Modeling and Optimization Tools for Solving Production Planning and Scheduling Problems

Alkis Vazacopoulos

CAPD Meeting
Carnegie Mellon University
Pittsburgh PA

March 2011
Production Planning and Scheduling Problems

» Strategic
» Tactical
» Operational
Software Companies/Academic Instit/Consulting Companies

» SAP
» Oracle
» Aspentech
» Honeywell
» KBC
» Consulting companies

» IBM-CPLEX, Gurobi, FICO, GAMS, .....
Technologies

» LP
» MIP
» NLP
» CP
» Modeling
» Platforms
» Business Rules
LP Performance across releases (FICO Xpress)

Internal test set of 796 public and customer models
LP Performance across releases

Internal test set of 796 public and customer models
LP Performance across releases

Internal test set of 796 public and customer models
MIP Performance across releases

- **Number Solved**
- **Total Solution Time**

**Release**
- 2003B
- 2004B
- 2005B
- 2006B
- 2007B
- 2008A
- 7.0
- 7.1

**Internal test set of 320 public and customer models**
MIP Performance across releases

- Branch and Bound
- Cuts
- Classic problem specific
- Heuristics
  - General and problem specific
  - parallelize

Release:
- 2003B
- 2004B
- 2005B
- 2006B
- 2007B
- 2008B
- FICO 7.0

Number Solved
- 2003B: 50000
- 2004B: 170
- 2005B: 230
- 2006B: 230
- 2007B: 170
- 2008B: 170
- FICO 7.0: 170

Total Time(s)
- 2003B: 500
- 2004B: 190
- 2005B: 230
- 2006B: 230
- 2007B: 170
- 2008B: 170
- FICO 7.0: 170
Feedback Loop

- Reengineer B&B
- Decomposition Algorithms – more flexible
- Perform Operations - Node

MIP Performance

Number Solved vs. Total Time (s)

- 2003B
- 2004B
- 2005B
- 2006B
- 2007B
- 2008A
- FICO 7.0

Releases:
- 2003B
- 2004B
- 2005B
- 2006B
- 2007B
- 2008A
- FICO 7.0

Number Solved

- 50000
- 100000
- 250000

Total Time (s)

- 150
- 170
- 190
- 210
- 230
- 250
- 270
- 290
Comparison GUROBI vs CPLEX

» Gurobi: developed to take advantage of multi-core machines
» Growing very fast (150 commercial licenses, 25 academic sales, in addition to the academic program, source: jtoedm.com)
» LP 10-30% better than CPLEX (Mittleman website)
» MIP 10% - 30% better on optimality
» CPLEX better on Feasibility problems
» New MIPLIB data soon
Open Source

» SKIP – ZIB, Very competitive: used by Dynadec
» CBC – used as an entry by many companies

» Comparison from Hans Mittelman’s website
Benchmarking MIP

» Optimality (what about problems won’t solve after 30 minutes, 60 minutes)
» Best feasible solutions
» Find k best solutions
» Size of the tree
» Presolve (reduction of the matrix)

» Develop Sets of Problems
» Etc....
### Xpress 7 parallel speed ups

<table>
<thead>
<tr>
<th></th>
<th>Xpress 7 1 thread</th>
<th>Xpress 7 4 thread</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Internal” deterministic</td>
<td>25,724</td>
<td>17,447</td>
<td>-47%</td>
</tr>
<tr>
<td>“Internal” opportunistic</td>
<td>25,724</td>
<td>13,067</td>
<td>-96%</td>
</tr>
<tr>
<td>Coral deterministic</td>
<td>24,484</td>
<td>16,129</td>
<td>-52%</td>
</tr>
<tr>
<td>Coral opportunistic</td>
<td>24,484</td>
<td>11,137</td>
<td>-120%</td>
</tr>
</tbody>
</table>
mzzv42z: Single/Two Threads Xpress FICO IVE
## Xpress 7 parallel speed ups

<table>
<thead>
<tr>
<th></th>
<th>Xpress 7 1 thread</th>
<th>Xpress 7 4 thread</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Internal” deterministic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Internal” opportunistic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coral deterministic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coral opportunistic</td>
<td>24,484</td>
<td>1,137</td>
<td>-120%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Parallel processors on a Network**
- **Parallel MIP (same computer) P-MIP**
- **Deterministic P-MIP**
Concurrent LP solver

» Efficient LP method parallelization

<table>
<thead>
<tr>
<th></th>
<th>Primal</th>
<th>Dual</th>
<th>Barrier</th>
<th>Concurrent</th>
<th>Overall Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 thread</td>
<td>19,940</td>
<td>13,459</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2 threads</td>
<td>19,940</td>
<td>13,459</td>
<td></td>
<td>6,193 (B+D)</td>
<td>23%</td>
</tr>
<tr>
<td>4 threads</td>
<td>19,940</td>
<td>13,459</td>
<td>7,540</td>
<td>7,491</td>
<td>70%</td>
</tr>
</tbody>
</table>

Time(concurrent) ~ min(Time(dual), Time(primal), Time(barrier))

Controls: LPTHREADS and DETERMINISTIC
(concurrent is nondeterministic!)
**Concurrent LP solver**

» Efficient LP method parallelization

<table>
<thead>
<tr>
<th></th>
<th>Primal</th>
<th>Dual</th>
<th>Barrier</th>
<th>Concurrent</th>
<th>Overall Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 thread</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>2 threads</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>23%</td>
</tr>
<tr>
<td>4 threads</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>70%</td>
</tr>
</tbody>
</table>

Time(concurrent) ~ min(Time(dual), Time(primal), Time(barrier))

Controls: LP_THREADS and DETERMINISTIC
(concurrent is nondeterministic!)

P-Barrier

P-D-B Combinations
Customers want multiple alternative solutions
Example: air04

» Set partitioning problem for airline crew scheduling
» Available from MIPLIB 2003

<table>
<thead>
<tr>
<th></th>
<th>Default Solver</th>
<th>5-Best Solution Solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>1\textsuperscript{st} Solution</td>
<td>56137</td>
<td>56137</td>
</tr>
<tr>
<td>2\textsuperscript{nd} Solution</td>
<td>56166</td>
<td>56137</td>
</tr>
<tr>
<td>3\textsuperscript{rd} Solution</td>
<td>56307</td>
<td>56137</td>
</tr>
<tr>
<td>4\textsuperscript{th} Solution</td>
<td>56854</td>
<td>56137</td>
</tr>
<tr>
<td>5\textsuperscript{th} Solution</td>
<td>57562</td>
<td>56137</td>
</tr>
<tr>
<td>Total B&amp;B Nodes</td>
<td>141</td>
<td>15,981</td>
</tr>
<tr>
<td>Total Time</td>
<td>26 sec</td>
<td>29 sec</td>
</tr>
</tbody>
</table>
Customers want multiple alternative solutions
Example: air04

- Set partitioning problem for airline crew scheduling
- Available from MIPLIB 2003

<table>
<thead>
<tr>
<th>Strategies - Business Rules</th>
<th>Robust - Strategies</th>
<th>Strategy Refinement Rebalancing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Solution 56137</td>
<td>2nd Solution 56166 56137</td>
<td>3rd Solution 56307</td>
</tr>
<tr>
<td>4th Solution 56654</td>
<td>5th Solution 57562</td>
<td>Total B&amp;B Nodes 141</td>
</tr>
<tr>
<td>Total Time 26 sec</td>
<td></td>
<td>Total Time 29 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>56137</td>
</tr>
<tr>
<td></td>
<td>57562</td>
<td>15 981</td>
</tr>
</tbody>
</table>

Total B&B Nodes 141
Total Time 26 sec
Xpress-Tuner: How to Tune (Automatically) an Optimization Problem?
Xpress-Tuner:

Tuning Process

1. Your goal: Seeking to reduce time to gap, target gap 0%, within 5 s.
2. Progress update:
   - Time elapsed: 1m 36s
   - Time remaining: 23h 58m 24s
   - Improvement: YES
   - View Detailed Strategy Rankings...
3. Activity log:
   - air04.mps
     - Time: 7s Gap: 1.28024%
     - Time: 7s Gap: 3.26709%
     - Time: 7s Gap: 1.40793%
     - Time: 7s Gap: 1e+030%
4. Current strategy:
   - Thread 1
5. Phase 1: basic 18/109
   - Dual simplex for the LP relaxation.
   - GUROBI=
     - MAXTIME=5
     - MIPRELSTOP=0
   - Objective
   - Best bound
   - # solutions
   - # nodes

Available physical memory: 2559 MB
Xpress-Tuner:
Detailed Results

![Diagram showing strategy rankings with machine, version, when, alg, and file columns, followed by results, rank, and gap columns. The diagram emphasizes for the record, control parameters, and results sections.]
Xpress-Tuner: Comparing logs

20071119.170353.203
Strategy: Phase 1: basic 14/109

Objective
Best bound
# solutions
# nodes

Problem Statistics
884 (0 spare) rows
8904 (0 spare) structural columns
10859 (0 spare) non-zero elements

Global Statistics
8904 entities 0 sets 0 set members
Presolved problem has: 564 rows 7504 cols 33570 non-zeros
LP relaxation tightened

Iter  Obj Value 0 Minf Ninf Nneg Sum Inf Time
0 652.000000 0 259 0 255.000000 0
100 17567.00000 0 237 0 355.000000 0
200 34131.00000 0 304.000000 0
300 41235.00000 0 305 0 493.333333 0
400 45628.00000 0 311 0 933.333333 0
500 46916.00000 0 327 0 718.125000 0
600 47459.00000 0 335 0 831.937500 0
700 48167.75000 0 328 0 427.246080 0
800 48779.80000 0 348 0 513.475588 0
900 49463.22885 0 396 0 340.504621 0
1000 49072.20735 0 356 0 659.066239 0

20071119.170805.859
Strategy: Phase 1: basic 43/109

Objective
Best bound
# solutions
# nodes

Problem Statistics
884 (0 spare) rows
8904 (0 spare) structural columns
10859 (0 spare) non-zero elements

Global Statistics
8904 entities 0 sets 0 set members
Presolved problem has: 564 rows 7604 cols 33570 non-zeros
LP relaxation tightened

Iter  Obj Value 0 Minf Ninf Nneg Sum Inf Time
0 632.000000 0 259 0 255.000000 0
100 17587.00000 0 237 0 355.000000 0
200 34236.00000 0 230 0 399.000000 0
300 41237.00000 0 319 0 493.333333 0
400 45628.00000 0 311 0 933.333333 0
500 46916.50000 0 327 0 718.125000 0
600 47459.00000 0 335 0 831.937500 0
700 48167.75000 0 328 0 427.246080 0
800 48779.90000 0 348 0 513.475588 0
900 49463.32310 0 396 0 340.504621 0
1000 49072.00750 0 356 0 659.066239 0
Xpress-Tuner:

Tuning MIP families
Xpress-Tuner: Tuning Methods

Select one tuning method:
- Adaptive flexible comprehensive
- Adaptive flexible quick
- Adaptive flexible root focus
- Adaptive flexible tree focus
- Adaptive pure comprehensive
- Adaptive pure quick
- Adaptive pure root focus
- Adaptive pure tree focus

Other options:
- Improve solution
- Improve bound
- Find any solution

Customize:
- Load directives: [an option]
- Load solution: [an option]

Baseline control parameters:

Load factory defaults

Ready.
Benchmarking - Optimality

Problems with less than 60 minutes
Problems with less than 30 minutes
Problems with less than 30 seconds
Solve in 1/10 of time – examples Google
Matrix Gen

Reduce time
Parallel MIP
Opportunistic MIP

Reduce time
Parallel MIP
Deterministic MIP

Modeling Languages

Problems more than 2 hours

What is a good solution?
Multiple Solutions
Parallel Opportunistic
Explain Solutions
MIP Research

» More research on Deterministic Parallel
  » Predict Size of BB Tree/ Structure
  » Unbalanced Trees/how to Balance

» Solve Smaller problems faster (E Danna – Google)

» New Cuts/Feasibility Pump

» More CP in Presolve

» Hard Constraints / Soft Constraints – Operations in BB nodes

» Parallelize cuts/Heuristics in root node

» How to scale better 4 to 8 cores

» Data mining in Parameters – using supervised learning

» **** Adaptive search – change the parameters dynamically during the Search
Non Linear

» Knitro

» CONOPT

» Xpress SLP : good for process industry

» MINLP CMU-IBM , minlp.org

» BONMIN Basic Open Source Nonlinear Mixed Integer programming
Comparisons Mittelman

» Knitro vs IPOPT
» 27 better Knitro
» IPOPT better in 14 cases
Global Optimization other technologies

» Baron
» Disjunctive programming
» Robust
» Stochastic
Constraint Programming

» ILOG-CP
» CHIP
» Artelys – Kalis
» COMET – Dynadec
» SIMPL (Yunes, Aron, Hooker)

» Benchmarking CP codes????
Example of relaxation for alldifferent

\[
\text{alldiff}(x_1, \ldots, x_n)
\]
\[
x_j \in \{1, \ldots, n\}
\]

Convex hull relaxation, which is the strongest possible linear relaxation (*JNH, Williams & Yan*):

\[
\sum_{j=1}^{n} x_j = \frac{1}{2} n(n + 1)
\]
\[
\sum_{j \in J} x_j \geq \frac{1}{2} |J| (|J| + 1), \quad \text{all } J \subseteq \{1, \ldots, n\} \text{ with } |J| < n
\]

For \( n = 4 \):

\[
\begin{align*}
x_1 + x_2 + x_3 + x_4 &= 10 \\
x_1 + x_2 + x_3 &\geq 6, \quad x_1 + x_2 + x_4 \geq 6, \quad x_1 + x_3 + x_4 \geq 6, \quad x_2 + x_3 + x_4 \geq 6 \\
x_1 + x_2 &\geq 3, \quad x_1 + x_3 \geq 3, \quad x_1 + x_4 \geq 3, \quad x_2 + x_3 \geq 3, \quad x_2 + x_4 \geq 3, \quad x_3 + x_4 \geq 3 \\
x_1, x_2, x_3, x_4 &\geq 1
\end{align*}
\]
Automatic relaxations

The following constraints can be relaxed automatically with Xpress-Kalis:

- Linear constraints
  - `alldifferent(X)`
  - `occurrence(X,v) op Y`
  - `distribute(X,A,m,M)`
  - `X = Min/Max(Y)`
  - Absolute value `X = | Y |`
  - Distance `v op |X-Y|`
  - Element (2D) `X = A[I]` / `X = A[I][J]`
- Cycle
- Logical constraints (⇒, ⇔, or, and)
An integer knapsack problem with side constraint

\begin{align*}
\text{min} & \quad 5x_1 + 8x_2 + 4x_3 \\
\text{subj.to} & \quad 3x_1 + 5x_2 + 2x_3 \geq 30 \\
& \quad \text{alldiff}(x_1, x_2, x_3) \\
& \quad x_j \in \{1,2,3,4\}
\end{align*}

»MILP needs additional constraints and 0-1 variables to express the alldiff:

\begin{align*}
& \quad x_i = \sum_j jy_{ij}, \quad \text{all } i \\
& \quad \sum_j y_{ij} = 1, \quad \text{all } i
\end{align*}
Pure CP model in Xpress-KALIS

! knapsack constraint
3 * X1 + 5 * X2 + 2 * X3 >= 30

! side global constraint
all_different({X1,X2,X3})

! objective definition
benefit = 5 * X1 + 8 * X2 + 4 * X3

! initial propagation
rs := cp_propagate

! show constraint propagation bounds on objective
writeln("Constraints propagation gives ",benefit) |

! solve the problem
if (cp_minimize(benefit)) then
    ! output optimal solution to the screen
    cp_show_sol
end-if
HYBRID Model in Xpress-KALIS

! knapsack constraint
3 * X1 + 5 * X2 + 2 * X3 >= 30

! side global constraint
all_different([(X1,X2,X3)])

! objective definition
benefit = 5 * X1 + 8 * X2 + 4 * X3

! initial propagation
rs := cp_propagate

! show constraint propagation bounds on objective
writeln("Constraints propagation gives ",benefit)

declarations
  myrelax0 : cplinrelax
end-declarations

! build an automatic 'LP' oriented linear relaxation
myrelax0 := cp_get_linear_relaxation(0)

! output the relaxation to the screen
cp_show_relax(myrelax0)

! define the linear relaxation
cp_set_linear_relaxation(myrelax0,benefit,KALIS_TOPNODE_RELAX_SOLVER,0)

! define a 'MIP' style branching scheme using solution of the optimal relaxation
cp_set_branching(assign_and_forbid(KALIS_LARGEST_REduced_COST,KALIS_NEAREST_RELAXED_VALUE))

! solve the problem
if (cp_minimize(benefit)) then
  ! output optimal solution to the screen
  cp_show_sol
end-if
Modeling Platforms

- Modeling + Programming
- Model building libraries
- GUI
- Modeling languages
- Matrix generators

Time:
- 70s
- 80s
- 90s
- 2000s
Modeling Languages

» OPL: Integrates MIP and CP

» GAMS: All solvers available, NLP, MIP, Stochastic

» Mosel: Programming and Modeling together
Modeling Languages

» OPL: Integrates MIP and CP
» GAMS: All solvers available, NLP, MIP, Stochastic
» Mosel: Programming and Modeling together
More

» IBM ODM: Optimization Decision Manager
» Next generation : Eclipse
» Simulation
» Python link
Eclipse – Platform - Future

Standalone SWT Application

1. Introduction
2. Open the Java Perspective
3. Create a Java project
4. Configure the Java project
5. Create the HelloWorld class
6. Edit the main method
7. Create and run a Java Application launch configuration

Select Run -> Run... to open the Launch configuration dialog. In the list of Configurations, select Java Application and click the New button. On the Main tab, select the Java project and HelloWorld main class. On the Arguments tab, add the following to the VM arguments: -Djava.library.path=\{system:ECLIPSE_HOME\}/plugins/org.eclipse.swt.\{system:OS\} -U\{system:ARCH\}. Now click the Run button to launch application. The Launch configuration dialog is automatically displayed when you click the "Click to Perform" button.
**CP, MIP and their combination**

- **Optimization Technologies**
  - Mixed Integer Programming: MIP
  - Finite Domain Constraint Programming: CP

- **Planning & Scheduling**
  - Long and mid-term planning: MIP
  - Short-term planning, scheduling: CP

Supply chain optimization
Requires MIP, CP, and their combination
Solving Planning and Scheduling Models

»INTEGRATED ENVIRONMENT

»Common Planning and Scheduling Model

»LP/MIP

»CP
Combining MIP & CP

- Master problem
  - MIP
    - Sub problem
      - CP
        - Feasible
          - (OK)
        - Infeasible
          - Add Cut to MIP

- Mixed logical/Linear Programming framework
  (Hooker et al)
Project

» LISCOS Project
  » European Union 5th Framework

» BASF - Chemicals

» P&G - Food

» PSA - Cars

» Barbot - Paint
Production Scheduling Problem

» Schedules Crude-Oil Blend-Shops in Oil-Refineries.
» In the Category of “Closed-Shop” Scheduling.
» Involves Quantity, Logic and Quality Decisions.
  » Quantity = Flow, Rate, Inventory
  » Logic = Mixing-Delay, Up-Time, One-Flow-In
  » Quality = Octane, Sulphur, Yield, Temperature
» Intended to Reduce Quality Variation.
Production Scheduling Problem

» Finite-Capacity Scheduling.

» Supports Marine and Pipeline Access Blend-Shops.
Production Scheduling Technology

» Involves the following OR Problems:
  » Fixed-Charge Network Flow
  » Lot-Sizing and Scheduling (Small Time-Bucket)
  » Facility-Location
  » Multi-Linear Pooling

» Uses Discrete-Time Formulation.

» Allows for all Logic Constraints to be Configurable.

» Creates Elastic/Penalty Variables for all Constraints.

» Modeled using Xpress-Mosel.
Production Scheduling Technology

» Involves the following Primal Heuristics programmed using Xpress-Mosel:
  » Relax-and-Fix
  » Dive-and-Fix
  » Fixed-Charge Adjustment
  » Chronological Decomposition
  » Smooth-and-Dive
  » Incumbent Elimination
  » Disaggregation and Warm-Start
  » Parallel Diving (Sorted One-Point Crossover)
  » Local Search Repair of Infeasible Solutions

» Embeds Xpress-Optimizer and Xpress-SLP.
Production Scheduling Approach

» Finds Optimized or Feasible Schedules.
» Maximizes Profit and Performance.

» At the core is Bender’s Decomposition.

» Uses Intelligent Problem Solving structure:
  » Solve Quantity-Logic (Q-L) first using MILP
  » Fix Logic, then solve for Quantity-Quality (Q-Q) using SLP
  » If Q-L infeasible then Q-Q will be infeasible
  » If Q-L feasible then Q-Q may or may not be feasible
  » (If Q-Q feasible then Q-L feasible)
Production Scheduler Demo
Existing Products with Xpress-MP

» Finished Product Blend Planning (BLEND) uses Xpress-Optimizer.

» Refinery and Petrochemical Planning (RPMS) uses Xpress-Optimizer.

» Supply and Distribution Planning (SAND) to use Xpress-Optimizer.
Examples of Production Systems
Petroleum Refining

» Includes many tanks, atmospheric & vacuum fractionators, blenders, reactors, separators, etc. as well as 1100 components and 50 properties.

» Logistics has 160K rows, 95K columns, 11K binaries: 12-seconds.

» Quality has 250K L rows, 25K NL rows, 100K columns: 900-seconds.
Fast-Moving Consumer Goods

» Includes an highly sequence-dependent edible oil-deodorizer (prize-collecting traveling salesman problem) and many types of edible oil customer demands.

» Logistics-only has 130K rows, 57K columns, 8K binaries: 400-seconds.
Oil & Gas Processing

» Includes spheroids, spheres, flash drums, stabilizers, vaporizers, splitters, etc.
  » Logistics has 60K rows, 30K columns, 9K binaries: 15-seconds.
  » Quality has: 70K L rows, 5K NL rows, 24K columns: 100-seconds.
Metals Casting

» Includes arcing-furnaces, decarburizers, cranes, ladles, batch and continuous casters – similar to open-shop/job-shop scheduling.

» Logistics-only has 75K rows, 24K columns, 3.5K binaries: 200-seconds.
Petrochemical Cracking & Separating

» Includes cracking furnaces, compressors, distillation towers, cold-box, tanks, bullets, etc.

» Logistics has 21K rows, 10K columns, 2.5K binaries: 50-seconds.

» Quality has 21K L rows, 2K NL rows, 8K columns: 5-seconds.
Steam & Power Generating

» Includes boilers, turbo-generators, steam headers, turbines, relief-values, etc.

» Logistics-only has 24K rows, 13K columns, 2K binaries: 30-seconds.