FICO is the leader in Decision Management — transforming business by making every decision count
Agenda

» FICO
  » Overview
  » Decisioning
  » Optimisation tools
  » Case study

» FICO Xpress
  » Overview
  » Recent Developments
    » MIP
    » LP
    » QP
    » Mosel
### Profile
- The leader in decision management
- Founded: 1956
- NYSE: FIC
- Revenues: $745 million (fiscal 2008)

### Products & Services
- Analytics: scores and models
- Decision management applications
- Decision management tools

### Clients & Market
- 5,000+ clients in 80 countries
- Primary Industries: Financial services, insurance, retail, healthcare

### Offices
- 20+ offices worldwide
- HQ in Minneapolis, Minnesota
- Regional Hubs: London, Birmingham (UK), Madrid, Sao Paulo, Bangalore, Beijing, Singapore
We Take a Systematic Approach to Decisions

**Automate complex decisions in real time**
- Increase consistency
- Reduce manual reviews
- Increase speed to market

**Improve decision quality with analytics**
- Reduce credit, fraud, claims losses
- Identify the best next offer
- Optimize results

**Connect decisions across the enterprise**
- Manage across product lines and business silos
- Execute coordinated customer-level strategies
Increase the Success of Your Decisions

<table>
<thead>
<tr>
<th>Predictive analytics for greater precision</th>
<th>Business rules management for greater control</th>
<th>Optimization for new levels of performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>» Execute more relevant and targeted customer offers</td>
<td>» Control decisions across business lines and geographic borders</td>
<td>» Advance your business strategy systematically, with every decision</td>
</tr>
<tr>
<td>» Control risk exposure and reduce losses</td>
<td>» Change faster than the competition to seize new opportunities</td>
<td>» Assign optimal actions to reach specified objectives</td>
</tr>
<tr>
<td>» Identify and stop fraud faster</td>
<td>» Comply with regulatory requirements faster and at lower cost</td>
<td></td>
</tr>
</tbody>
</table>
We Offer the Most Complete Set of Solutions for Decision Management

- Simplify & reduce cost of developing, managing & updating complex decision processes
- Operationalize & measure complex, high-volume decisions across the customer lifecycle
- Improve predictions & determine optimum decisions

Decision Management

Tools

Applications

Analytics
Fair Isaac’s Optimization Offerings

» Optimization software tools
  » Xpress product suite
    » Mathematical modelling
    » Optimization
  » Decision Optimizer

» Optimization software together with business rules management system
  » Xpress and Blaze Advisor® business rules management system

» Industry-specific point solutions, optimization software, business rules and analytics
  » Pricing optimization
  » Shelf optimization
  » Marketing optimization
Debt Consolidation — Problem Statement

» Consolidation Problem
  » Based on user inputs and preferences determine:
    » Loan product, exact loan amount, outgoing debt ratio, etc.
  » While managing the payoff of debts

Credit Card A — owe $2000
Annual Interest 12.5%

Credit Card B — owe $1000
Annual Interest 15%

Credit Card C — owe $5000
Annual Interest 17.5%

Desired Loan Amount
First Mortgage Amount
Home Value and etc

Optimization Engine
Minimize repayment period
Minimize Monthly Payment
Loan Amount
Monthly Payment
Which debt to payoff
Business Rules: Eligibility: Product

Eligible Products:
- Loan A
- Loan B
- ...
- Loan M
FICO Xpress
» 25+ years of experience in modelling and optimization
» 20+ years of experience in mixed integer optimization
» 10+ years of experience in nonlinear optimization
» 9 years Xpress-Mosel, modelling and solving environment

» Focus on
  » (potentially) exact solution methods and
  » integrating modelling and optimization

» Since Januar 2008 part of Fair Isaac / FICO
Fair Isaac Optimization Offers Multiple Solvers

- Solvers
- GUI / studio
- Extensions / NI
- Vertical applications
- Modeling platform

Solvers:
- LP
- MIP
- MIQP
- MISLP
- QP
- QCQP
- MIQCQP
- Heuristics
- CP
- SP
- Kalis
- Mosel
- IVE
- XAD
- User Extension
What’s in the Integrated Suite?

   » A set of robust, high performance multi-threaded solvers covering all of the major optimization problem types.

» Xpress-Mosel
   » An easy-to-learn rapid development modelling language.

» Xpress-BCL
   » A powerful API for programmatic model construction.

» Xpress-IVE
   » Visual development environment with deployment wizards.

» Xpress-Tuner
   » Automatic configuration tuning for maximum performance.
## Benefits Our Customers Value

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Customer Insight</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Superior technology</strong></td>
<td>» Provide reliable solutions to problems with millions of variables and constraints</td>
</tr>
<tr>
<td></td>
<td>» Able to solve numerically difficult or unstable problems</td>
</tr>
<tr>
<td></td>
<td>» Finds high quality solutions – fast</td>
</tr>
<tr>
<td><strong>Greater efficiencies</strong></td>
<td>» Rapid prototyping, analysis and deployment</td>
</tr>
<tr>
<td><strong>Ease of use</strong></td>
<td>» Sophisticated yet easy-to-use tools for building, solving, testing and deploying optimization models</td>
</tr>
</tbody>
</table>
# Key Features and Benefits of Xpress-Mosel

<table>
<thead>
<tr>
<th>Features</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced programming languages:</td>
<td>Entire Mathematical Model can be stored in one place for rapid development and easy maintenance.</td>
</tr>
<tr>
<td>» Algebraic modeling language</td>
<td></td>
</tr>
<tr>
<td>» Procedural programming language</td>
<td></td>
</tr>
<tr>
<td>Utilize different solvers in the same model</td>
<td>From Mosel you can solve LPs, MIPs, MIQPs, Non-Linear problems, Stochastic problems, and Constraint problems</td>
</tr>
<tr>
<td>Decompose &amp; parallelize a model to take advantage of multiple CPUs/cores</td>
<td>Faster solve times</td>
</tr>
<tr>
<td>Build a GUI exclusively within Mosel code</td>
<td>Decreases development time, gets optimization in front of business user quicker</td>
</tr>
<tr>
<td>Portable across operating systems</td>
<td>Mosel Model compiled in one OS can be deployed on all other supported Operating Systems, decreasing development time</td>
</tr>
<tr>
<td>Open, modular architecture, User extensible</td>
<td>User flexibility to solve the most complicated optimization problems</td>
</tr>
<tr>
<td>Compiled</td>
<td>Protects intellectual property</td>
</tr>
</tbody>
</table>
Xpress-IVE: Mosel & Optimizer

- Editor
- Debugger
- Profiler
- Progress graphs
- Visualization
- Wizards
- Mosel extensions
- Deployment
Xpress Customers

- CBS
- NFL
- ORACLE
- AVIS
- P&G
- Budget
- Honeywell
- aspentech
- American Airlines
- Jeppesen
- RedPrairie
- Amazon.com
Xpress 7 - MIP performance

» Multi-threaded performance
» Reengineered B&B tree
» Improved cuts, branching
» Parallel heuristics
» Improved decision space reduction (presolve)
» N-Best and the multiple solution pool
Xpress 7 vs 2008A on the “internal” sets

<table>
<thead>
<tr>
<th></th>
<th>2008A 1 thread</th>
<th>Xpress 7 1 thread</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Internal” Number of instances solved</td>
<td>277</td>
<td>294</td>
<td>+6%</td>
</tr>
<tr>
<td>“Internal” Total solution time</td>
<td>63,000</td>
<td>37,375</td>
<td>-70%</td>
</tr>
</tbody>
</table>

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

Internal test set of 320 public and customer models
Reengineered B&B tree

» No global file
  (unless getting close to – adjustable – memory limit)

» Auto compression of node information

» New user branching object
  allows user to add own global entities,
  let the optimizer decide to pick the most promising one
  (essential just provide a list of candidate branches)
New MIP features

» Improved cuts including **zero-half cuts**
   (combining integer rows with 0.5 weights to get integer row with an odd rhs)

» Branching on **split disjunctions** (e.g. on sum of variables)
   (e.g. branch on the sum of generals)

Efficient for general integers
» **Parallel heuristics**
   Efficient on problems where the root solve is expense

» **Improved decision space reduction (presolve)**
   (extended probing and implications from multiple rows)
Multiple Solution Storage - revisited

» A MIP Solution Pool (MSP) stores the solution vector values for multiple solutions

» This can be useful in cases when there are constraints or costs not reflected in the problem that the user wants to use to select a solution from a set that have been found by Xpress

» Many functions to manage/query the pool
  » e.g. the user perhaps want to get the list of solutions that are feasible for a given problem

» Solution Pools have Attributes
  » e.g. number of solution nonzros
  » e.g. number of solutions in the pool
  » e.g. objective value of a solution in the pool for a given problem
Computing the $N$-best Solution

» MIP Solution Enumerator (MSE)

» Runs a customized MIP search on a user provided problem (XPRSprob)

» The search is customized such that nodes are not cut-off by bounding and integer solution nodes are branched

» The MSE stores the solutions found in a user provided MIP Solution Pool

» Is useful for generating a set of solutions for a problem

» It can be used to generate the $N$-best solutions to a problem

/* Run the enumeration */

nMaxSols = 10;
XPRS_mse_minim(mse, prob, msp, XPRS_mse_defaulthandler, NULL, &nMaxSols);

» XPRS_mse_defaulthandler function manages the storage of at most nMaxSols (=10) solutions

» either rejects the current solution or deletes the worst existing solution depending on their objective values
In addition to the MIP objective, the MSE provides a metric for solutions based on the ‘diversity’ of the solution with respect to the other stored solutions.

The user can delete $p$ solutions based on the MIP objective values and then delete the remaining $n_{MaxSols} - p$ solutions based on a diversity metric.

The diversity metric for a solution is the sum of difference metrics between the solution and the other stored solutions.

The user can provide their own difference metric calculation for solution pairs by using a callback.
Xpress 7 – Linear models

» Barrier improvements
» Improved primal simplex performance
“Serial” LP improvements

- Improved barrier sparser

- Steepest edge pricing in primal simplex
  (18% average speed up)

- Dual simplex improvements
  (9% average speed up)
LP Performance across releases

LP Performance

Internal test set of 796 public and customer models
Xpress 7 – Nonlinear models

- Improved QCQP
- New problem class: MIQCQP
- Linear constrained convex nonlinear solver
- Dramatically improved quadratic simplex
The convexity checker now attempts to reformulate nonconvex problems if possible, like binary MIQP / MIQCQP.

Greatly improved CNP / QCQP presolve.

Notes on CNP:

- Given any solution to the problem, these callbacks are used to evaluate the value, the gradient and the Hessian of the nonlinear objective respectively.

- The problem is expected to be convex, which means that all Hessians must be defined and positive semi-definite for minimization (negative for maximization) (even outside the feasible region).
The many Qs

» Quadratically Constrained Quadratic Programming (QCQP) solver:
  » support for very large sparse problems
  » greatly increased QCQP presolve

» New model class support:
  Mixed Integer Quadratically Constrained Quadratic Programming (MIQCQP)
» Reengineered quadratic primal simplex  
(dramatic performance improvement)

» New quadratic dual simplex  
(fast dual reoptimize, less fractional solutions)

» Spectacular MIQP improvements on many problem classes  
(expected to improve even further)
## Xpress 7 vs 2008A on the “internal” test sets

<table>
<thead>
<tr>
<th></th>
<th>2008A</th>
<th>Xpress 7</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Quadratic primal</strong></td>
<td>283,107 (3227)</td>
<td>44,296 (3299)</td>
<td>-539% (+3%)</td>
</tr>
<tr>
<td><strong>QCCP barrier</strong></td>
<td>3,510 (210)</td>
<td>219 (227)</td>
<td>-1500% (+8%) (presolve)</td>
</tr>
<tr>
<td><strong>MIQP barrier</strong></td>
<td>58,168 (142)</td>
<td>38,002 (195)</td>
<td>-53% (+35%)</td>
</tr>
<tr>
<td><strong>Convex barrier</strong></td>
<td>547 (255)</td>
<td>142 (280)</td>
<td>-284% (+10%)</td>
</tr>
</tbody>
</table>
Xpress 7 – Mosel

» Several Models in One File
» Advanced Modelling
» Automatic Finalization and Counters
» XAD Drag and Drop
Multi-problem vs. multi-model

» Multi-problem (mpproblem)
  » single model file
  » problems share data
  » integrated; no direct access to (sub)problems by other models/applications
  » sequential access to problems

» Multi-model (mmjobs)
  » several model files
  » communication of data (in memory)
  » stand-alone execution of submodels or use of submodels with other master models/applications
  » sequential or parallel execution of models
Multiple problem handling

» The new statement with allows to open a problem (= select the active problem):

```
declarations
  myprob: mpproblem
end-declarations

...

with myprob do
  x+y >= 0
end-do
```
Multiple problem handling

» Problem types support assignment: \( P_1 := P_2 \)
   and additive assignment: \( P_1 += P_2 \)

» The same decision variable (type \text{mpvar}) may be used in several problems

» Constraints belong to the problem where they are defined
Module advmod: Logic constructs

- Type `logctr` for defining and working with logic constraints in MIP models
- Implementation of these constraints is based on indicator constraints
- Build logic constraints with linear constraints using the operations `and, or, xor, implies, and not`
- Must include the package `advmod` instead of the Optimizer library `mmxprs`
» The aggregate operator count returns the number of times that a test succeeds

\[ S := \{1, 5, 8, -1, 4, 7, 2\} \]

\[
\text{writeln("Number of odd numbers in S: ", count(i in S | isodd(i))) }
\]

» Use the construct as counter to specify a counter variable in a bounded loop (i.e., forall or aggregate operators such as sum): at each iteration, the counter is incremented

\[ S := \{1, 5, 8, -1, 4, 7, 2\} \]

\[ \text{cnt}: = 0.0 \]

\[
\text{writeln("Average of odd numbers in S: ", (sum(cnt as counter, i in S | isodd(i)) i) / cnt)}
\]
Xpress 7 – Something extra
Due to customer demand, the solvable model size was increased by a factor of 2.

Examples of user problems solved by Xpress:

<table>
<thead>
<tr>
<th>Rows</th>
<th>Columns</th>
<th>Nonzeros</th>
<th>Entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>55 501</td>
<td>2 313 900</td>
<td>406 828 000</td>
<td>2 313 500</td>
</tr>
<tr>
<td>1 344 660</td>
<td>18 525 200</td>
<td>244 245 000</td>
<td>0</td>
</tr>
<tr>
<td>1 344 660</td>
<td>18 525 200</td>
<td>244 245 000</td>
<td>0</td>
</tr>
<tr>
<td>11 741</td>
<td>2 339 830</td>
<td>200 214 000</td>
<td>2 339 430</td>
</tr>
<tr>
<td>897 662</td>
<td>9 488 130</td>
<td>152 017 000</td>
<td>0</td>
</tr>
<tr>
<td>1 687 180</td>
<td>10 000 000</td>
<td>121 892 000</td>
<td>0</td>
</tr>
<tr>
<td>229 441</td>
<td>28 765 300</td>
<td>86 084 800</td>
<td>0</td>
</tr>
<tr>
<td>554 013</td>
<td>6 327 120</td>
<td>79 489 300</td>
<td>0</td>
</tr>
<tr>
<td>8 917 760</td>
<td>16 207 100</td>
<td>77 966 100</td>
<td>12 497</td>
</tr>
<tr>
<td>1 200 130</td>
<td>4 884 110</td>
<td>72 813 678</td>
<td>0</td>
</tr>
<tr>
<td>546 304</td>
<td>6 163 666</td>
<td>69 583 836</td>
<td>0</td>
</tr>
<tr>
<td>160 218</td>
<td>19 884 900</td>
<td>59 507 200</td>
<td>0</td>
</tr>
<tr>
<td>436 562</td>
<td>5 500 097</td>
<td>58 915 368</td>
<td>0</td>
</tr>
<tr>
<td>644 584</td>
<td>5 628 760</td>
<td>57 468 100</td>
<td>0</td>
</tr>
<tr>
<td>5 034 580</td>
<td>7 448 220</td>
<td>53 129 400</td>
<td>7 215</td>
</tr>
</tbody>
</table>
» Hard MIPs can benefit from speedups at the range of 10X with proper problem class specific tuning.

» The tuner provides a convenient way of checking what you (or us) will check first anyway…

» … taking it to lengths humans just don’t have the time to.
Xpress-Tuner: How to Tune (Automatically) an Optimization Problem?

Matrix or set of matrices:
- File name (LP or MPS or SET): C:\xpressMP\examples\tuner\air04.mps
- Create a set of matrices...
- Objective: Minimize, Maximize
- View past results
- Delete past results

Goal:
- Target gap: 0%
- Target run time: 5 s

If gap AND time targets are both met:
- Reduce time to gap
- Reduce gap

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

Customize:
- Load directives (<File name>.dir)
- Load solution (<File name>.slx)

Baseline control parameters:

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
- Combinations
- Mixer
- Single run 1
- Single run 2

Run: 1 simultaneous threads
- Stop tuning after: 86400 s

Start

Adaptive flexible

"Adaptive flexible" starts by running all the basic strategies, one by one. The top 'N' strategies are mated with each other in pair-wise fashion to create a second generation. After that, the evolutionary selection algorithm begins: each new strategy is obtained by mating the best strategy so far with the second best. If there's a conflict, the third best will be selected, etc. Every 'R'-th run, the "worst performing" control parameter is eliminated from the current strategy.

Relaxation:
- Primal
- Dual
- Barrier
- Network

Basic strategies, one per line:
- COVERCUTS=0
- COVERCUTS=2
- COVERCUTS=5
- COVERCUTS=20
- CUDEPHT=0
- CUDEPHT=10
- CUDEPHT=20
- CUDEPHT=1000
- CUTFREQ=0
- CUTFREQ=2
- CUTFREQ=5

Load factory defaults

Ready.
Xpress-Tuner: Tuning Process

Your goal:
Seeking to: Reduce time to gap
Target gap: 0 %
Within: 5 s

Progress update:
Time elapsed: 1m 36s
Time remaining: 23h 58m 24s
Improvement: YES

Current strategy:
Thread 1

Phase 1: basic 18/109
Dual simplex for the LP relaxation.
GOMCUTS=0
Auxiliary:
MAXTIME=5
MIPRELSTOP=0

Optimizer output:
824 [ 0 spare] rows
8904 [ 0 spare] structural columns
81869 [ 0 spare] non-zero elements
Global Statistics:
8904 entities 0 sets 0 set members

Best settings:
Dual simplex for the LP relaxation.
CUTSELECT=-8193
Auxiliary:
MAXTIME=5
MIPRELSTOP=0
### Xpress-Tuner: Detailed Results

#### Strategy Rankings

<table>
<thead>
<tr>
<th>Machine</th>
<th>Xpress Version</th>
<th>When</th>
<th>Alg</th>
<th>File</th>
<th>Max Time</th>
<th>Hrs Executed</th>
<th>FTF</th>
<th>RPM</th>
<th>TSP</th>
<th>GAP</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>HORIAMD 10.00.00</td>
<td>2007/11/17</td>
<td>08:53:203</td>
<td>Dual</td>
<td>msc</td>
<td>-5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HORIAMD 10.00.00</td>
<td>2007/11/17</td>
<td>09:05:259</td>
<td>Dual</td>
<td>msc</td>
<td>-5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HORIAMD 10.00.00</td>
<td>2007/11/17</td>
<td>09:22:171</td>
<td>Dual</td>
<td>msc</td>
<td>-5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HORIAMD 10.00.00</td>
<td>2007/11/17</td>
<td>09:46:295</td>
<td>Dual</td>
<td>msc</td>
<td>-5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### For the record

- One run per line

#### Control parameters


#### Results

- GAP
- Rank

---

THANK YOU

Oliver Bastert
FICO Xpress Team
oliverbastert@fico.com

Berlin, 29 Sep, 2009