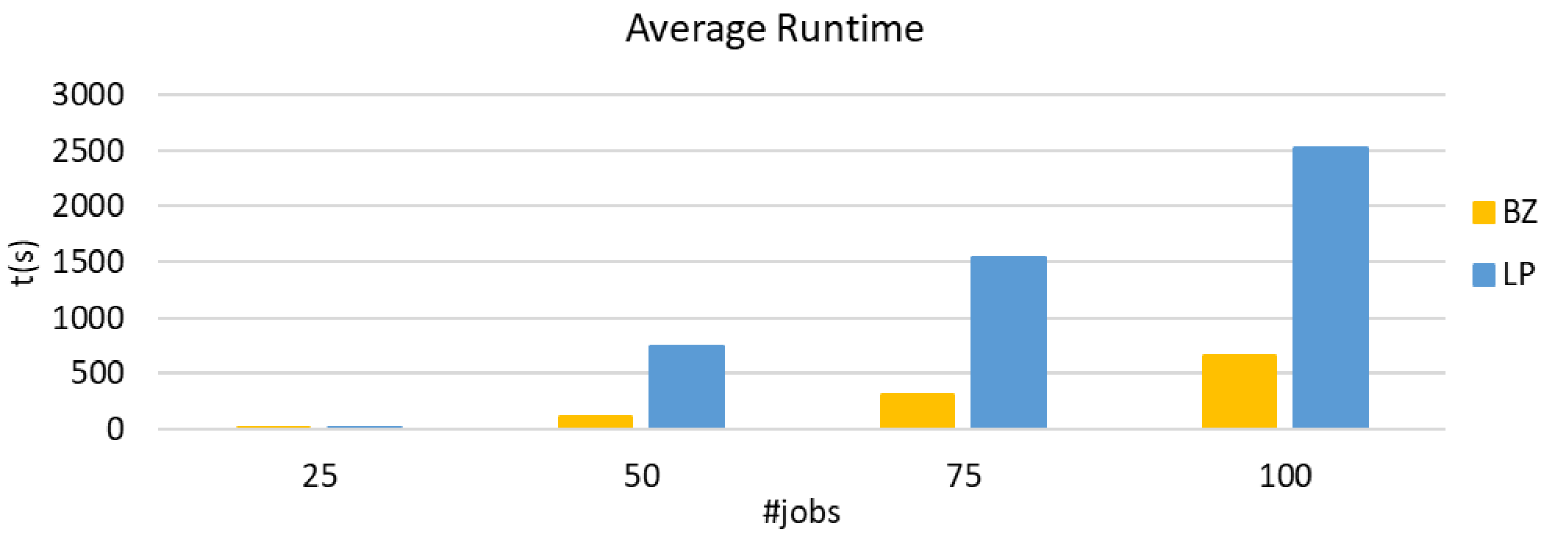
Literature

*Vanhoucke (2001) "On maximizing the net present value of a project under renewable resource constraints", *Management Science*

BZ vs. CPLEX Performance (RCPSP-DC Literature Instances*)

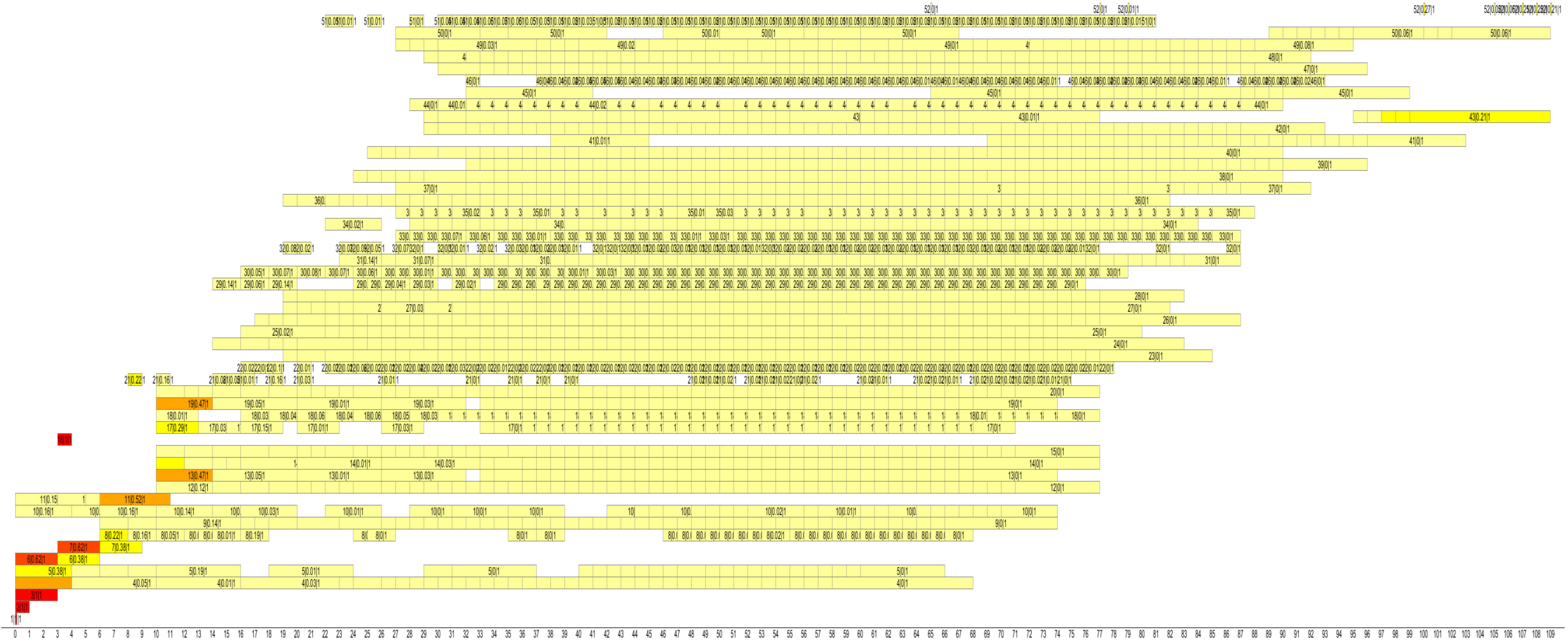
Literature



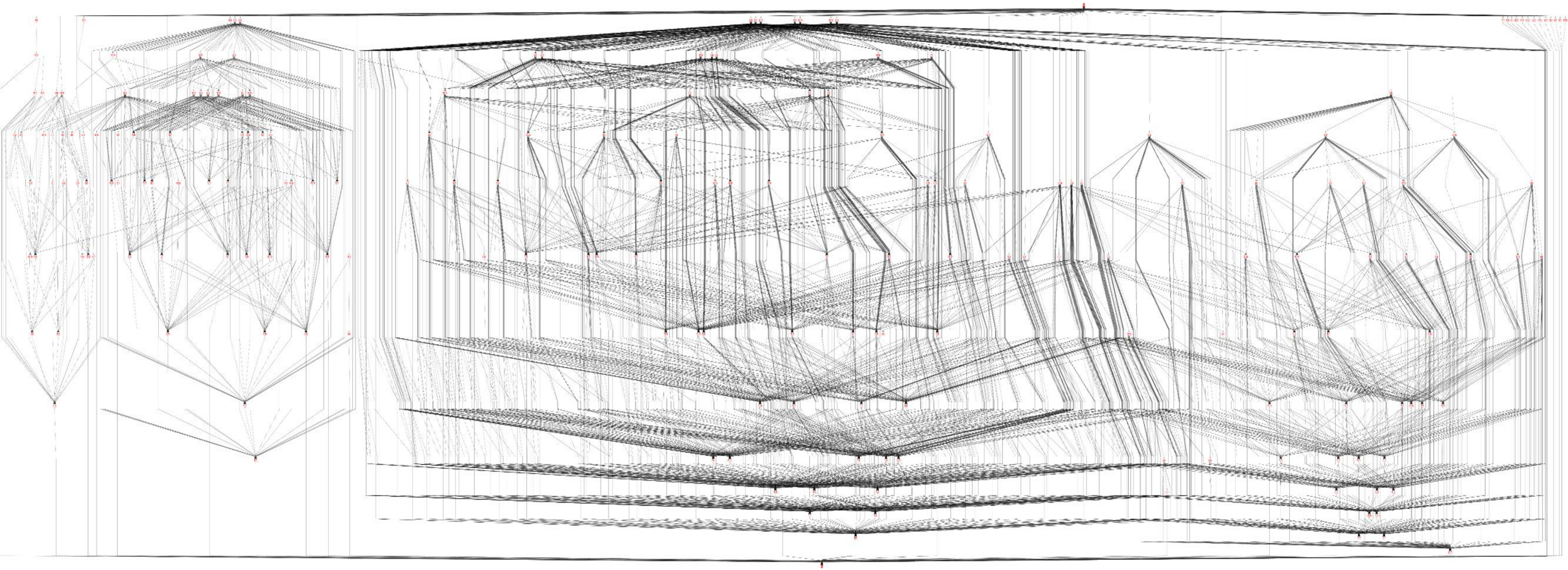


*Vanhoucke (2001) "On maximizing the net present value of a project under renewable resource constraints", *Management Science*

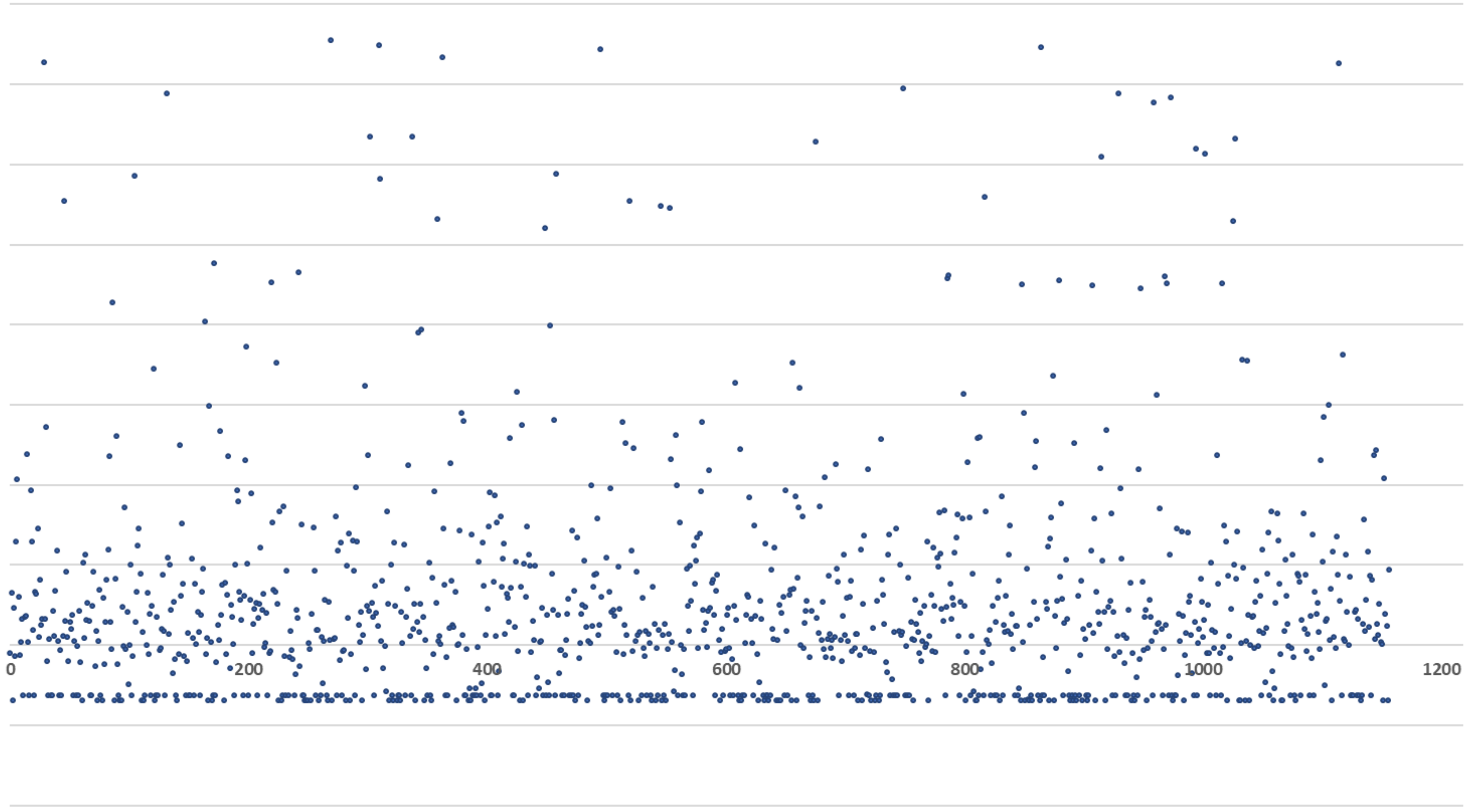
Fractional Solution (50 jobs)



Precedences (Real Mine)



Cash Flows (Real Mine)



Results: Reduction Techniques

Instance	Number of Periods	Before All Preprocessing Variables	Before All Preprocessing Activities	Before All Preprocessing Precedences	After All Preprocessing Variables	After All Preprocessing Activities	After All Preprocessing Precedences
A1	60	87,377	1,598	38,016	71,192	1,296	4,295
A2	60	101,684	1,881	65,621	79,093	1,451	5,381
A3	60	104,580	1,944	67,898	80,003	1,473	5,575
A4	60	107,840	1,961	94,508	85,504	1,539	6,321
A5	60	116,302	2,140	71,583	89,541	1,625	7,450
A6	60	132,283	2,453	111,065	99,581	1,817	8,945
B1	30	24,479	1,157	104,754	24,287	1,157	3,629
B2	30	24,774	1,160	103,624	24,622	1,160	3,694
B3	30	25,339	1,176	110,801	25,293	1,176	3,834
Agricola	7,200	65,193,806	14,160	16,507	56,790,620	13,343	13,606
Catan	1,800	6,486,956	8,497	14,632	5,356,876	7,553	10,052
Dominion	3,600	65,405,074	28,883	49,313	55,382,828	28,393	29,788

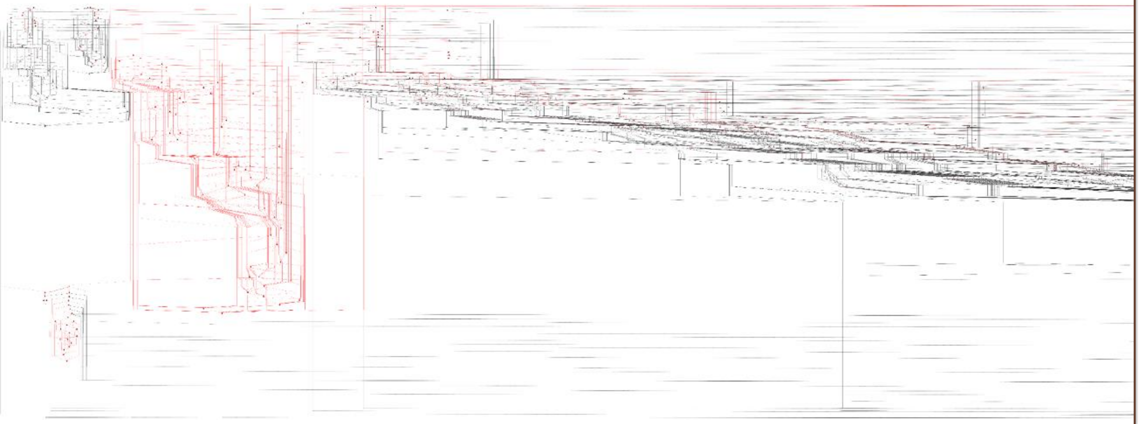
Computational Results - Hybrid Optimization Algorithms

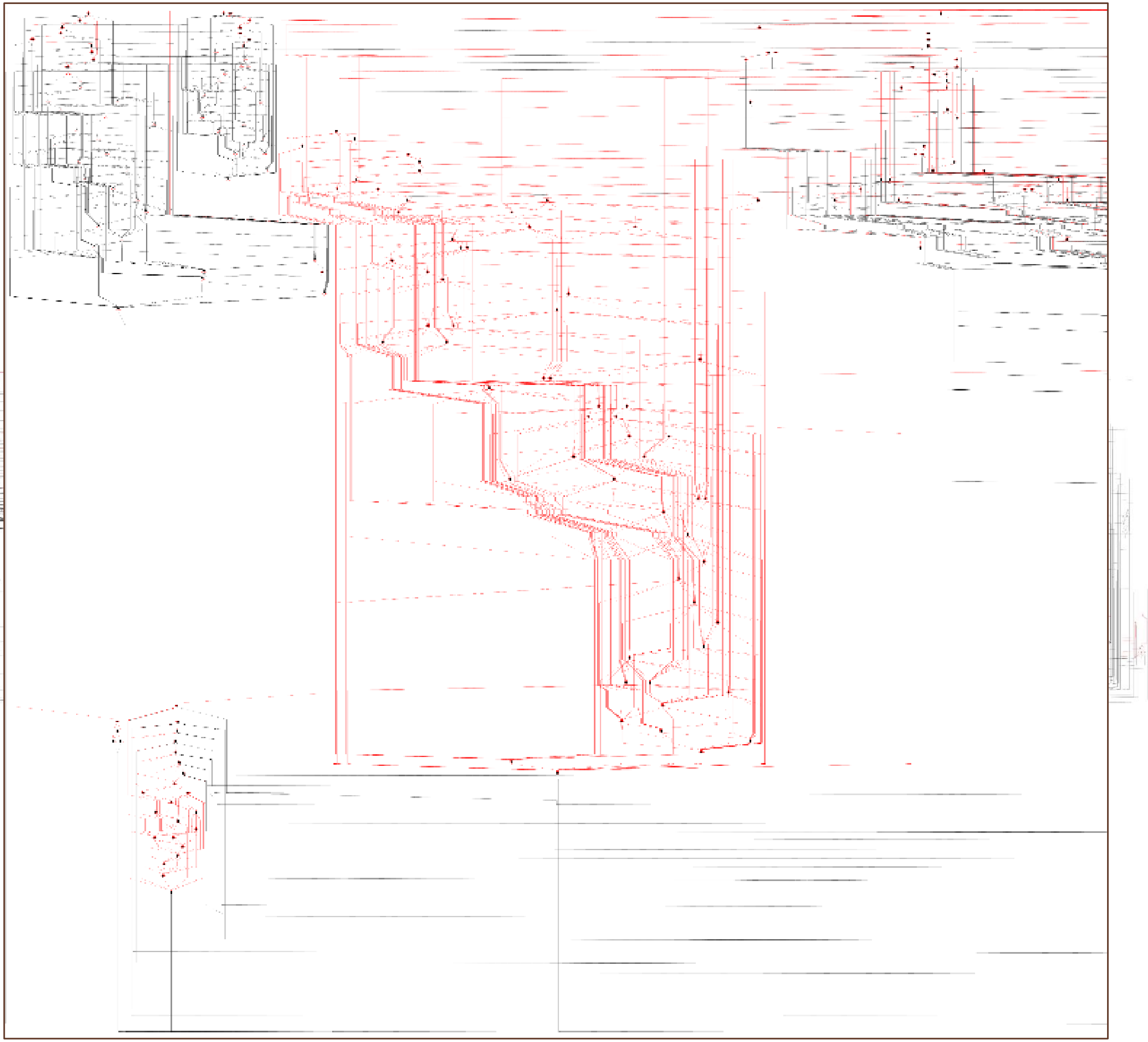
Constraint Programming (CP) — **CP with Start Schedule** — **CP with Start Schedule and Mathematical Programming Upper Bound**

Instance	TopoGap (%)	CP Alone: Gap at 600 sec. (%)	CP Alone: Gap at 21,600 sec. (%)	CP Alone: Gap at 86,400 (%)	CP with UB: Gap at 600 sec. (%)	CP with UB: Gap at 21,600 sec. (%)	CP with UB: Gap at 86,400 (%)	CP with UB and LB: Gap at 600 sec. (%)	CP with UB and LB: Gap at 21,600 sec. (%)	CP with UB and LB: Gap at 86,400 (%)
A-1	1.00	21.65	20.99	20.99	1.10	1.00	1.00	1.00	1.00	1.00
A-2	1.34	23.67	23.07	22.99	1.72	1.00	1.00	1.13	1.00	1.00
A-3	1.6	26.42	22.85	22.83	1.91	1.00	1.00	1.34	1.00	1.00
A-4	2.80	32.51	29.92	29.79	3.73	2.46	2.28	2.01	1.55	1.54
A-5	2.53	40.46	38.20	38.00	3.46	2.19	1.88	2.03	1.67	1.59
A-6	2.60	42.00	41.03	41.00	3.00	1.40	1.34	2.10	1.40	1.40
B-1	4.05	72.26	72.11	72.06	15.44	15.00	14.82	2.44	2.10	2.02
B-2	2.89	71.03	70.80	70.80	14.66	13.89	13.85	2.00	2.00	2.00
B-3	2.40	67.60	67.41	67.41	7.78	7.60	7.60	2.40	2.40	2.40
C-1	13.30	16.35	4.84	4.10	12.00	9.53	8.82	1.00	1.00	1.00
C-2	11.64	25.89	23.77	23.29	21.37	19.12	18.61	7.34	5.92	5.89
C-3	13.57	23.25	21.24	21.10	13.72	11.47	11.31	7.62	5.36	5.06
C-4	11.50	57.78	53.50	52.93	27.87	20.58	19.58	9.56	6.85	6.12
C-5	14.69	58.28	55.38	54.83	27.14	22.08	21.12	13.14	10.80	9.94
D	8.70	16.15	13.51	13.23	8.94	6.07	5.78	7.60	5.88	5.88
E	20.60	36.96	35.73	35.28	13.28	11.65	11.02	8.44	7.88	7.69
Linear Average (%)	7.20	39.52	37.15	36.91	11.07	9.13	8.81	4.45	3.61	3.47

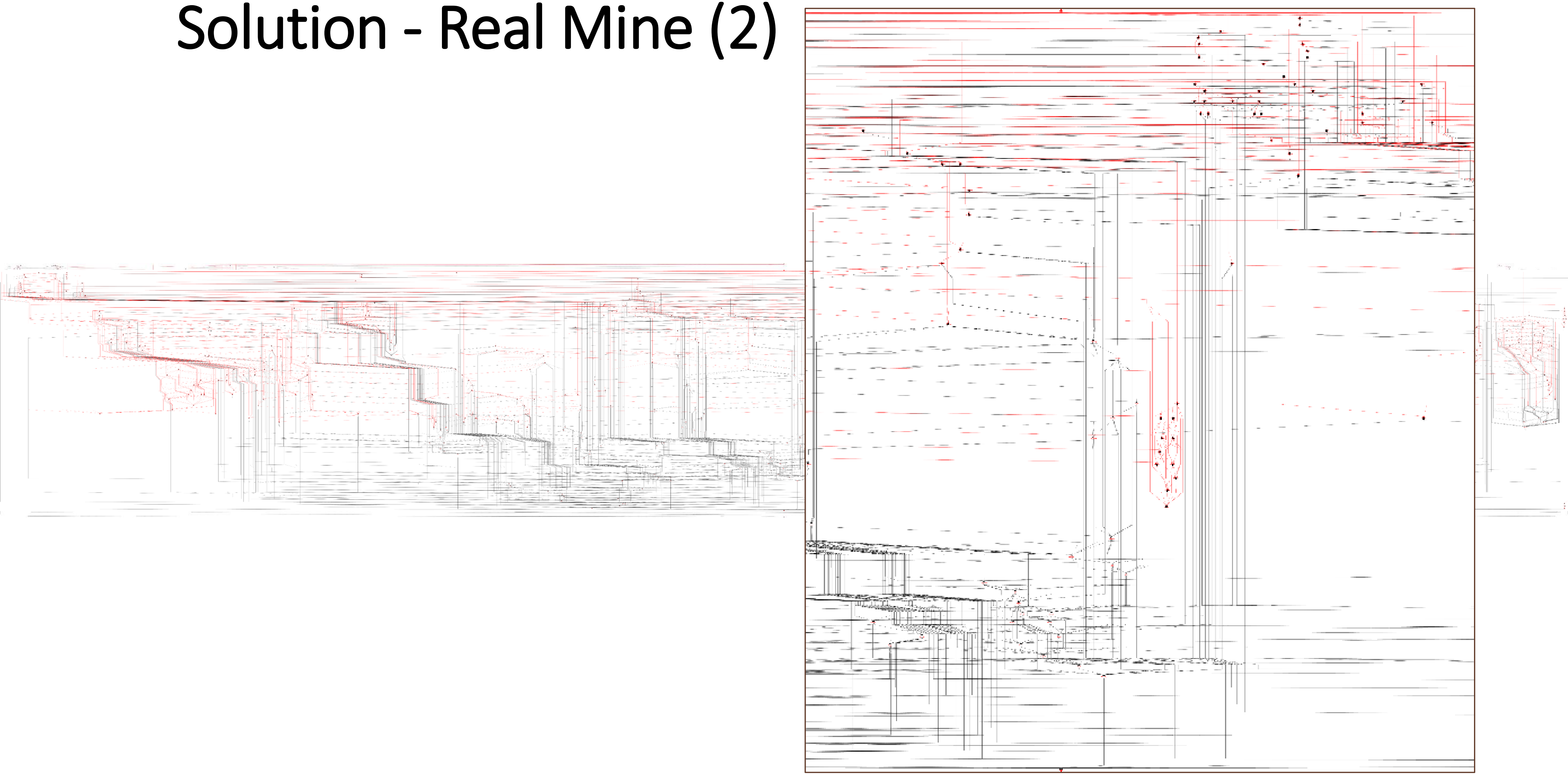
Optimality Gaps

Solution - Real Mine (1)





Solution - Real Mine (2)



Conclusion

- **Underground planning problems can be represented as RCPSPs!**
- **Constraint Programming can solve real-world instances, but only after preprocessing!**
- **Constraint Programming works best when hybridized!**

Thank you!

Alessandro Hill

ahill29@calpoly.edu

Hill, Lalla-Ruiz, Goycoolea and Voβ (2018)
“A Multi-Mode Resource-Constrained Project Scheduling Reformulation for the Waterway Ship Scheduling Problem”
Journal of Scheduling

Hill, Brickey, Newman, Ciprano and Goycoolea (2022)
“Optimization Strategies for Resource-Constrained Project Scheduling Problems in Underground Mining”
INFORMS Journal on Computing