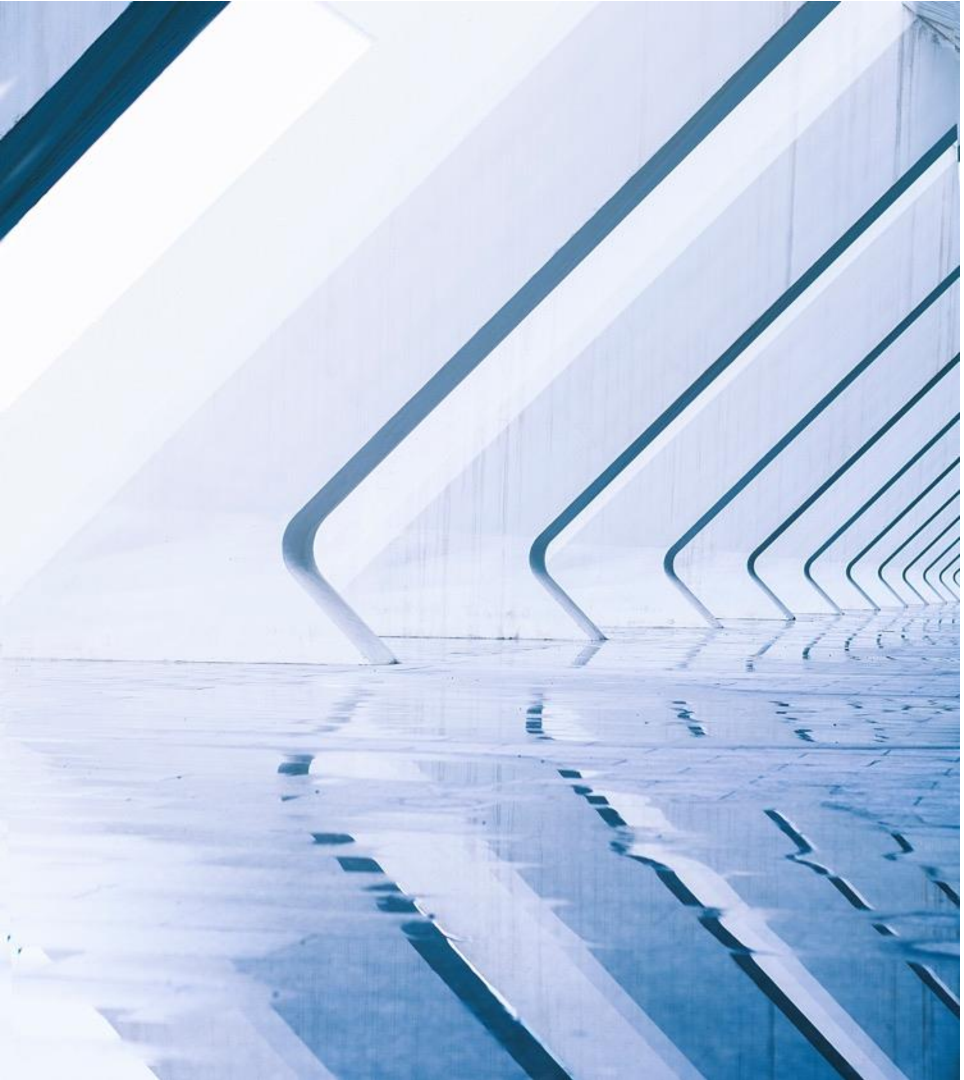


Learning To Scale

Timo Berthold

FICO Xpress Optimization

FICO Xpress Optimization

Optimization Solutions

for use by

Business Users and

Business Analysts

Xpress Solutions

Optimization Technology

4 Modules

for use by

Operations Researchers,

Data Scientists and

Solution Developers

Xpress Mosel

Xpress Workbench

Develop models, services and solutions

Xpress Insight

Visualize, collaborate and decide using Solutions

Xpress Executor

Deploy and execute models

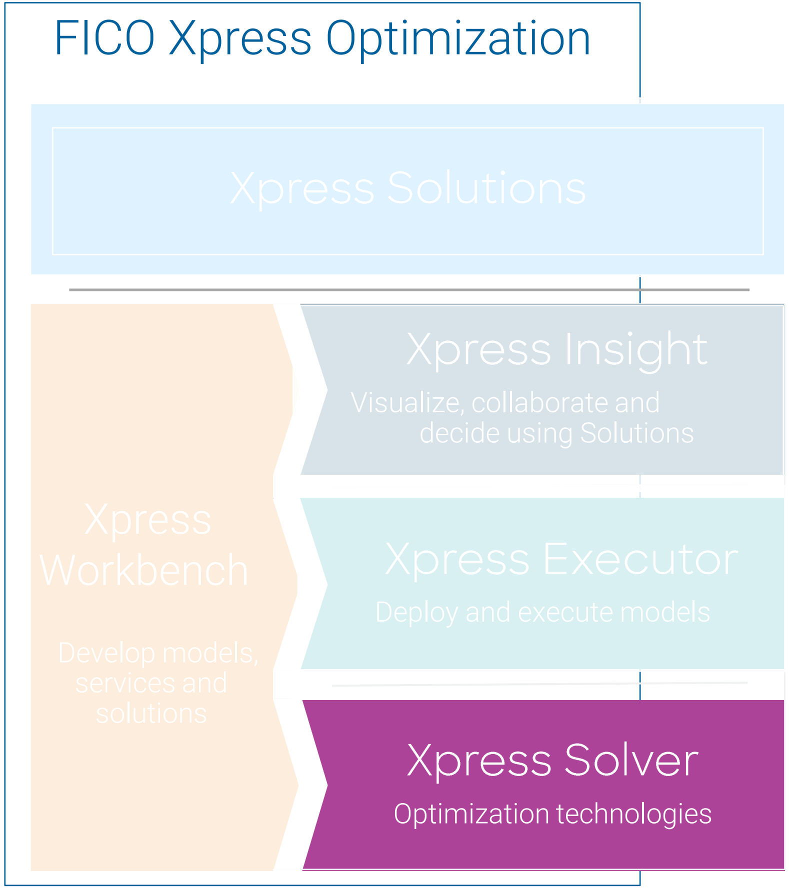
Xpress Solver

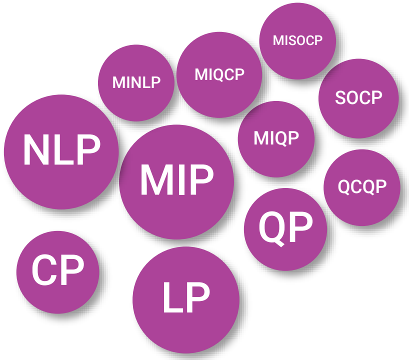
Optimization technologies

FICO Professional Services

Make Decisions that meet your Objectives

FICO Xpress Optimization – Optimization Technology





Performance → Widest breadth of industry-leading optimization algorithms and technologies

Features → Highest degree of customizability through advanced callbacks, permissive controls and flexible modelling

Robustness → Highest degree of determinism and a variety of features for numeric stability

Motivation: ML for Mathematical Optimization

Huge Activity in Machine Learning for Mathematical Optimization

Ding et al.	Accelerating Primal Solution Findings for Mixed Integer Programs Based on Solution Prediction	2019
Bertsimas & Stellato	Online Mixed-Integer Optimization in Milliseconds	2019
Fischetti & Fraccaro	Machine learning meets mathematical optimization to predict the optimal production of offshore wind parks	2018
Misra et al.	Learning for constrained optimization: Identifying optimal active constraint sets	2018
Bertsimas & Stellato	The Voice of Optimization	2018
Tang et al.	Reinforcement Learning for Integer Programming: Learning to Cut	2019
Baltean-Lugojan et al.	Selecting cutting planes for quadratic semidefinite outer-approximation via trained neural networks	2018
Etheve et al.	Reinforcement Learning for Variable Selection in a Branch and Bound Algorithm	2020
Zarpellon et al.	Parameterizing Branch-and-Bound Search Trees to Learn Branching Policies	2020
Yang et al.	Learning Generalized Strong Branching for Set Covering, Set Packing, and 0-1 Knapsack Problems	2020
Song et al.	Learning to Search via Retrospective Imitation	2019
Gasse et al.	Exact Combinatorial Optimization with Graph Convolutional Neural Networks	2019
Lee et al.	Learning to Branch: Accelerating Resource Allocation in Wireless Networks	2019
Hansknecht et al.	Cuts, Primal Heuristics, and Learning to Branch for the Time-Dependent Traveling Salesman Problem	2018
Balcan et al.	Learning to branch	2018
Václavík et al.	Accelerating the branch-and-price algorithm using machine learning	2018
Hottung et al.	Deep Learning Assisted Heuristic Tree Search for the Container Pre-marshalling Problem	2017
Lodi & Zarpellon	On learning and branching: a survey	2017
Alvarez et al.	A Machine Learning-Based Approximation of Strong Branching	2017
Alvarez et al.	Online Learning for Strong Branching Approximation in Branch-and-Bound	2016
Khalil et al.	Learning to branch in mixed integer programming	2016
Khalil	Machine Learning for Integer Programming	2016
He et al.	Learning to Search in Branch and Bound Algorithms	2014
Alvarez et al.	A Supervised Machine Learning Approach to Variable Branching in Branch-And-Bound	2014
Di Liberto et al.	Dynamic Approach for Switching Heuristics	2013
Sabharwal et al.	Guiding Combinatorial Optimization with UCT	2012

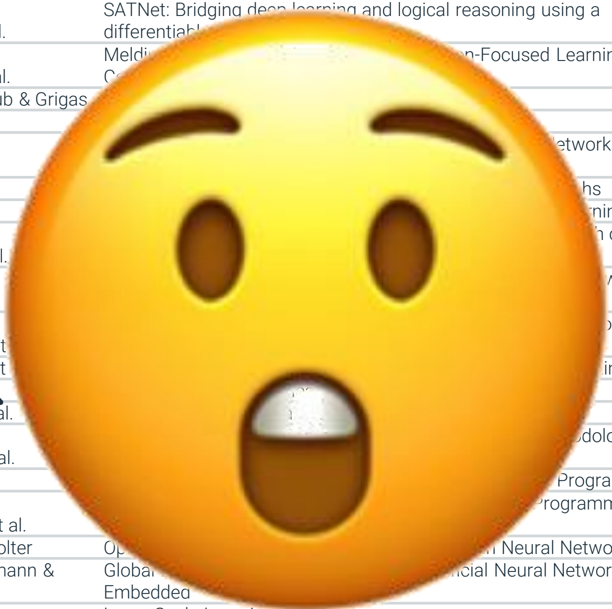
Khalil et al.	Learning to Run Heuristics in Tree Search	2017
Hutter et al.	Algorithm Runtime Prediction: Methods & Evaluation	2012
Hutter et al.	Automated Configuration of Mixed Integer Programming Solvers	2010
Ferber et al.	MIPaaL: Mixed Integer Program as a Layer	2019
Wilder et al.	End to end learning and optimization on graphs	2019
Wang et al.	SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver	2019
Wilder et al.	Melding the Data-Decisions Pipeline: Decision-Focused Learning for Combinatorial Optimization	2018
Elmachtoub & Grigas	Smart "Predict, then Optimize"	2017
Kool et al.	Attention, Learn to Solve Routing Problems!	2018
Li et al.	Combinatorial Optimization with Graph Convolutional Networks and Guided Tree Search	2018
Dai et al.	Learning Combinatorial Optimization Algorithms over Graphs	2017
Bello et al.	Neural Combinatorial Optimization with Reinforcement Learning	2016
Bowly et al.	Generation techniques for linear programming instances with controllable properties	2017
Bowly	Stress testing mixed integer programming solvers through new test instance generation methods	2019
François et al.	How to Evaluate Machine Learning Approaches for Combinatorial Optimization: Application to the Travelling Salesman Problem	2019
Fischetti et al.	Learning MILP Resolution Outcomes Before Reaching Time-Limit	2018
Kuhlmann	Learning to steer nonlinear interior-point methods	2019
Kruber et al.	Learning when to use a decomposition	2018
Bengio et al.	Machine Learning for Combinatorial Optimization: a Methodological Tour d'Horizon	2018
Hendel	Adaptive Large Neighborhood Search for Mixed Integer Programming	2018
Bonami et al.	Learning a Classification of Mixed-Integer Quadratic Programming Problems	2017
Amos & Kolter	OptNet: Differentiable Optimization as a Layer in Neural Networks	2017
Schweidtmann & Mitsos	Global Deterministic Optimization with Artificial Neural Networks Embedded	2018
Sculley	Large Scale Learning To Rank	2020
Song et al.	A General Large Neighborhood Search Framework for Solving Integer Programs	2020

Huge Activity in Machine Learning for Mathematical Optimization

Only very few of these implemented in general purpose MIP solvers!
(and activated by default)

Ding et al.	Accelerating Primal Solution Findings for Mixed Integer Programs Based on Solution Prediction	2019
Bertsimas & Stellato	Online Mixe	2019
Fischetti & Fraccaro	Mac	
Misra et al.		
Bertsimas & S		
Tang et al.		
Baltean-Lug		
Etheve et al.		
Zarpellon et al.		
Yang et al.	Lea 0-1 Knapsa	20
Song et al.	Learning to Search via	
Gasse et al.	Exact Combinatorial Optimization with Graph Convolutional Neural Networks	
Lee et al.	Learning to Branch: Accelerating Resource Allocation in Wireless Networks	2019
Hansknecht et al.	Cuts, Primal Heuristics, and Learning to Branch for the Time-Dependent Traveling Salesman Problem	2018
Balcan et al.	Learning to branch	2018
Václavík et al.	Accelerating the branch-and-price algorithm using machine learning	2018
Hottung et al.	Deep Learning Assisted Heuristic Tree Search for the Container Pre-marshalling Problem	2017
Lodi & Zarpellon	On learning and branching: a survey	2017
Alvarez et al.	A Machine Learning-Based Approximation of Strong Branching	2017
Alvarez et al.	Online Learning for Strong Branching Approximation in Branch-and-Bound	2016
Khalil et al.	Learning to branch in mixed integer programming	2016
Khalil	Machine Learning for Integer Programming	2016
He et al.	Learning to Search in Branch and Bound Algorithms	2014
Alvarez et al.	A Supervised Machine Learning Approach to Variable Branching in Branch-And-Bound	2014
Di Liberto et al.	Dynamic Approach for Switching Heuristics	2013
Sabharwal et al.	Guiding Combinatorial Optimization with UCT	2012

Khalil et al.	Learning to Run Heuristics in Tree Search	2017
Hutter et al.	Algorithm Runtime Prediction: Methods & Evaluation	2012
Hutter et al.	Automated Configuration of Mixed Integer Programming Solvers	2010
Ferber et al.	MIPaaL: Mixed Integer Program as a Layer	2019
Wilder et al.	End to end learning and optimization on graphs	2019
ang et al.	SATNet: Bridging deep learning and logical reasoning using a differentia	2019
r et al.	Meldi n-Focused Learning for C	2018
chtoub & Grigas		2017
et al.		2018
et al.	etworks and	2018
Dai et al.	hs	2017
Bello et al.	ning	2016
Bowly et al.	h controllable	2017
Bowly	w test	2019
cois et	rial	2019
tti et	imit	2018
Kuhlma		2019
Kruber et al.		2018
Bengio et al.	dological Tour	2018
Hendel	Programming	2018
Bonami et al.	rogramming	2017
Amos & Kolter	Op Neural Networks	2017
Schweidtmann & Mitsos	Global Embedded cial Neural Networks	2018
Sculley	Large Scale Learning To Rank	2020
Song et al.	A General Large Neighborhood Search Framework for Solving Integer Programs	2020



Huge Activity in Machine Learning for Mathematical Optimization

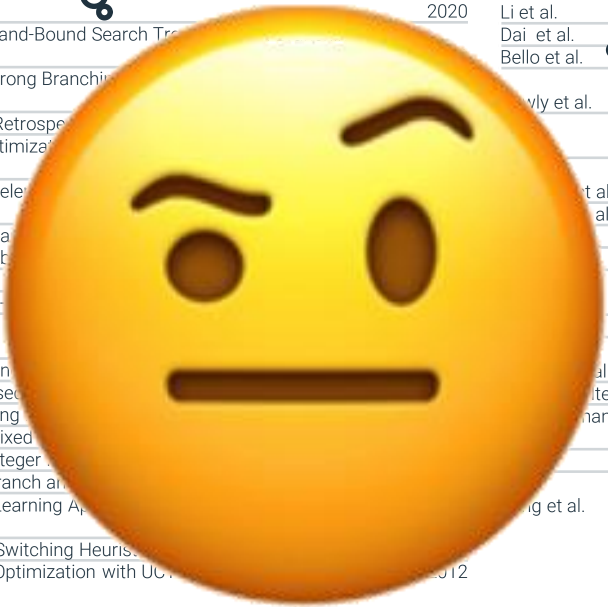
Ding et al.	Accelerating Primal Solution Findings for Mixed Integer Programs Based …	2019
Bertsimas & S…		2019
Fisch…	… optimal	2018
Misra e…		2018
Bertsim…		2018
Tang et al.	… Learning to Cut	2019
Baltean-Lugojan et al.	Selecting cutting p… …tic semidefinite outer-approximation via trained neural networks	2018
Etheve et al.	Reinforcement Learning for Varia… Selection in a Branch and Bound Algorithm	2020
Zarpellon et al.	Parameterizing Branch-and-Bound Search Tr… Policies	
Yang et al.	Learning Generalized Strong Branchi… 0-1 Knapsack Problems	
Song et al.	Learning to Search via Retrospe…	
Gasse et al.	Exact Combinatorial Optimiza… Networks	
Lee et al.	Learning to Branch: Accele… Networks	
Hansknecht et al.	Cuts, Primal Heuristics, a… Traveling Salesman Prob…	
Balcan et al.	Learning to branch	
Václavík et al.	Accelerating the branch-…	
Hottung et al.	Deep Learning Assisted … marshalling Problem	
Lodi & Zarpellon	On learning and branchin…	
Alvarez et al.	A Machine Learning-Base…	
Alvarez et al.	Online Learning for Strong…	
Khalil et al.	Learning to branch in mixed…	
Khalil	Machine Learning for Integer …	
He et al.	Learning to Search in Branch a…	
Alvarez et al.	A Supervised Machine Learning A… Branch-And-Bound	
Di Liberto et al.	Dynamic Approach for Switching Heuris…	
Sabharwal et al.	Guiding Combinatorial Optimization with UC…	…012

Khalil et al.	Learning to Run Heuristics in Tree Search	2017
Hutter et al.	Algorith… …tion	2012
Hutter et…	…ing Solvers	2010
Ferbe…		2019
W…		2019
	…a	
…		2019
Wil…	…ased Learning for	2018
Elmachtoub & …		2017
Kool et al.	Attention, Learn to Solve Routing Problems!	2018
Li et al.	…ombinatorial Optimization with Graph Convolutional Networks and Guided Tree Search	2018
Dai et al.	Learning Combinatorial Optimizati… Algorith…	2017
Bello et al.	Neural Com…	2016
…wly et al.	Genera… pr…	
	In…	
	H…	
…t al.	Opti…	2019
…al.	Learning MILP … …ime-Limit	2018
	Learning …teer nonlinear interior-point methods	2019
	L…rning when to use a decomposition	2018
	Machine Learning for Combinatorial Optimization: a Methodological Tour d'Horizon	2018
	Adaptive Large Neighborhood Search for Mixed Integer Programming	2018
…al.	Learning a Classification of Mixed-Integer Quadratic Programming Problems	2017
…ter	OptNet: Differentiable Optimization as a Layer in Neural Networks	2017
…ann &	Global Deterministic Optimization with Artificial Neural Networks Embedded	2018
	Large Scale Learning To Rank	2020
…g et al.	A General Large Neighborhood Search Framework for Solving Integer Programs	2020

Complex decisions, no easy answers

Sophisticated rules already in place

We don‘t even know good features



Huge Activity in Machine Learning for Mathematical Optimization

Ding et al.	Accelerating Primal Solution Findings for Mixed Integer Programs Based	2019
Bertsimas & S		2019
Fisch	optimal	2018
Misra e		2018
Bertsim		2018
Tang et al.	Learning to Cut	2019
Baltean-Lugojan et al.	Selecting cutting tic semidefinite outer-approximation via trained neural networks	2018
Etheve et al.	Reinforcement Learning for Varia Selection in a Branch and Bound Algorithm	2020
Zarpellon et al.	Parameterizing Branch-and-Bound Searc Policies	
Yang et al.	Learning Generalized Strong Bran 0-1 Knapsack Problems	
Song et al.	Learning to Search via Retro	
Gasse et al.	Exact Combinatorial Optim Networks	
Lee et al.	Learning to Branch: Acc Networks	
Hansknecht et al.	Cuts, Primal Heuristics Traveling Salesman P	
Balcan et al.	Learning to branch	
Václavík et al.	Accelerating the branc	
Hottung et al.	Deep Learning Assiste marshalling Problem	
Lodi & Zarpellon	On learning and branch	
Alvarez et al.	A Machine Learning-Bas	
Alvarez et al.	Online Learning for Stron	
Khalil et al.	Learning to branch in mixe	
Khalil	Machine Learning for Intege	
He et al.	Learning to Search in Branch a	
Alvarez et al.	A Supervised Machine Learning Branch-And-Bound	
Di Liberto et al.	Dynamic Approach for Switching Heur	13
Sabharwal et al.	Guiding Combinatorial Optimization with UC	2012

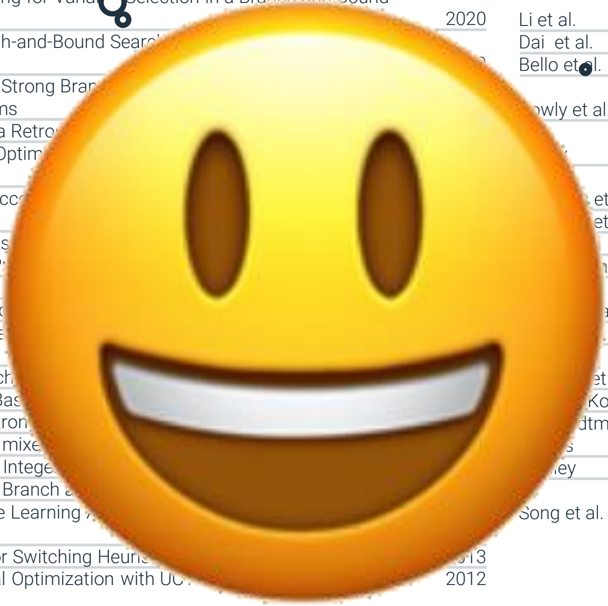
Khalil et al.	Learning to Run Heuristics in Tree Search	2017
Hutter et al.	Algorithm Runtime Prediction: Methods & Evaluation	2012
Hutter et al.	Automated Configuration of Mixed Integer Programming Solvers	2010
Ferber et al.	MIPaaL: Mixed Integer Program as a Layer	2019
Wilder et al.	raphs	2019
	ing using a	2019
	sed Learning for	2018
Elmachtou		2017
Kool et al.	Attention, Learn to Solve Routing Problems!	2018
Li et al.	Combinatorial Optimization with Graph Convolutional Networks and Guided Tree Search	2018
Dai et al.	Learning Combinatorial Optim ms over Graphs	2017
Bello et al.	Neural C ement Learning	2016
owly et al.	Gene prop ces with controllable	2017
	ough new test	2019
et al.	mbinatorial Problem	2019
et al.	L aching Time-Limit	2018
	hods	2019
l.	ning w tion	2018
al.	Machine Learning for Combinatorial Optimization: a Methodological Tour d'Horizon	2018
	Adaptive Large Neighborhood Search for Mixed Integer Programming	2018
et al.	Learning a Classification of Mixed-Integer Quadratic Programming Problems	2017
Kolter	OptNet: Differentiable Optimi	2017
dtmann &	G Dete Embedde	2018
ey	Larg	2020
Song et al.	A Prog	2020

Scaling!!!

Either-or decision

We know meaningful features

Currently no rule in place



Numerical Stability

Numerical Stability

- Numerical stability is a crucial topic in many applications
  - Recent blog series on Numerics, visit https://community.fico.com/
    - Numerics I: Solid Like a Rock or Fragile Like a Flower?
    - Numerics II: Zoom Into the Unknown
    - Numerics III: Learning to Scale
    - tbc...
- Real-life applications often complex and numerically challenging to handle:
  - More than half of client problems seen in the past year had some mild numeric issues.
  - After performance, numeric failures are the most common support request.
    - Unexpected solution status
    - Inconsistent results
    - Performance issues (e.g. simplex cycling)

Information on numeric stability

- Since Xpress 8.6, we provide numeric analysis tools
- A priori: distribution of matrix, objective, rhs coefficients

Coefficient range		original	solved
Coefficients	[min,max] :	[2.00e-06, 2.34e+02]	/ [1.25e-01, 1.67e+00]
RHS and bounds	[min,max] :	[1.67e-01, 9.23e+03]	/ [1.67e-01, 8.21e+02]
Objective	[min,max] :	[2.00e-06, 2.34e+02]	/ [2.00e-06, 2.34e+02]

comprises effects of presolving AND scaling

- A posteriori: report on numerical failures that the solver encountered

```
Numerical issues encountered:
   Dual failures     :      78 out of      2194 (ratio: 0.0356)
   Primal failures   :       5 out of       247 (ratio: 0.0202)
   Singular bases    :       5 out of     11180 (ratio: 0.0004)
   Nodes w/LP fails  :       9 out of        70 (ratio: 0.1286)
```

Condition Number

- The condition number κ of a matrix A provides a bound on how much a small change in b can affect x.
- For a square, invertible matrix B
$$\kappa = \|B\| \cdot \|B^{-1}\|$$
- One purpose of scaling is to reduce the condition number.
- Sampling the condition number is an optional feature (MIPKAPPAFREQ=1)

```
Nodes kappa stable     :       3757 (ratio: 0.0051)
Nodes kappa suspicious :       8476 (ratio: 0.0115)
Nodes kappa unstable   :     723171 (ratio: 0.9831)
Nodes kappa ill-posed  :        193 (ratio: 0.0003)
Largest kappa seen     : 4.959805e+14
Kappa attention level  : 0.2953
```

- Summarized in a single attention level from 0.0 (all stable) to 1.0 (anything goes).

Condition Number

- The condition number κ of a matrix A provides a bound on how much a small change in b can affect x.
- For a square, invertible matrix B

$$\kappa = \|B\| \cdot \|B^{-1}\|$$

- One purpose of scaling is to reduce the condition number.
- Sampling the condition number is an optional feature.

```
Nodes kappa stable      :     3757 (ratio: 0.0051)
Nodes kappa suspicious  :     8476 (ratio: 0.0115)
Nodes kappa unstable    :   723171 (ratio: 0.9831)
Nodes kappa ill-posed   :      193 (ratio: 0.0003)
Largest kappa seen      : 4.959805e+14
Kappa attention level   : 0.2953
```

- Summarized in a single attention level from 0.0 (all stable) to 1.0 (anything goes).

Numerical information raises awareness!

Adapt models or use non-default controls to prevent issues

Scaling

What is Scaling?

- Scaling is a widely used preconditioning technique, used by various kinds of algorithms
  - to reduce the condition number of the constraint matrix
    - to reduce error propagation
  - to improve the numerical behavior of the algorithms
  - to reduce the number of iterations required to solve the problem
- More precisely, LP scaling refers to the (iterative) multiplication of rows and columns by scalars
  - to reduce the absolute magnitude of nonzero coefficients in matrix, rhs and objective
  - to reduce the relative difference of nonzero coefficients in matrix, rhs and objective

Scaling in Linear Programming

- Basic Linear Program (LP):

$$\begin{array}{ll} max & cx \\ s.t. & Ax \leq b \end{array}$$

- Scaling multiplies rows and columns to bring coefficients “on one scale”.
  - Typically, close to 1

- Two scaling methods:
  - Standard: Divide rows by largest coefficient and then divide columns by largest coefficient. Repeat.
  - Curtis-Reid: Minimize least-squares deviation from 1 (logarithmically).

$$\begin{array}{ll} max & (cD^C)(D^{C^{-1}}x) \\ s.t. & (D^R A D^C)(D^{C^{-1}}x) \leq D^R b \end{array}$$

$$\Downarrow$$

$$c' = cD^C, A' = D^R A D^C, b' = D^R b$$
$$x' = D^{C^{-1}}x$$

$$\Downarrow$$

$$\begin{array}{ll} max & c'x' \\ s.t. & A'x' \leq b' \end{array}$$

Example

- We want to set up our home business to make boxes or chess pieces
  - We want to maximize profit [$5/box, $10/chess piece]
  - We have a limited amount of wood [100]
  - We have to buy tools [$30 for boxes, $500 for chess sets]
- A mixed integer programming (MIP) problem:

$$
\begin{array}{lrcl}
\max & 5x^{box} + 10x^{chess} - 30b^{box} - 500b^{chess} & & \\
s.t. & x^{box} + x^{chess} & \leq & 100 \\
 & x^{box} & \leq & 100b^{box} \\
 & x^{chess} & \leq & 100b^{chess} \\
 & b^{box}, b^{chess} & \in & \{0,1\}
\end{array}
$$

- Coefficient matrix:

  … with a potential basis matrix

$$
\begin{bmatrix}
1 & 1 & & \\
1 & & -100 & \\
 & 1 & & -100
\end{bmatrix}
$$

Example - Continued

- Unscaled:

$$
\begin{aligned}
x^{box} + x^{chess} &\le 100 \\
x^{box} \phantom{+ x^{chess}} &\le 100 b^{box} \\
x^{chess} &\le 100 b^{chess}
\end{aligned}
\qquad
\begin{bmatrix}
-\frac{1}{100} & -\frac{1}{100} & \frac{1}{100} \\
 & -1 & 1 \\
 & 1 &
\end{bmatrix}
\qquad
\kappa \approx 245
$$

Example - Continued

- Unscaled:

$$\begin{aligned} x^{box} + x^{chess} &\le 100 \\ x^{box} \qquad\qquad &\le 100 b^{box} \\ x^{chess} &\le 100 b^{chess} \end{aligned}$$

$$\begin{bmatrix} -\frac{1}{100} & -\frac{1}{100} & \frac{1}{100} \\ & -1 & 1 \\ & 1 & \end{bmatrix}$$

$$\kappa \approx 245$$

- Standard scaling:

$$\begin{aligned} x^{box} + x^{chess} &\le 100 \\ \frac{1}{100} x^{box} \qquad\qquad &\le b^{box} \\ \frac{1}{100} x^{chess} &\le b^{chess} \end{aligned}$$

$$\begin{bmatrix} -1 & -1 & \frac{1}{100} \\ & -100 & 1 \\ & 100 & \end{bmatrix}$$

$$\kappa \approx 245$$

Same!

Example - Continued

- Unscaled:

$$\begin{aligned} x^{box} + x^{chess} &\le 100 \\ x^{box} \qquad\qquad &\le 100 b^{box} \\ x^{chess} &\le 100 b^{chess} \end{aligned}$$

$$\begin{bmatrix} -\frac{1}{100} & -\frac{1}{100} & \frac{1}{100} \\ & -1 & 1 \\ & 1 & \end{bmatrix}$$

$\kappa \approx 245$

- Standard scaling:

$$\begin{aligned} x^{box} + x^{chess} &\le 100 \\ \frac{1}{100} x^{box} \qquad\qquad &\le b^{box} \\ \frac{1}{100} x^{chess} &\le b^{chess} \end{aligned}$$

$$\begin{bmatrix} -1 & -1 & \frac{1}{100} \\ & -100 & 1 \\ & 100 & \end{bmatrix}$$

$\kappa \approx 245$

Same!

- "Best" scaling
  x=fraction of all material to use

$$\begin{aligned} x^{box} + x^{chess} &\le 1 \\ x^{box} \qquad\qquad &\le b^{box} \\ x^{chess} &\le b^{chess} \end{aligned}$$

$$\begin{bmatrix} -1 & -1 & 1 \\ & -1 & 1 \\ & 1 & \end{bmatrix}$$

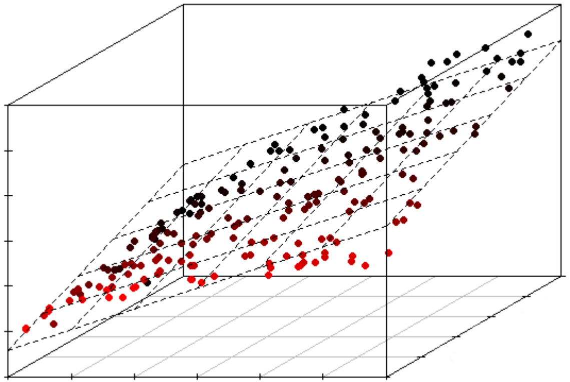
$\kappa \approx 25$

Best!

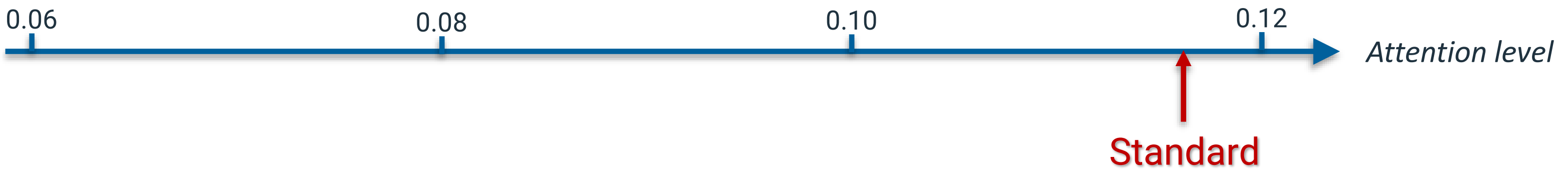
Learning To Scale

Learning to Scale

- New approach: Learn to Scale
  - Try each scaling method: Standard and Curtis-Reid.
  - One fixed method not always best.
  - Use an ML model based on linear regression to predict which method will result in the smallest attention level.
  - Features drawn from coefficient distributions.

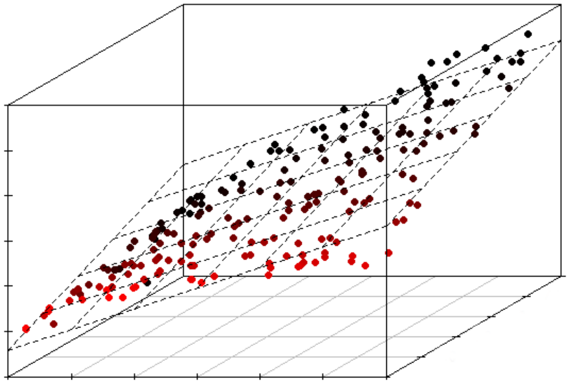


- Trained on more than 1000 MIP instances
- Validation outcome:

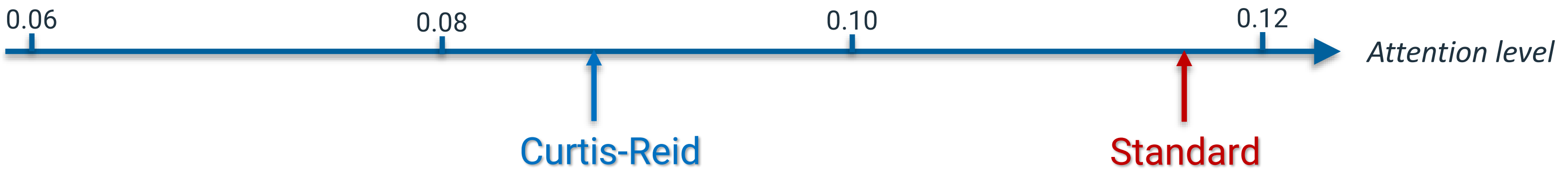


Learning to Scale

- New approach: Learn to Scale
  - Try each scaling method: Standard and Curtis-Reid.
  - One fixed method not always best.
  - Use an ML model based on linear regression to predict which method will result in the smallest attention level.
  - Features drawn from coefficient distributions.

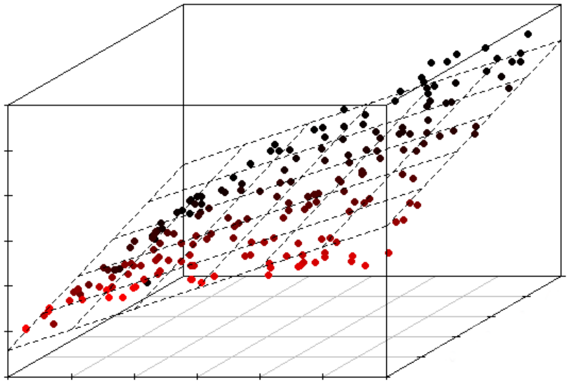


- Trained on more than 1000 MIP instances.
- Validation outcome:

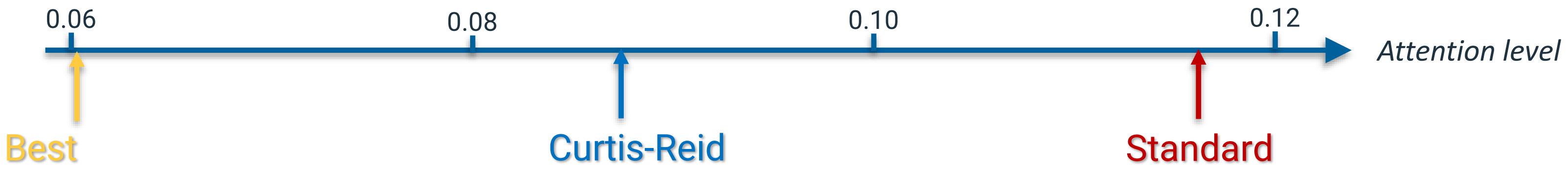


Learning to Scale

- New approach: Learn to Scale
  - Try each scaling method: Standard and Curtis-Reid.
  - One fixed method not always best.
  - Use an ML model based on linear regression to predict which method will result in the smallest attention level.
  - Features drawn from coefficient distributions.

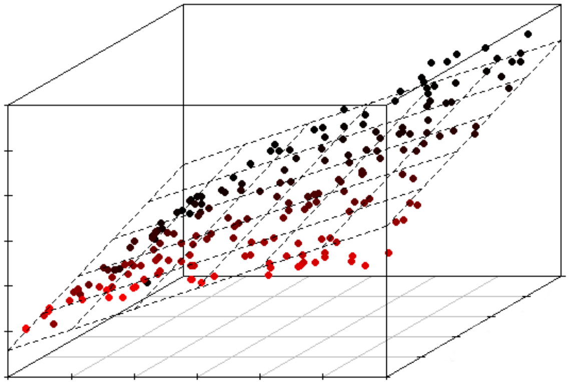


- Trained on more than 1000 numerically challenging instances.
- Validation outcome:

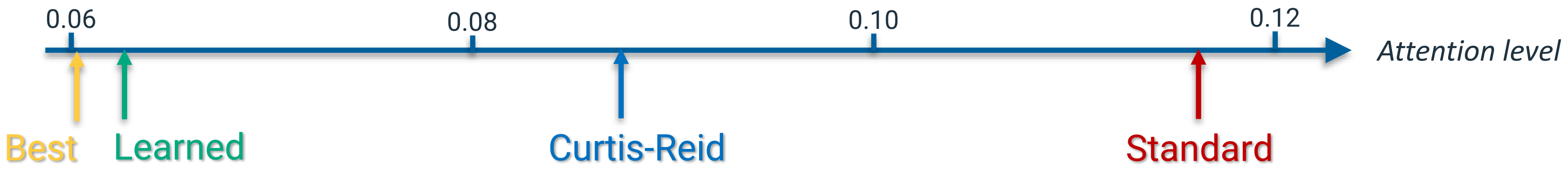


Learning to Scale

- New approach: Learn to Scale
  - Try each scaling method: Standard and Curtis-Reid.
  - One fixed method not always best.
  - Use an ML model based on linear regression to predict which method will result in the smallest attention level.
  - Features drawn from coefficient distributions.



- Trained on more than 1000 numerically challenging instances.
- Validation outcome:



Computational Results

- On our set of Numerically Challenging instances:
  - Tremendous improvements in all stability criteria

Dual Fails	-64%	Primal Fails	-67%	Singular Inverts	-48%
Infeasibilities	-26%	Inconsistencies	-35%	Violated Sols	-12%
Kappa Stable	+148%	Kappa Max	-979%	**Attn. Level**	**-88%**

- ≈10% performance improvements on our simplex test sets.
- On our MIP Performance set: performance-neutral
- New control: `AUTOSCALE`
  - Setting `SCALING` control will override `AUTOSCALE`.

Conclusion

Machine Learning
for general MIP

Works best:

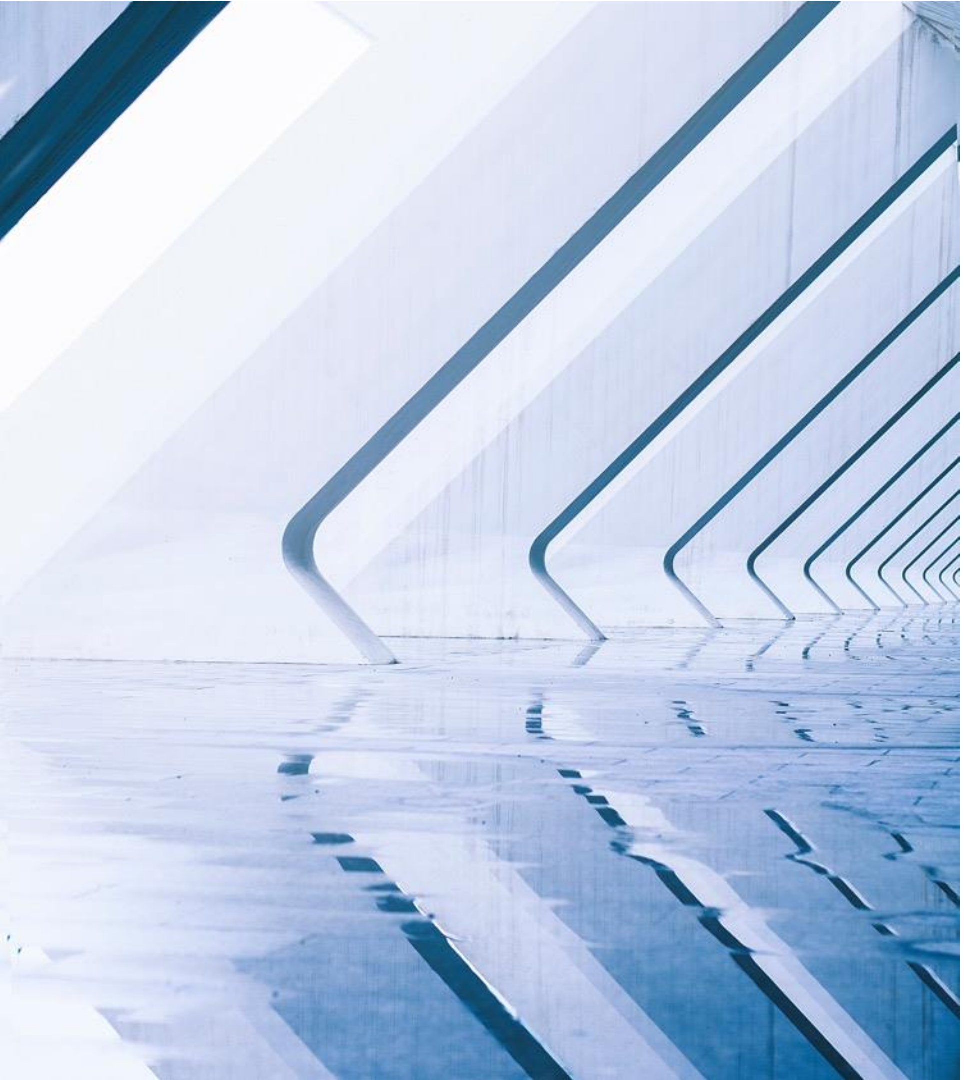
- For categorical decisions,
- With suitable features,
- With established label that connects to the features,
- When you are not competing against a rule that has been finetuned over decades.

Learning to scale:

- ML module to predict scaling method for MIP and LP solving
- Drastically improves numerical stability
- Does not deteriorate performance
- One of many recent components in Xpress that address numeric stability

Sneak peek: Learning the Attention Level

- A-priori prediction: Will the current solve lead to a high attention level?
  - Called after initial LP relaxation
  - Prints a warning for the user
- Similar features as in “Learning to scale”
  - Additionally use conditioning of matrix w.r.t. right-hand side
- Uses random forest
  - Accuracy > 95%
  - False negative rate < 2%
  - Threshold biased towards false positives
- To be released with the next major Xpress version



Thank You!

Timo Berthold