Aggregation Heuristics for the Open Pit Scheduling Problem

Enrique Jelvez (UTFSM, Valparaíso)
Nelson Morales (UCHILE, Santiago)
Juan Peypouquet (UTFSM, Valparaíso)
Patricio Reyes (Carlos III Madrid)

X ISORA, Lima, Perú, October 2011
OVIMINE Session
Motivation

• Mining is a very relevant industry in Chile (and Peru, and...): 45% of exportations, 18% PIB.
• Optimization Problems related to open pit mine planning are old (60’s) but solving them remains difficult:
  – Commercial software does not solve the real problem. (Because the real problem is very difficult – algorithmically, conceptually, and because of its size.)
Open Pit Mining

- Ore is relatively close to the surface, so removing waste material to reach mineralized zones pays off.
- Cheaper to operate, larger production than underground mines.
- Mine is discretized into “small” blocks (10x10x10 m³, for example). Full database is called the block model.
Mine Planning

• Discipline that transforms the mine information and economic parameters into a production plan (how much to produce and when) and therefore a business plan and economical value.

• The production plan is supported by a scheduling, which determines what blocks are going to be extracted, whether they will be processed or not and when all this should happen.
Example: Final Pit

- Given:
  - A set of blocks \( B = \{1,2, \ldots, N\} \)
  - Real values (associated to economic value)
    \[ v(1), v(2), \ldots, v(N) \]
  - Precedence A relation so
    \( (i, j) \in A \iff j \text{ must be extracted before } i \)
- Find a pit (set of blocks compatible with the precedence) of maximal contained value.
  \[ \sum_{i \in \mathcal{E}} v(i) \]
Example: Final Pit

\[(i, j) \in A \iff j \text{ must be extracted before } i\]
Good things about the final pit

- If value $v(i)$ is replaced with $v'(i)$ with $v'(i)$ smaller than $v(i)$, then the optimal solution in the new setting is smaller (in the sense of inclusion). This allows to create nested pits.
- It is “easy”: can be solved quickly (Lerchs & Grossman 65, Hochbaum 2001-2009), even for hundreds of thousands of blocks.
- It is the algorithm used in commercial software for mine planning.
Bad things about the final pit

- It does **not** consider production and mining capacities, hence, it does **not** take time into account.
- It requires to make the decision about the **destination** of the block **beforehand**.
- It is **the** algorithm used in commercial software for mine planning.

This talk is about solving this issue, but the techniques can be easily extended to the second one.
Open Pit block scheduling problem
(still simple version)

• Given:
  – Set of blocks with values and precedences (so graph $G=(B,A)$), as before.
  – A number $T$ of time-periods $t=1,2,...,T$.
  – Resources $r=1,2,...,R$:
    • Capacities per resource and time-period
      \[ w_r(1), w_r(2), ..., w_r(T) \]
    • Resource consumption per block
      \[ w_r(1), w_r(2), ..., w_r(N) \]
OPBSP

\[
x_{it} \in \{0,1\} \quad \text{1 iif block } i \text{ has been extracted between } l, 2, \ldots, T
\]

\[
\max \sum_i \sum_t \rho^t v(i)(x_{it} - x_{it-1})
\]

\[
x_{it-1} \leq x_{it} \quad (\forall i = 1, \ldots, N)(\forall t = 2, \ldots, T)
\]

\[
x_{it} \leq x_{jt} \quad (\forall(i, j) \in A)(\forall t = 1, \ldots, T)
\]

\[
\sum_i w_r(i)(x_{it} - x_{it-1}) \leq W_r(t) \quad (\forall t = 1, \ldots, T)(\forall r = 1, \ldots, R)
\]

Problem is very **big** (ex: 50,000 blocks 10 periods = half a million binary variables, and about 750,000 constraints), and this is only the deterministic case!
Main idea: Are we overkilling?

• Use aggregation to reduce the size of the problem and heuristics to generate feasible (good) solutions:
  – Reduce number of periods: Consider longer periods of time and refining within, or take fewer periods (a part), solve, fix and move on; and
  – Reblocking: Aggregate blocks into larger units to provide a guide for the actual problem.
Heuristics

- **H1** (incremental, greedy approach):
  - Solve the problem for the first time period,
  - remove mined blocks,
  - iterate for remaining periods.

- **H2** (reverse approach):
  - Solve the problem for one large monolithic time-period (adding 1,2,...,T),
  - remove NOT mined blocks
  - Iterate within the remaining blocks, but periods (1,2,...,T-1)

- **H3** (aggregate blocks):
  - Aggregate blocks into larger units
  - “Solve” the problem
  - Use mining period of larger units as guide for mining periods of actual blocks.
one-period incremental heuristic (H1)
Select pit satisfying capacity constraints...
Remove those blocks...
... and start over (update capacities)
... one last time...
Final solution
done
one-period reverse heuristic (H2)
Most profitable pit with overall capacity
... blocks outside that pit are left out out of the game.

Capacity equals the sum overall periods of time.
... next blocks are selected within previous solution...

Capacity equals sum up to period T-1

These blocks are out of the game.
...and we iterate...
... until an overall feasible solution is found.
Block aggregation (H3)
Blocks are grouped into larger units
Alternative problem is solved
"Important" decisions are made
Blocks at the *interior* are fixed
Blocks in the borders are then refined and re-optimized
Final solution is reported at original block level

Some notes:
• Careful with time-limits.
• Capacities must be adjusted.
• Many options when having many time-periods
H3 used recursively

Many Blocks

Problem is too big, REBLOCK

Still a lot...

Problem is too big, REBLOCK

Now maybe...

Insert solution

Solution FOUND

Insert solution

Solution FOUND

Solution FOUND
About the “solving” step

• How to try to solve the instances?
  – Integer Programming (call solver directly)
  – Use some of the heuristics (H1, H2, others)

• How do we know the problem “is too big”?
  – Time limit
  – Memory limit
  – Experience...
Case study : Marvin

- Imaginary mine (but well known).
- About 12,500 blocks (after some preprocessing).
- Blocks have:
  - Value (by fixed destination)
  - Tonnage (transportation usage)
  - Plant Tonnage (processing usage, zero for waste blocks).
- Two capacity constraints:
  - Mining (transportation) is 150,000 tpd
  - Processing (plant) is 60,000 tpd.
- Block predecessors:
  - Slope angle of 45 degrees (actually 5 blocks above in upper level).
- 8 time periods.
- Discount rate of 10% per year.
### Results

Each cell reports the value (imaginary-MUS$) and time (seconds) to reach the solution.

The time does not consider creating the instance (reading block model files, calculating precedences, reblocking for H3, etc.), which is about 2 minutes in the larger models.

OoM = Out of Memory

<table>
<thead>
<tr>
<th>Mine</th>
<th>10x10x10</th>
<th>15x15x15</th>
<th>30x30x30</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Blocks</td>
<td>336,636</td>
<td>99,744</td>
<td>12,468</td>
</tr>
<tr>
<td>IP</td>
<td>OoM</td>
<td>OoM</td>
<td>OoM</td>
</tr>
<tr>
<td>H1</td>
<td>OoM</td>
<td>OoM</td>
<td>1,491 690</td>
</tr>
<tr>
<td>H2</td>
<td>OoM</td>
<td>1,416 9,000</td>
<td>1,446 220</td>
</tr>
<tr>
<td>H3/IP</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>H3/H1</td>
<td>1,853 10,500</td>
<td>1,850 730</td>
<td>(=H1)</td>
</tr>
<tr>
<td>H3/H2</td>
<td>-</td>
<td>-</td>
<td>(=H2)</td>
</tr>
</tbody>
</table>

LP (upper bound) is available only for the larger blocks, and gives gaps of about 70%.
Conclusions

• Aggregation techniques allow to tackle instances that are “unsolvable” otherwise.
• Execution time is still long, but there is a lot of room for on.
• There are natural extensions that should improve solution quality. (Lagrangian relaxation of capacity constraints has shown very promising results: Chicoisne 2009, Bienstock 2010.)
• Still lot of work to do (in particular, obtain good upper bounds to measure solution quality).
Further work

• Calculate upper bound:
  – When available, LP relaxation usually delivers a tight bound.

• Improve heuristics:
  – Consider incremental and reverse heuristics with more than 1 time-period per iteration,
  – Consider incremental and reverse heuristics with many periods, but some of them with relaxed variables.

• Extend to the case where coefficients in capacity constraints may be negative (constraints on the average value of some attribute)

• Extend to the case where the model decides the destination of the block (many possible economic values per block --theoretically easy)

• Stochastic case (geology, prices, operations).

• ...