Solving Linear and Integer Programs

Robert E. Bixby
ILOG, Inc. and Rice University

Ed Rothberg
ILOG, Inc.

Outline

- **Linear Programming:** *Bob Bixby*
  - Example and introduction to basic LP, including duality
  - Primal and dual simplex algorithms
  - Computational progress in linear programming
  - Implementing the dual simplex algorithm
- **Mixed-Integer Programming:** *Ed Rothberg*

An Example

Diet Problem*

Bob wants to plan a nutritious diet, but he is on a limited budget, so he wants to spend as little money as possible. His nutritional requirements are as follows:

1. 2000 kcal
2. 55 g protein
3. 800 mg calcium

* From Linear Programming, by Vašek Chvátal

Diet Problem

Nutritional values

Bob is considering the following foods:

Food	Serving Size	Energy (kcal)	Protein (g)	Calcium (mg)	Price per serving
Oatmeal	28 g	110	4	2	$0.30
Chicken	100 g	205	32	12	$2.40
Eggs	2 large	160	13	54	$1.30
Whole milk	237 cc	160	8	285	$0.90
Cherry pie	170 g	420	4	22	$0.20
Pork and beans	260 g	260	14	80	$1.90

Diet Problem

Variables

We can represent the number of servings of each type of food in the diet by the variables:

x_1 servings of oatmeal

x_2 servings of chicken

x_3 servings of eggs

x_4 servings of milk

x_5 servings of cherry pie

x_6 servings of pork and beans

Diet Problem

Nutritional values

Bob is considering the following foods:

Food	Serving Size	Energy (kcal)	Protein (g)	Calcium (mg)	Price per serving	
Oatmeal	28 g	110	4	2	$0.30	x_1
Chicken	100 g	205	32	12	$2.40	x_2
Eggs	2 large	160	13	54	$1.30	x_3
Whole milk	237 cc	160	8	285	$0.90	x_4
Cherry pie	170 g	420	4	22	$2.00	x_5
Pork and beans	260 g	260	14	80	$1.90	x_6

KCAL constraint:

$$110x_1 + 205x_2 + 160x_3 + 160x_4 + 420x_5 + 260x_6 \geq 2000$$

$(110x_1 = \text{kcals in oatmeal})$

Diet Problem

LP formulation

Minimize **Cost**

$$0.3x_1 + 2.40x_2 + 1.30x_3 + 0.90x_4 + 2.0x_5 + 1.9x_6$$

subject to: **Nutritional requirements**

$$\begin{aligned}
110x_1 + 205x_2 + 160x_3 + 160x_4 + 420x_5 + 260x_6 &\geq 2000 \\
4x_1 + 32x_2 + 13x_3 + 8x_4 + 4x_5 + 14x_6 &\geq 55 \\
2x_1 + 12x_2 + 54x_3 + 285x_4 + 22x_5 + 80x_6 &\geq 800
\end{aligned}$$

Bounds

$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$$

Diet Problem

Solution

When we solve the preceding LP (using CPLEX, of course) we get a solution value of **$6.71**, which is achieved with the following menu:

14.24 servings of oatmeal

0 servings of chicken

0 servings of eggs

2.71 servings of milk

0 servings of cherry pie

0 servings of pork and beans

Some Basic Theory

Linear Program – Definition

A **linear program** (**LP**) in **standard form** is an optimization problem of the form

$$\begin{aligned} &\textit{Minimize} && c^T x \\ &\textit{Subject to} && Ax = b \qquad \text{(P)} \\ & && x \geq 0 \end{aligned}$$

Where $c \in \boldsymbol{R}^n$, $b \in \boldsymbol{R}^m$, $A \in \boldsymbol{R}^{m \times n}$, and x is a vector of n **variables**. $c^T x$ is known as the **objective function**, $Ax=b$ as the **constraints**, and $x \geq 0$ as the **nonnegativity** conditions. b is called the **right-hand side**.

Dual Linear Program – Definition

The **dual** (or **adjoint) linear program** corresponding to (P) is the optimization problem

$$\begin{aligned} &\textit{Maximize} && b^T \pi \\ &\textit{Subject to} && A^T \pi \leq c \qquad \text{(D)} \\ & && \pi \ \textit{free} \end{aligned}$$

In this context, (P) is referred to as the **primal linear program**.

Primal

$$\begin{aligned} &\textit{Minimize} && c^T x \\ &\textit{Subject to} && Ax = b \\ & && x \geq 0 \end{aligned}$$

Weak Duality Theorem

(von Neumann 1947)

Let x be feasible for (P) and π feasible for (D). Then

$$\boxed{\text{Maximize}} \quad b^T\pi \le c^T x \quad \boxed{\text{Minimize}}$$

If $b^T\pi = c^T x$, then x is optimal for (P) and π is optimal for (D); moreover, if either (P) or (D) is **unbounded**, then the other problem is **infeasible**.

Proof:

$$\pi^T b \underset{\boxed{Ax = b}}{=} \pi^T A x \underset{\boxed{\pi^T A \le c^T \ \& \ x \ge 0}}{\le} c^T x \quad \square$$

Solving Linear Programs

- **Three types of algorithms are available**
  - Primal simplex algorithms (Dantzig 1947)
  - Dual simplex algorithms (Lemke 1954)
    - Developed in context of game theory
  - Primal-dual log barrier algorithms
    - Interior-point algorithms (Karmarkar 1989)
    - Reference: Primal-Dual Interior Point Methods, S. Wright, 1997, SIAM

Primary focus: Dual simplex algorithms

Basic Solutions – Definition

Let B be an ordered set of m distinct indices $(B_1,\dots,B_m)$ taken from $\{1,\dots,n\}$. B is called a **basis** for (P) if A_B is nonsingular. The variables x_B are known as the **basic variables** and the variables x_N as the **non-basic** variables, where $N = \{1,\dots,n\}\backslash B$. The corresponding **basic solution** $X \in \mathbf{R}^n$ is given by $X_N=0$ and $X_B=A_B^{-1}\,b$. B is called **(primal) feasible** if $X_B \geq 0$.

Note: $AX = b \Rightarrow A_B X_B + A_N X_N = b \Rightarrow A_B X_B = b \Rightarrow X_B = A_B^{-1} b$

Primal Simplex Algorithm

(Dantzig, 1947)

Input: A feasible basis B and vectors

$$X_B = A_B^{-1}b \quad \text{and} \quad D_N = c_N - A_N^T B^{-T} c_B.$$

- **Step 1:** (Pricing) If $D_N \geq 0$, stop, B is optimal; else let
$$j = argmin\{D_k : k \in N\}.$$
- **Step 2:** (FTRAN) Solve $A_B y = A_j$.
- **Step 3:** (Ratio test) If $y \leq 0$, stop, (P) is unbounded; else, let
$$i = argmin\{X_{Bk}/y_k : y_k > 0\}.$$
- **Step 4:** (BTRAN) Solve $A_B^T z = e_i$.
- **Step 5:** (Update) Compute $\alpha_N = -A_N^T z$. Let $B_i = j$. Update X_B (using y) and D_N (using α_N)

Note: x_j is called the **entering** variable and x_{Bi} the **leaving** variable. The D_N values are known as **reduced costs** – like partial derivatives of the objective function relative to the nonbasic variables.

Dual Simple Algorithm – Setup

Simplex algorithms apply to problems with constraints in equality form. We convert (D) to this form by adding the dual **slacks** d:

$$\begin{aligned} &\text{Maximize} \quad b^T\pi \\ &\text{Subject to} \quad A^T\pi + d = c \\ &\qquad\qquad \pi \text{ free}, \; d \geq 0 \end{aligned} \quad \boxed{\Leftrightarrow A^T\pi \leq c}$$

17

Dual Simple Algorithm – Setup

Simplex algorithms apply to problems with constraints in equality form. We convert (D) to this form by adding the dual **slacks** d:

$$\begin{aligned} &\text{Maximize} \quad b^T\pi \\ &\text{Subject to} \quad A^T\pi + d = c \\ &\qquad\qquad \pi \text{ free}, \; d \geq 0 \end{aligned} \longleftarrow \begin{pmatrix} A_B^T & I_B & 0 \\ A_N^T & 0 & I_N \end{pmatrix} \begin{pmatrix} \pi \\ d_B \\ d_N \end{pmatrix} = \begin{pmatrix} c_B \\ c_N \end{pmatrix}$$

Given a basis B, the corresponding **dual basic solution** Π, D is determined as follows:

$$D_B = 0 \;\Rightarrow\; \Pi = A_B^{-T}c_B \;\Rightarrow\; D_N = c_N - A_N^T\Pi.$$

B is **dual feasible** if $D_N \geq 0$.

18

An Important Fact

If X and Π, D are the respective primal and dual basic solutions determined by a basis B, then

$$\Pi^T b = c^T X.$$

Hence, by weak duality, if B is both primal and dual feasible, then X is optimal for (P) and Π is optimal for (D).

Proof: $c^T X = c_B^T X_B$ (since $X_N=0$)
$= \Pi^T A_B X_B$ (since $\Pi = A_B^{-T} c_B$)
$= \Pi^T b$ (since $A_B X_B = b$) □

Dual Simplex Algorithm

(Lemke, 1954)

Input: A dual feasible basis B and vectors

$$X_B = A_B^{-1} b \quad \text{and} \quad D_N = c_N - A_N^T B^{-T} c_B.$$

- **Step 1:** (Pricing) If $X_B \geq 0$, stop, B is optimal; else let
  $$i = \mathrm{argmin}\{X_{Bk} : k \in \{1, \ldots, m\}\}.$$
- **Step 2:** (BTRAN) Solve $B^T z = e_i$. Compute $\alpha_N = -A_N^T z$.
- **Step 3:** (Ratio test) If $\alpha_N \leq 0$, stop, (D) is unbounded; else, let
  $$j = \mathrm{argmin}\{D_k/\alpha_k : \alpha_k > 0\}.$$
- **Step 4:** (FTRAN) Solve $A_B y = A_j$.
- **Step 5:** (Update) Set $B_i = j$. Update X_B (using y) and D_N (using α_N)

Note: d_{Bi} is the **entering** variable and d_j is the **leaving** variable. (Expressed in terms of the primal: x_{Bi} is the leaving variable and x_j is the entering variable)

Simplex Algorithms

Input: A primal feasible basis B and vectors

$X_B=A_B^{-1}b$ & $D_N=c_N-A_N^TA_B^{-T}c_B$.

- **Step 1:** (Pricing) If $D_N \geq 0$, stop, B is optimal; else, let

  $j = argmin\{D_k : k \in N\}$.
- **Step 2:** (FTRAN) Solve $A_By=A_j$.
- **Step 3:** (Ratio test) If $y \leq 0$, stop, (P) is unbounded; else, let

  $i = argmin\{X_{Bk}/y_k: y_k > 0\}$.
- **Step 4:** (BTRAN) Solve $A_B^Tz = e_i$.
- **Step 5:** (Update) Compute $\alpha_N = -A_N^Tz$. Let $B_i=j$. Update X_B (using y) and D_N (using α_N)

Input: A dual feasible basis B and vectors

$X_B=A_B^{-1}b$ & $D_N=c_N-A_N^TA_B^{-T}c_B$.

- **Step 1:** (Pricing) If $X_B \geq 0$, stop, B is optimal; else, let

  $i = argmin\{X_{Bk} : k \in \{1,\ldots,m\}\}$.
- **Step 2:** (BTRAN) Solve $A_B^Tz = e_i$. Compute $\alpha_N=-A_N^Tz$.
- **Step 3:** (Ratio test) If $\alpha_N \leq 0$, stop, (D) is unbounded; else, let

  $j = argmin\{D_k/\alpha_k: \alpha_k > 0\}$.
- **Step 4:** (FTRAN) Solve $A_By = A_j$.
- **Step 5:** (Update) Set $B_i=j$. Update X_B (using y) and D_N (using α_N)

Correctness: Dual Simplex Algorithm

- **Termination criteria**
  - Optimality **(DONE – by "An Important Fact" !!!)**
  - Unboundedness
- **Other issues**
  - Finding starting dual feasible basis, or showing that no feasible solution exists
  - Input conditions are preserved (i.e., that B is still a feasible basis)
  - Finiteness

Summary:
What we have done and what we have to do

- **Done**
  - Defined primal and dual linear programs
  - Proved the weak duality theorem
  - Introduced the concept of a basis
  - Stated primal and dual simplex algorithms
- **To do (for dual simplex algorithm)**
  - Show correctness
  - Describe key implementation ideas
  - Motivate

Dual Unboundedness
($\Rightarrow$ primal infeasible)

- We carry out a key calculation
- As noted earlier, in an iteration of the dual

d_{Bi} enters basis
d_j leaves basis
in

$$\begin{array}{ll} \textit{Maximize} & b^T\pi \\ \textit{Subject to} & A^T\pi + d = c \\ & \pi \text{ free}, d \geq 0 \end{array}$$

- **The idea:** Currently $\boldsymbol{d_{Bi} = 0}$, and $\boldsymbol{X_{Bi} < 0}$ has motivated us to increase $\boldsymbol{d_{Bi}}$ to $\boldsymbol{\theta > 0}$, leaving the other components of $\boldsymbol{d_B}$ at $\boldsymbol{0}$ (the object being to increase the objective). Letting $\underline{d}, \underline{\pi}$ be the corresponding dual solution as a function of θ, we obtain

$$\underline{\boldsymbol{d}}_{\boldsymbol{B}} = \theta \boldsymbol{e_i} \qquad \underline{\boldsymbol{d}}_N = \boldsymbol{D}_N - \theta\, \alpha_N \qquad \underline{\pi} = \pi - \theta z,$$

where α_N and z are as computed in the algorithm.

(Dual Unboundedness – cont.)

- Letting $\underline{d}, \underline{\pi}$ be the corresponding dual solution as a function of θ. Using α_N and z from dual algorithm,

$$\underline{d}_B = \theta e_i \quad \underline{d}_N = D_N - \theta \alpha_N \quad \underline{\pi} = \pi - \theta z.$$

- Using $\theta > 0$ and $X_{Bi} < 0$ yields

$$\begin{aligned} new\_objective = \underline{\pi}^T b &= (\pi - \theta z)^T b \\ &= \pi^T b - \theta X_{Bi} \\ &= old\_objective - \theta X_{Bi} > old\_objective \end{aligned}$$

- **Conclusion 1:** If $\alpha_N \leq 0$, then $\underline{d}_N \geq 0 \ \forall \ \theta > 0 \Rightarrow$ (D) is unbounded.
- **Conclusion 2:** If α_N **not**≤ 0, then

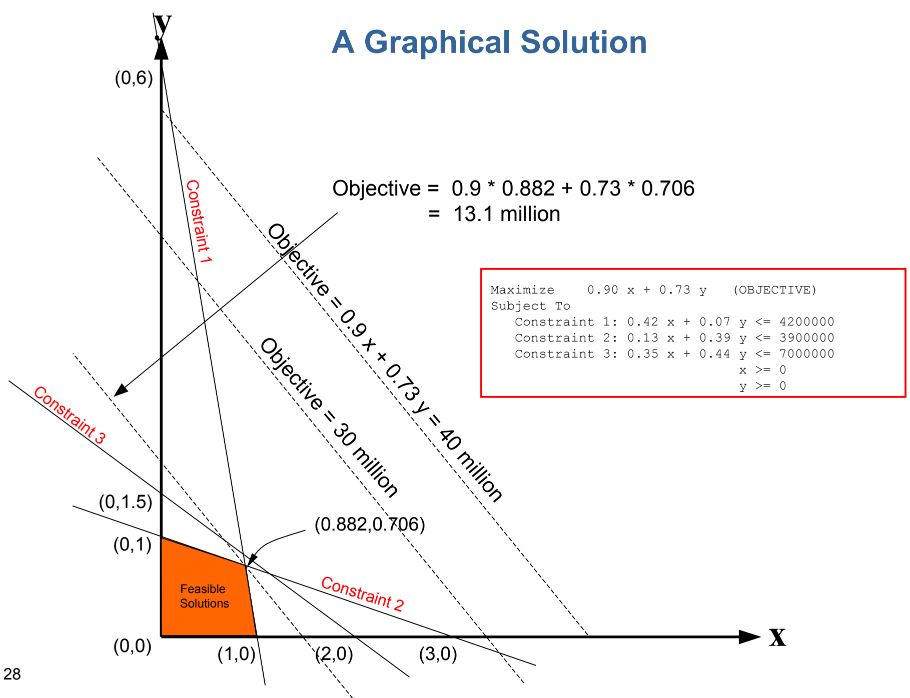
$$\underline{d}_N \geq 0 \Rightarrow \theta \leq D_j / \alpha_j \ \forall \ \alpha_j > 0$$
$$\Rightarrow \theta_{max} = min\{D_j / \alpha_j : \alpha_j > 0\}$$

(Dual Unboundedness – cont.)

- **Finiteness:** If $D_B > 0$ for all dual feasible bases B, then the dual simplex method is finite: The dual objective strictly increases at each iteration $\Rightarrow$ no basis repeats, and there are a finite number of bases.
- There are various approaches to guaranteeing finiteness in general:
  - **Bland's Rules:** Purely combinatorial, bad in practice.
  - **CPLEX:** A perturbation is introduced to guarantee $D_B > 0$.

Graphical Interpretation of Simplex Algorithms

A Graphical Solution



The Simplex Algorithm

A graphical representation

We now look at a graphical representation of the simplex method as it solves the following problem:

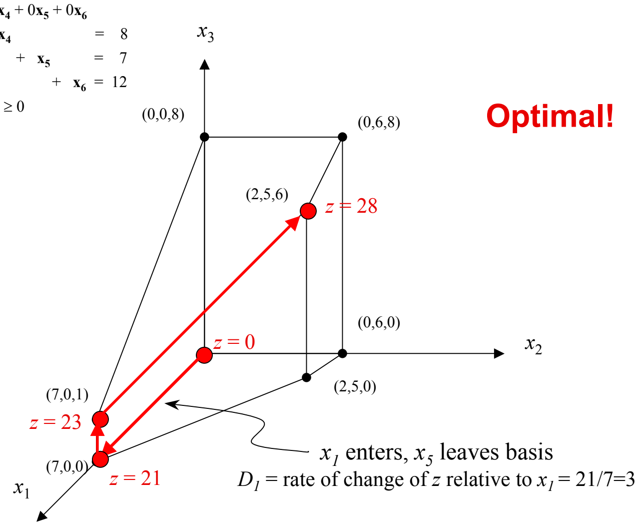
$$
\begin{array}{ll}
\text{Maximize} & 3x_1 + 2x_2 + 2x_3 \\
\text{Subject to} & x_1 + \; x_3 \le 8 \\
 & x_1 + \; x_2 \le 7 \\
 & x_1 + 2x_2 \le 12 \\
 & x_1, x_2, x_3 \ge 0
\end{array}
$$

The Primal Simplex Algorithm

$$\text{Maximize } z = 3x_1 + 2x_2 + 2x_3$$

Add slacks: Initial basis B = (4,5,6)

$$
\begin{array}{ll}
\text{Maximize} & 3x_1 + 2x_2 + 2x_3 + 0\mathbf{x_4} + 0\mathbf{x_5} + 0\mathbf{x_6} \\
\text{Subject to} & x_1 + x_3 + \mathbf{x_4} = 8 \\
 & x_1 + x_2 + \mathbf{x_5} = 7 \\
 & x_1 + 2x_2 + \mathbf{x_6} = 12 \\
 & x_1, x_2, x_3, \mathbf{x_4}, \mathbf{x_5}, \mathbf{x_6} \ge 0
\end{array}
$$



Computational History of Linear Programming

"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

MIP

Application of LP & MIP - I

- **Transportation-airlines**
  - Fleet assignment
  - Crew scheduling
  - Ground personnel scheduling
  - Yield management
  - Fuel allocation
  - Passenger mix
  - Booking control
  - Maintenance scheduling
  - Load balancing/freight packing
  - Airport traffic planning
  - Gate scheduling/assignment
  - Upset recover and management
- **Transportation-other**
  - Vehicle routing
  - Freight vehicle scheduling and assignment
  - Depot/warehouse location
  - Freight vehicle packing
  - Public transportation system operation
  - Rental car fleet management
- **Process industries**
  - Plant production scheduling and logistics
  - Capacity expansion planning
  - Pipeline transportation planning
  - Gasoline and chemical blending

MIP

Application of LP & MIP - II

- **Financial**
  - Portfolio selection and optimization
  - Cash management
  - Synthetic option development
  - Lease analysis
  - Capital budgeting and rationing
  - Bank financial planning
  - Accounting allocations
  - Securities industry surveillance
  - Audit staff planning
  - Assets/liabilities management
  - Unit costing
  - Financial valuation
  - Bank shift scheduling
  - Consumer credit delinquency management
  - Check clearing systems
  - Municipal bond bidding
  - Stock exchange operations
  - Debt financing
- **Manufacturing**
  - Product mix planning
  - Blending
  - Manufacturing scheduling
  - Inventory management
  - Job scheduling
  - Personnel scheduling
  - Maintenance scheduling and planning
  - Steel production scheduling
- **Coal Industry**
  - Coal sourcing/transportation logistics
  - Coal blending
  - Mining operations management
- **Forestry**
  - Forest land management
  - Forest valuation models
  - Planting and harvesting models

MIP

Application of LP & MIP - III

- **Agriculture**
  - Production planning
  - Farm land management
  - Agricultural pricing models
  - Crop and product mix decision models
  - Product distribution
- **Public utilities and natural resources**
  - Electric power distribution
  - Power generator scheduling
  - Power tariff rate determination
  - Natural gas distribution planning
  - Natural gas pipeline transportation
  - Water resource management
  - Alternative water supply evaluation
  - Water reservoir management
  - Public water transportation models
  - Mining excavation models
- **Oil and gas exploration and production**
  - Oil and gas production scheduling
  - Natural gas transportation scheduling
- **Communications and computing**
  - Circuit board (VLSI) layout
  - Logical circuit design
  - Magnetic field design
  - Complex computer graphics
  - Curve fitting
  - Virtual reality systems
  - Computer system capacity planning
  - Office automation
  - Multiprocessor scheduling
  - Telecommunications scheduling
  - Telephone operator scheduling
  - Telemarketing site selection

MIP

Application of LP & MIP - IV

- **Food processing**
  - Food blending
  - Recipe optimization
  - Food transportation logistics
  - Food manufacturing logistics and scheduling
- **Health care**
  - Hospital staff scheduling
  - Hospital layout
  - Health cost reimbursement
  - Ambulance scheduling
  - Radiation exposure models
- **Pulp and paper industry**
  - Inventory planning
  - Trim loss minimization
  - Waste water recycling
  - Transportation planning
- **Textile industry**
  - Pattern layout and cutting optimization
  - Production scheduling
- **Government and military**
  - Post office scheduling and planning
  - Military logistics
  - Target assignment
  - Missile detection
  - Manpower deployment
- **Miscellaneous applications**
  - Advertising mix/media scheduling
  - Pollution control models
  - Sales region definition
  - Sales force deployment

LP 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 MAN-DAYS.
- **First computer code – 1951**
- **LP used commercially – Early 60s**
- **Powerful mainframe codes introduced – Early 70s**
- **Computational progress stagnated – Mid 80s**
- **Remarkable progress last 15 years (PCs, new computer science and mathematics)**
  - We now have three algorithms: Primal & Dual Simplex, Barrier

Example: A Production Planning Model

401,640 constraints 1,584,000 variables

Solution time line (2.0 GHz P4)**:**

- 1988 (CPLEX 1.0): Houston, 13 Nov 2002

Example: A Production Planning Model

401,640 constraints 1,584,000 variables

Solution time line (2.0 GHz P4)**:**

- 1988 (CPLEX 1.0): 8.0 days (Berlin, 21 Nov)

Example: A Production Planning Model

401,640 constraints 1,584,000 variables

Solution time line (2.0 GHz P4)**:**

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

Example: A Production Planning Model

401,640 constraints 1,584,000 variables

Solution time line (2.0 GHz P4)**:**

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

Example: A Production Planning Model

401,640 constraints 1,584,000 variables

Solution time line (2.0 GHz P4)**:**

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

Example: A Production Planning Model

401,640 constraints 1,584,000 variables

Solution time line (2.0 GHz P4)**:**

- 1988 (CPLEX 1.0): 29.8 days
- 1997 (CPLEX 5.0): 1.5 hours
- 2002 (CPLEX 8.0): 86.7 seconds
- 2003 (February): 59.1 seconds

Speedup: >43500x

BIG TEST: The testing methodology

- Not possible for one test to cover 10+ years: Combined several tests.
- The biggest single test:
  - **Assembled 680 real LPs**
  - Test runs: Using a time limit **(4 days per LP) two chosen methods would be compared as follows:**
    - Run method 1: Generate 680 solve times
    - Run method 2: Generate 680 solve times
    - Compute 680 ratios and form **GEOMETRIC MEAN** (not arithmetic mean!)

(part of a) BIG Test Set

Model	Rows	Cols	NZs	Model	Rows	Cols	NZs	Model	Rows	Cols	NZs
M1	16223	28568	88340	M28	31770	272372	829040	M55	99578	326504	2102273
M2	16768	39474	203112	M29	33440	56624	161831	M56	105127	154699	358171
M3	17681	165188	690273	M30	34994	87510	208179	M57	108393	112955	602948
M4	18262	23211	136324	M31	35519	43582	557466	M58	118158	487427	974854
M5	19103	33490	276895	M32	35645	34675	208769	M59	123964	93288	459680
M6	19374	180670	5392558	M33	36400	92878	246006	M60	125211	159109	457198
M7	19519	45832	124280	M34	38782	261079	1508199	M61	129181	467192	1025706
M8	19844	55528	152952	M35	39951	125000	381259	M62	155265	377918	930166
M9	19999	85191	170369	M36	41340	64162	370839	M63	175147	358239	1211488
M10	21019	115761	728432	M37	41344	163569	1928534	M64	179080	707556	1570514
M11	22513	99785	337746	M38	41366	78750	2110518	M65	185929	189867	2787708
M12	22797	63995	172018	M39	43387	107164	189864	M66	186441	23732	397080
M13	23610	44063	154822	M40	43687	164831	722066	M67	209760	363092	1061495
M14	23700	23005	169045	M41	44150	200077	4966017	M68	269640	1205640	6481640
M15	23712	31680	81245	M42	44211	37199	321663	M69	280756	920198	5936426
M16	24377	46592	2139096	M43	47423	81915	228565	M70	319256	638512	1231403
M17	26618	38904	1067713	M44	48548	163200	617683	M71	344297	559428	1909649
M18	27349	97710	288421	M45	54447	326504	1807146	M72	589250	1533590	5327318
M19	27441	15128	96118	M46	55020	117910	391081	M73	716772	1169910	2511088
M20	27899	26243	261968	M47	55463	191233	840986	M74	1000000	1685236	3370472
M21	28240	55200	161640	M48	60384	100078	485414	M75	1128152	1587664	4496138
M22	28420	164024	505253	M49	63856	144693	717229	M76	1285665	1313795	4998121
M23	29002	111722	2632880	M50	66185	157496	418321	M77	1372333	2627821	13321433
M24	29017	20074	2001102	M51	67745	111891	305125	M78	1709857	1903725	4959650
M25	29147	9984	1013168	M52	69418	612608	1722112	M79	5034171	7365337	25596099
M26	29724	98124	196524	M53	84840	316800	1899600	M80	5822606	15228380	30960543
M27	30190	57000	623730	M54	95011	197489	749771	M81	6662791	9781747	34053329
								M82	10810259	19916187	49228262

LP Progress: 1988 – 2002

- ❑ Algorithms

  Primal simplex in 1988 *versus*

  Best(primal,dual,barrier) today **2360x**

- ❑ Machines **800x**

Net: Algorithm * Machine ~ 1 900 000x

(January Operations Research, http://www.ilog.com/products/technology/TR01-20.pdf)

This beats Moore's Law!

Algorithm comparison and other remarks …

- Dual simplex vs. primal: Dual 2.6x faster
- Best simplex vs. barrier: Barrier 1.2x faster
- Best of three vs. primal: Best 7.5x faster
- CPLEX 9.0 – 2003
  - Primal 1.2x improvement
  - Dual 1.7x improvement