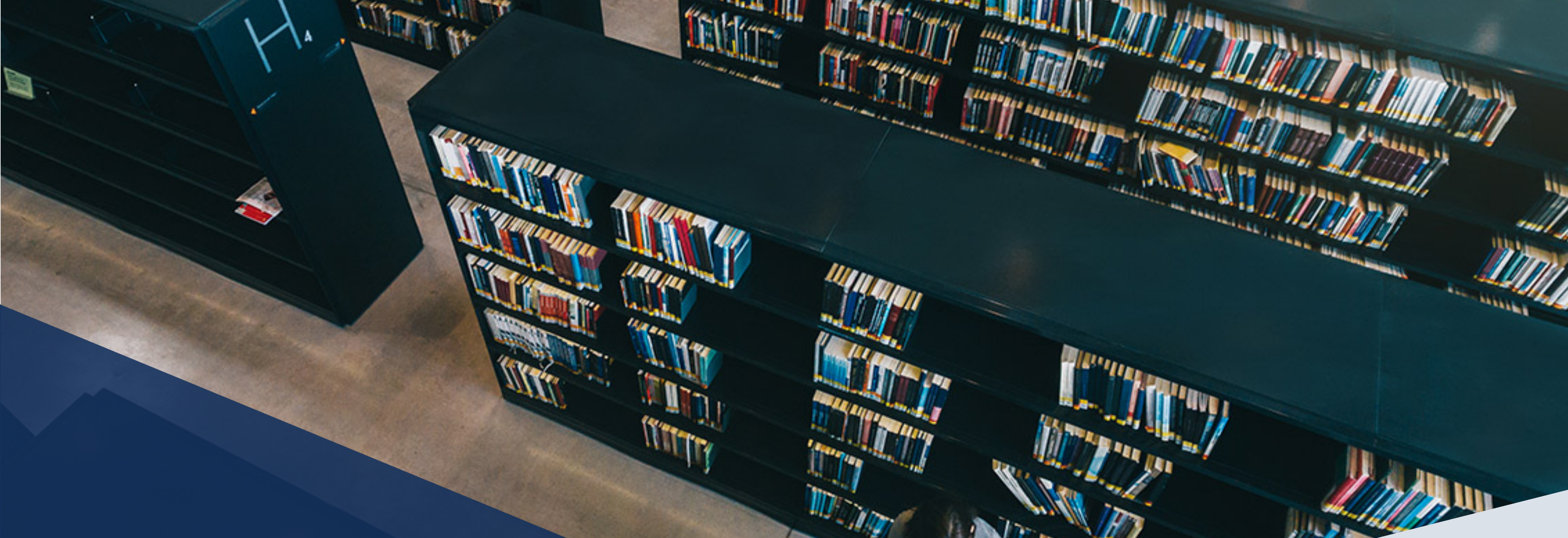

Linear Programming

A Historical View

Robert E. Bixby
bixby@gurobi.com

Sept 2022

GUROBI OPTIMIZATION

Agenda

01

The "Early" History
1947-1990

02

New Developments
Three Examples

03

LP Today

LP History

References

- “Programming in a linear structure,” George B. Dantzig, published August 1948 by U.S. Air Force Comptroller
- “History of the Development of LP Solvers,” William Orchard-Hays, *Interfaces* Vol. 20, 4 July-August 1990 (pp. 61-73)
- “Solving Real-World Linear Programs: A Decade and more of progress,” Robert E. Bixby, *Operations Research* Vol. 50, No. 1, January-February 2002, pp. 3-15.
- “A Brief History of Linear and Mixed-Integer Programming Computation,” Robert E. Bixby, *Optimization Stories*, Ed. M. Grötschel, Documenta Mathematica, 2012, pp. 107-121

The Early History

George Dantzig, 1947

- Introduced LP and recognized it as more than a conceptual tool: Computing answers important.
- Invented “primal” simplex algorithm.
- First LP solved: Laderman, 9 cons., 77 vars., 120 PERSON-DAYS.

First computer code – 1951

- National Bureau of Standards (now NIST)
- SEAC computer
- 18 hours and 73 simplex iterations to solve an instance with 48 equations and 71 variables.

Orchard-Hays

- 1952-54 “Card Programmable Calculator” implementation
- First implementation used explicit inverse – not encouraging. The introduced product form update.
- 8 hours to solve an instance with 26 equations and 71 variables.

"A certain wide class of practical problems appears to be just beyond the range of modern computing machinery. These problems occur in everyday life; they run the gamut from some very simple situations that confront an individual to those connected with the national economy as a whole. Typically, these problems involve a complex of different activities in which one wishes to know which activities to emphasize in order to carry out desired objectives under known limitations."

George B. Dantzig, 1948

The Early History
Continued

1954-55: IBM 701, 100-200 rows

1956: IBM 704, 4 K “core”, RSLP1, 255 rows

- Used by large oil companies

1962-66: 7090/94, LP/90/94, 1024 rows

1966-70: IBM 360, MPS/360 & MPSX/370

- First real LP systems

1971-73: MPS III/Whizard, 32000 rows

- Supersparsity: Kalan 1971
- P4, Pre-assigned pivot factorization: Hellerman & Rarick 1971
- Presolve came later with Whizard (~1980)

The Decade of the 70’s

Interest in optimization flowered

Numerous new applications identified

- Large scale planning applications particularly popular
- Funding widely available

Significant difficulties emerged

Building application was very expensive and very risky

- 3-4 year development cycles
- Developers and application owners had to be multi-faceted experts: Computer, Data, Algorithm, and Modeling skills necessary.
  - “Deploying an application was virtually impossible”
- Technology just wasn’t ready: LP were hard and MIP was a disaster

Result: *Disillusionment and much of that disillusionment persists to this day.*

The Decade of the 80’s

Mid 80’s:

There was perception was that LP software had progressed about as far as it could go – MPSX/370 and MPSIII

BUT LP was definitely **not a solved problem** … example: “Unsolvable” airline LP model with 4420 constraints, 6711 variables

There were several key developments

IBM PC introduced in 1981

Relational databases developed:

- Separation of logical and physical allocation of data.
- ERP systems introduced.

LP solvable in polynomial time

- Khachian’s 1979 paper on the “ellipsoid method”
- Karmarkar’s 1984 paper on interior-point methods

The Genesis of CPLEX: 1983–1988

1972-1985: Very theoretical research in OR (“matroid theory”)

1983: Began developing classroom LP code (IBM had introduced PCs)

1985-1987
- Invited by Tom Baker to provide LP code for Chesapeake Decision Sciences MIMI product
- First actual sale of that code was to AMOCO is 1987

1987-1988
- Could not get funding for research in LP computation -- zero success

1988: Founded CPLEX optimization
- Cofounder: Janet Lowe

End of 80's: Still not out of the woods

Grötschel and Holland, "Solution of large-scale symmetric travelling salesman problems," 1991, Mathematical Programming

LP codes were still far from satisfactory:

- "Some of the linear programs that arose were hard to solve, even for highly praised commercial LP-codes like IBM's MPSX"

What were they referring to?

- LPs that had ~1500 rows and ~2500 variables and took almost 3 hours to solve, if they solved
- Degeneracy, slow convergence, and system inflexibility remained major obstacles in commercial software.

1st Public CPLEX Results –1986

TEST RUNS*

NETLIB PROBLEM	# Rows	# Cols	# ≠0	LOPT # ITER	LOPT Time	XMP # ITER	XMP Time
GFRD-PNC	616	1092	2378	628	80.5	983	144.1
SCRS8	490	1169	3183	735	112.2	1271	182.7
SIERRA	1227	2036	7303	493	93.8	950	270.0
STANDAT	359	1183	3032	181	14.7	75	8.1
SCAGR25	471	500	1555	517	69.1	1470	248.3
SHARE2B	97	79	695	93	4.7	138	6.5
SHARE1B	118	225	1152	217	16.1	411	28.8
E226	224	282	2579	444	50.8	655	70.2
CAPRI	272	353	1768	364	32.4	550	44.5
BANDM	306	472	2495	362	59.5	1679	250.9
STAIR	357	467	3847	528	260.4	1667	530.9
ETAMACRO	401	688	2410	997	105.3	1140	144.
SHIP12L	1152	5427	16171	1283	242.5	1510	515.7

* Run on SUN 3/50 workstation. Compiled with -O and -f68881 options set.

The Decade of the 90's

LP performance takes off

Primal-dual log-barrier algorithms completely reset the bar

Simplex algorithms unexpectedly kept pace

Data became plentiful and accessible

ERP systems became commonplace

Popular new applications begin to show that integer programming could work on difficult, real-world problems

Airlines, Supply-Chain

Agenda

01
The “Early” History
1947-1990

02
New Developments
Three Examples

03
LP Today

AA Challenge, 1989

Example 1

Bob Bixby
Irv Lustig
Roy Marsten
Dave Shanno
John Gregory

AA made a huge crew-scheduling model available

837 rows, 12,753, 313 columns

The Challenge: Solve the LP

Sheer size of model dictated a special-purpose algorithm

Developed algorithm called **Sifting**, an idea of John Forrest

Step 1: Select a subset of columns and solve

Step 2: Use optimal duals to price out the remaining columns

- No violation implies optimal, stop.
- Otherwise select a "good set" of violated columns

Step 3: Augment column subset and return to Step 1.

Two observations

Barrier (interior-point) much faster solving from scratch

Advanced basis helps near the end. Primal feasibility → use primal.

AA Challenge, 1989

Continued

Implemented on Cray Y-MP

Considerable experimentation

Solved in 10 applications of Step 1

- 5 barrier steps at start
- 5 primal steps to complete computation
    - Total computation time: **247 seconds**
    - Optimization 204, pricing 43

What did we learn

Developed first really fast crossover (based on a result of Megiddo)

Generated a great deal of enthusiasm for solving larger, harder models.

Table X

Iteration Log for Problem 13mil With Hybrid Scheme

Major Iteration	Column in Problem	Method	Iterations	Objective Value	Optimize Time	Price Time	Columns Added	Duplicates Deleted
1	1,688	**OB1**	9	57448.00	8.549	7.262	25,449	14,638
2	12,499	**OB1**	14	57448.00	15.324	6.041	25,857	12,135
3	26,221	**OB1**	24	57448.00	28.582	6.493	25,857	12,774
4	39,304	**OB1**	33	50743.910	44.399	5.750	13,018	7,239
5	45,083	**OB1**	31	48530.461	43.398	0.000	0	0
*	45,083	Crossover	327	48530.461	30.538			
6	5,217	**CPLEX**	0	48530.461	3.105	3.555	1,832	1,000
7	6,049	**CPLEX**	297	48510.117	9.572	3.532	1,798	1,022
8	6,825	**CPLEX**	648	48418.199	13.754	3.571	382	238
9	6,969	**CPLEX**	316	48400.129	6.244	3.259	18	7
10	6,980	**CPLEX**	7	48400.129	0.972	3.145	0	0

Sifting

Today

Instance:	837 rows
	1,322,797 columns

Primal	**93.1** seconds
Dual	**416.0** seconds
Barrier	**25.9** seconds
Sifting	**11.9** seconds

AA & US Air Merger?

Example 2

As reported by Ed Rothberg,
~early to mid 1990s

CPLEX having trouble breaking into airlines

OSL had a firm hold

SGI and Ed Rothberg entered the picture

CPLEX wanted a better barrier solver

SGI wanted to sell computers

SGI-CPLEX collaboration produced a new barrier solver

Used latest SGI hardware: 3-4x

Parallelized code, 8 processors: 4-5x

Improved Cholesky Ordering algorithm

- Nested Dissection versus Approximate Min Degree: 3-4x
- Net effect: SGI-CPLEX barrier was about 40x OSL

Not enough: 40x was a nice to have not a must have!

AA & US Air Merger

Continued

Then: A critical business event occurred

AA & US Air were considering a merger

Due diligence ➔ needed to evaluate combined fleet model and do it quickly

- OSL took 3-4 hours, SGI-CPLEX took 5 minutes!

A nice to have became a must have.

- Result: A key business breakthrough for CPLEX.

What did we learn?

Parallel barrier became the standard

We learned the “must-have” story

LAU2 (Late 1980s)

Example 3

LAU2

Fleet-assignment model
4420 rows, 6711 Columns, 101377 Nonzeros

Challenge presented by UA

If you “solve” this problem, we will buy computers – John Gregory, Cray Research

GUROBI OPTIMIZATION

LAU2 (Late 1980s)

Continued

Attempts to solve LAU2 with CPLEX

Step 1: 1987 – Ran CPLEX 1.0 for 7 hours on Cray X-MP. Stuck in Phase I

Step 2: Introduced degeneracy handling

- Problem solvable but still slow

Step 3: John Gregory reported solving with OB1 in 1200 seconds.

- Recommended giving up on simplex!!

Step 4: An idea

- Maybe the dual isn't nearly as degenerate.
- Took explicit dual and solved with primal. Closer to barrier.
- Implemented dual simplex, with standard pricing. Closer still.

Step 5: Another idea

- John Forrest had success using primal Devex.
- Conclusion: Implemented dual steepest edge.
- Dual simplex was now winner

Step 6: Don Goldfarb (steepest-edge guru) suggests simpler SE

- Implemented new dual steepest edge
- Dual simplex was the clear winner

LAU2 (Late 1980s)

Continued

CPLEX 3.0 on HP9000/730

Primal - Partial pricing	27765 seconds
Barrier	1400 seconds
Dual – Dantzig pricing	2440 seconds
Dual – Steepest-edge pricing	273 seconds

What did we learn?

This led to the development of dual steepest edge, today's LP algorithm of choice

Agenda

01
The "Early" History
1947-1990

02
New Developments
Three Examples

03
LP Today

Example

A Production Planning Model

401,640 constraints
1,584,000 variables

Solution timeline (2.0 GHz Pentium 4):

Test: Went back to 1st CPLEX (1988)

1988 (CPLEX 1.0): 15.0 days (Dagstuhl, 28 Nov)

Example

A Production Planning Model

401,640 constraints
1,584,000 variables

Solution timeline (2.0 GHz Pentium 4):

Test: Went back to 1st CPLEX (1988)

1988 (CPLEX 1.0): 19.0 days (Amsterdam, 2 Dec)

GUROBI
OPTIMIZATION

Example

A Production Planning Model

401,640 constraints
1,584,000 variables

Solution timeline (2.0 GHz Pentium 4):

Test: Went back to 1st CPLEX (1988)

1988 (CPLEX 1.0): 23.0 days (Houston, 6 Dec)

Example

A Production Planning Model

401,640 constraints
1,584,000 variables

Solution timeline (2.0 GHz Pentium 4):

Test: Went back to 1st CPLEX (1988)		**Speedup**
1988 (CPLEX 1.0):	29.8 days	**1x**
1997 (CPLEX 5.0):	1.5 hours	**480x**
2003 (CPLEX 9.0):	59.1 seconds	**43500x**

Progress in LP

1988-2004

Operations Research, Jan 2002, pp. 3—15, updated in 2004

Algorithms (machine independent): **3,300x**

Primal versus best of Primal/Dual/Barrier

Machines (workstations → PCs): **1,600x**

NET: Algorithm × Machine **5,300,000x**

(2 months/5300000 ~= 1 second)

What progress has occurred since 2004?

No significant algorithmic improvements between 2004 and ~2015.

In the last 7 years that has changed: on “large” models, LP simplex algorithms have improved by approximately 3x.

LP Today

Practitioners consider LP a solved problem

Large models can now be solved robustly and quickly

- Regularly solve models with millions of variables and constraints

Thank You

Robert E. Bixby
bixby@gurobi.com
September 2022