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
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:

<table>
<thead>
<tr>
<th>Food</th>
<th>Serving Size</th>
<th>Energy (kcal)</th>
<th>Protein (g)</th>
<th>Calcium (mg)</th>
<th>Price per serving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oatmeal</td>
<td>28 g</td>
<td>110</td>
<td>4</td>
<td>2</td>
<td>$0.30</td>
</tr>
<tr>
<td>Chicken</td>
<td>100 g</td>
<td>205</td>
<td>32</td>
<td>12</td>
<td>$2.40</td>
</tr>
<tr>
<td>Eggs</td>
<td>2 large</td>
<td>160</td>
<td>13</td>
<td>54</td>
<td>$1.30</td>
</tr>
<tr>
<td>Whole milk</td>
<td>237 cc</td>
<td>160</td>
<td>6</td>
<td>285</td>
<td>$0.90</td>
</tr>
<tr>
<td>Cherry pie</td>
<td>170 g</td>
<td>420</td>
<td>4</td>
<td>22</td>
<td>$0.20</td>
</tr>
<tr>
<td>Pork and beans</td>
<td>260 g</td>
<td>260</td>
<td>14</td>
<td>80</td>
<td>$1.90</td>
</tr>
</tbody>
</table>

Variables

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

\[ x_1 \text{ servings of oatmeal} \]
\[ x_2 \text{ servings of chicken} \]
\[ x_3 \text{ servings of eggs} \]
\[ x_4 \text{ servings of milk} \]
\[ x_5 \text{ servings of cherry pie} \]
\[ x_6 \text{ servings of pork and beans} \]
### Diet Problem

#### Nutritional values

Bob is considering the following foods:

<table>
<thead>
<tr>
<th>Food</th>
<th>Serving Size</th>
<th>Energy (kcal)</th>
<th>Protein (g)</th>
<th>Calcium (mg)</th>
<th>Price per serving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oatmeal</td>
<td>28 g</td>
<td>110</td>
<td>4</td>
<td>2</td>
<td>$0.30</td>
</tr>
<tr>
<td>Chicken</td>
<td>100 g</td>
<td>205</td>
<td>32</td>
<td>12</td>
<td>$2.40</td>
</tr>
<tr>
<td>Eggs</td>
<td>2 large</td>
<td>160</td>
<td>13</td>
<td>54</td>
<td>$1.30</td>
</tr>
<tr>
<td>Whole milk</td>
<td>237 cc</td>
<td>160</td>
<td>8</td>
<td>285</td>
<td>$0.90</td>
</tr>
<tr>
<td>Cherry pie</td>
<td>170 g</td>
<td>420</td>
<td>4</td>
<td>24</td>
<td>$2.00</td>
</tr>
<tr>
<td>Pork and beans</td>
<td>260 g</td>
<td>260</td>
<td>14</td>
<td>80</td>
<td>$1.90</td>
</tr>
</tbody>
</table>

**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**

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

\[4x_1 + 32x_2 + 13x_3 + 8x_4 + 4x_5 + 14x_6 \geq 55\]

\[2x_1 + 12x_2 + 54x_3 + 285x_4 + 22x_5 + 80x_6 \geq 800\]

**Bounds**

\[x_1, x_2, x_3, x_4, x_5, x_6 \geq 0\]
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

\[
\begin{align*}
    \text{Minimize} & \quad c^T x \\
    \text{Subject to} & \quad Ax = b \\
    & \quad x \geq 0
\end{align*}
\]

(P)

Where \( c \in \mathbb{R}^n \), \( b \in \mathbb{R}^m \), \( A \in \mathbb{R}^{m \times n} \), 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

\[
\begin{align*}
    \text{Maximize} & \quad b^T \pi \\
    \text{Subject to} & \quad A^T \pi \leq c \\
    & \quad \pi \text{ free}
\end{align*}
\]

(D)

In this context, (P) is referred to as the primal linear program.
**Weak Duality Theorem**  
*(von Neumann 1947)*

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

\[
\begin{align*}
\text{Maximize} & \quad b^T \pi \\
\text{Minimize} & \quad c^T x
\end{align*}
\]

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

**Proof:**

\[
\begin{align*}
\pi^T b & = \pi^T Ax \\
& \leq c^T x
\end{align*}
\]

\[
\begin{align*}
Ax & = b \\
\pi^T A & \leq c^T & x \geq 0
\end{align*}
\]

---

**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 $(B_1, ..., B_m)$ taken from $\{1, ..., n\}$. $B$ 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 $N = \{1, ..., n\} \setminus B$. The corresponding basic solution $X \in \mathbb{R}^n$ is given by $X_N = 0$ and $X_B = A_B^{-1} b$. $B$ is called (primal) feasible if $X_B \geq 0$.

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

Primal Simplex Algorithm

(Dantzig, 1947)

Input: A feasible basis $B$ and vectors $X_B = A_B^{-1} b$ and $D_N = c_N - A_N^T c_B$.

- **Step 1**: (Pricing) If $D_N \geq 0$, stop, $B$ is optimal; else let $j = \text{argmin}\{D_k : k \in N\}$.

- **Step 2**: (FTRAN) Solve $A_By = A_j$.

- **Step 3**: (Ratio test) If $y \leq 0$, stop, (P) is unbounded; else, let $i = \text{argmin}\{X_{Bi}/y_i : y_i > 0\}$.

- **Step 4**: (BTRAN) Solve $A_B^T z = e_i$.

- **Step 5**: (Update) Compute $\alpha_i = -A_i^T z$. Let $B_i = j$. Update $X_B$ (using $y$) and $D_N$ (using $\alpha_i$).

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 $d$:

\[
\begin{align*}
\text{Maximize} & \quad b^T \pi \\
\text{Subject to} & \quad A^T \pi + d = c \\
& \quad \pi \text{ free, } d \geq 0 \iff A^T \pi \leq c
\end{align*}
\]

Given a basis $B$, the corresponding dual basic solution $\Pi, D$ is determined as follows:

\[
D_B=0 \quad \Rightarrow \quad \Pi = A_B^T c_B \quad \Rightarrow \quad D_N = c_N - A_N^T \Pi.
\]

$B$ is dual feasible if $D_N \geq 0$. 
An Important Fact

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

$$\Pi^Tb = c^TX.$$  

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

**Proof:** $c^TX = c_B^TX_B$ (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

  $$i = \arg\min\{X_{B_k} : k \in \{1, \ldots, m\}\}.$$

- **Step 2:** (BTRAN) Solve $B^T z = e_r$. Compute $\alpha_N = -A_N^T z$.

- **Step 3:** (Ratio test) If $\alpha_N \leq 0$, stop, (D) is unbounded; else, let

  $$j = \arg\min\{D_k / \alpha_k : \alpha_k > 0\}.$$

- **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 $\alpha_N$)

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

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

**Input:** A dual feasible basis $B$ and vectors $X_B = A_B^{-1}b$ & $D_N = c_N - A_N^T A_B^{-1} e_B$.

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

**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**
(⇒ primal infeasible)

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

\[ d_{B_i} \text{ enters basis } \quad d_j \text{ leaves basis} \]

in

\[
\begin{align*}
\text{Maximize} & \quad b^T \pi \\
\text{Subject to} & \quad A^T \pi + d = c \\
\pi \text{ free, } d & \geq 0
\end{align*}
\]

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

\[
\begin{align*}
d_B &= \theta e_i \\
d_N &= D_N - \theta \alpha_N \\
\pi &= \pi - \theta z,
\end{align*}
\]

where \( \alpha_N \) and \( z \) are as computed in the algorithm.
(Dual Unboundedness – cont.)

- Letting \( d, \pi \) be the corresponding dual solution as a function of \( \theta \). Using \( \alpha_N \) and \( z \) from dual algorithm,
  \[
  d_B = \theta e_i, \quad d_N = D_N - \theta \alpha_N, \quad \pi = \pi - \theta z.
  \]

- Using \( \theta > 0 \) and \( X_{Bi} < 0 \) yields
  \[
  \text{new\_objective} = \pi^T b = (\pi - \theta z)^T b
  \]
  \[
  = \pi^T b - \theta X_{Bi}
  \]
  \[
  = \text{old\_objective} - \theta X_{Bi} > \text{old\_objective}
  \]

- **Conclusion 1:** If \( \alpha_N \leq 0 \), then \( d_N \geq 0 \) \( \forall \theta > 0 \) \( \Rightarrow \) (D) is unbounded.

- **Conclusion 2:** If \( \alpha_N \) not \( \leq 0 \), then
  \[
  d_N \geq 0 \Rightarrow \theta \leq D_j/\alpha_j \ \forall \ \alpha_j > 0
  \]
  \[
  \Rightarrow \theta_{\text{max}} = \min\{D_j/\alpha_j: \alpha_j > 0\}
  \]

---

(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

Maximize $0.90x + 0.73y$ [OBJECTIVE]
Subject To
Constraint 1: $0.42x + 0.07y \leq 4200000$
Constraint 2: $0.13x + 0.39y \leq 3900000$
Constraint 3: $0.35x + 0.44y \leq 7000000$
$x \geq 0$
y \geq 0

Objective = $0.9 \times 0.882 + 0.73 \times 0.706$
= 13.1 million

Feasible Solutions

(0,0) (0,1) (0,1.5) (0,6)
(1,0) (2,0) (3,0) (0.882,0.706)
A graphical representation

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

Maximize \[ 3x_1 + 2x_2 + 2x_3 \]
Subject to \[
\begin{align*}
x_1 + x_3 & \leq 8 \\
x_1 + x_2 & \leq 7 \\
x_1 + 2x_2 & \leq 12 \\
x_1, x_2, x_3 & \geq 0
\end{align*}
\]

The Primal Simplex Algorithm

Add slacks: Initial basis \( B = (4,5,6) \)
Maximize \[ z = 3x_1 + 2x_2 + 2x_3 \]
Subject to \[
\begin{align*}
x_1 + x_3 + x_4 & = 8 \\
x_1 + x_2 + x_5 & = 7 \\
x_1 + 2x_2 + x_6 & = 12 \\
x_1, x_2, x_3, x_4, x_5, x_6 & \geq 0
\end{align*}
\]

Optimal!

\( x_1 \) enters, \( x_1 \) leaves basis
\[ D_j = \text{rate of change of } z \text{ relative to } x_j = 21/7 = 3 \]
“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
**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

**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
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

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): 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!)
<table>
<thead>
<tr>
<th>Model</th>
<th>Rows</th>
<th>Cols</th>
<th>NZs</th>
<th>Model</th>
<th>Rows</th>
<th>Cols</th>
<th>NZs</th>
<th>Model</th>
<th>Rows</th>
<th>Cols</th>
<th>NZs</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>16229</td>
<td>28586</td>
<td>86340</td>
<td>M27</td>
<td>30190</td>
<td>57000</td>
<td>623730</td>
<td>M38</td>
<td>33707</td>
<td>23272</td>
<td>62940</td>
</tr>
<tr>
<td>M2</td>
<td>16768</td>
<td>38474</td>
<td>203112</td>
<td>M28</td>
<td>33440</td>
<td>58624</td>
<td>161831</td>
<td>M39</td>
<td>99788</td>
<td>32604</td>
<td>320273</td>
</tr>
<tr>
<td>M3</td>
<td>17881</td>
<td>161588</td>
<td>66273</td>
<td>M40</td>
<td>34984</td>
<td>87510</td>
<td>208719</td>
<td>M41</td>
<td>100127</td>
<td>154099</td>
<td>358171</td>
</tr>
<tr>
<td>M4</td>
<td>18282</td>
<td>23211</td>
<td>13624</td>
<td>M42</td>
<td>35519</td>
<td>43582</td>
<td>557466</td>
<td>M43</td>
<td>110350</td>
<td>112555</td>
<td>972586</td>
</tr>
<tr>
<td>M5</td>
<td>19123</td>
<td>53490</td>
<td>77965</td>
<td>M44</td>
<td>35645</td>
<td>34775</td>
<td>206769</td>
<td>M45</td>
<td>122864</td>
<td>92988</td>
<td>495000</td>
</tr>
<tr>
<td>M6</td>
<td>19754</td>
<td>189070</td>
<td>532558</td>
<td>M46</td>
<td>36400</td>
<td>92678</td>
<td>246606</td>
<td>M47</td>
<td>125211</td>
<td>159109</td>
<td>497108</td>
</tr>
<tr>
<td>M7</td>
<td>19519</td>
<td>48522</td>
<td>124850</td>
<td>M48</td>
<td>38712</td>
<td>261709</td>
<td>1502819</td>
<td>M49</td>
<td>129181</td>
<td>497102</td>
<td>125706</td>
</tr>
<tr>
<td>M8</td>
<td>19944</td>
<td>55528</td>
<td>152625</td>
<td>M50</td>
<td>39661</td>
<td>125000</td>
<td>36129</td>
<td>M51</td>
<td>155265</td>
<td>377918</td>
<td>939199</td>
</tr>
<tr>
<td>M9</td>
<td>19989</td>
<td>65191</td>
<td>170589</td>
<td>M52</td>
<td>41340</td>
<td>61402</td>
<td>370830</td>
<td>M53</td>
<td>170147</td>
<td>382239</td>
<td>121486</td>
</tr>
<tr>
<td>M10</td>
<td>21309</td>
<td>115701</td>
<td>728432</td>
<td>M54</td>
<td>41344</td>
<td>163569</td>
<td>152834</td>
<td>M55</td>
<td>179806</td>
<td>707656</td>
<td>197014</td>
</tr>
<tr>
<td>M11</td>
<td>22513</td>
<td>98785</td>
<td>337476</td>
<td>M56</td>
<td>41356</td>
<td>79750</td>
<td>2115108</td>
<td>M57</td>
<td>189629</td>
<td>188867</td>
<td>278708</td>
</tr>
<tr>
<td>M12</td>
<td>22797</td>
<td>68656</td>
<td>17218</td>
<td>M58</td>
<td>43387</td>
<td>107164</td>
<td>189864</td>
<td>M59</td>
<td>196441</td>
<td>23712</td>
<td>39736</td>
</tr>
<tr>
<td>M13</td>
<td>23610</td>
<td>44083</td>
<td>154822</td>
<td>M60</td>
<td>43967</td>
<td>164831</td>
<td>722996</td>
<td>M61</td>
<td>205760</td>
<td>382982</td>
<td>101495</td>
</tr>
<tr>
<td>M14</td>
<td>23700</td>
<td>25265</td>
<td>168445</td>
<td>M62</td>
<td>44150</td>
<td>200277</td>
<td>498617</td>
<td>M63</td>
<td>209940</td>
<td>120564</td>
<td>481640</td>
</tr>
<tr>
<td>M15</td>
<td>23712</td>
<td>31680</td>
<td>81245</td>
<td>M64</td>
<td>44211</td>
<td>37199</td>
<td>321863</td>
<td>M65</td>
<td>244018</td>
<td>92198</td>
<td>593426</td>
</tr>
<tr>
<td>M16</td>
<td>24377</td>
<td>48552</td>
<td>213009</td>
<td>M66</td>
<td>47143</td>
<td>81915</td>
<td>220865</td>
<td>M67</td>
<td>315256</td>
<td>68512</td>
<td>123163</td>
</tr>
<tr>
<td>M17</td>
<td>26218</td>
<td>38554</td>
<td>156713</td>
<td>M68</td>
<td>46048</td>
<td>162200</td>
<td>617853</td>
<td>M69</td>
<td>344297</td>
<td>505428</td>
<td>100649</td>
</tr>
<tr>
<td>M18</td>
<td>27349</td>
<td>97710</td>
<td>286421</td>
<td>M70</td>
<td>54447</td>
<td>328504</td>
<td>1507146</td>
<td>M71</td>
<td>555250</td>
<td>153590</td>
<td>537318</td>
</tr>
<tr>
<td>M19</td>
<td>27441</td>
<td>15126</td>
<td>66118</td>
<td>M72</td>
<td>55220</td>
<td>117910</td>
<td>301081</td>
<td>M73</td>
<td>715722</td>
<td>1165910</td>
<td>251108</td>
</tr>
<tr>
<td>M20</td>
<td>27899</td>
<td>26243</td>
<td>281568</td>
<td>M74</td>
<td>55463</td>
<td>191233</td>
<td>840896</td>
<td>M75</td>
<td>1000000</td>
<td>1685236</td>
<td>3370742</td>
</tr>
<tr>
<td>M21</td>
<td>28240</td>
<td>55200</td>
<td>181840</td>
<td>M76</td>
<td>60384</td>
<td>100578</td>
<td>485414</td>
<td>M77</td>
<td>1123032</td>
<td>1587664</td>
<td>4469136</td>
</tr>
<tr>
<td>M22</td>
<td>28420</td>
<td>16404</td>
<td>50253</td>
<td>M78</td>
<td>63856</td>
<td>144993</td>
<td>717209</td>
<td>M79</td>
<td>1288665</td>
<td>1317395</td>
<td>4999121</td>
</tr>
<tr>
<td>M23</td>
<td>29032</td>
<td>11172</td>
<td>262280</td>
<td>M80</td>
<td>68105</td>
<td>157496</td>
<td>416521</td>
<td>M81</td>
<td>1373253</td>
<td>2670398</td>
<td>13221433</td>
</tr>
<tr>
<td>M24</td>
<td>29617</td>
<td>20014</td>
<td>200102</td>
<td>M82</td>
<td>67745</td>
<td>111891</td>
<td>305125</td>
<td>M83</td>
<td>170957</td>
<td>1037262</td>
<td>4995650</td>
</tr>
<tr>
<td>M25</td>
<td>29517</td>
<td>19824</td>
<td>19824</td>
<td>M84</td>
<td>94940</td>
<td>31850</td>
<td>198963</td>
<td>M85</td>
<td>505269</td>
<td>1523580</td>
<td>3990543</td>
</tr>
<tr>
<td>M26</td>
<td>29724</td>
<td>88124</td>
<td>155242</td>
<td>M86</td>
<td>95011</td>
<td>197489</td>
<td>749771</td>
<td>M87</td>
<td>6662791</td>
<td>982747</td>
<td>349252509</td>
</tr>
<tr>
<td>M27</td>
<td>30190</td>
<td>57000</td>
<td>623730</td>
<td>M88</td>
<td>95011</td>
<td>197489</td>
<td>749771</td>
<td>M89</td>
<td>1796259</td>
<td>1991667</td>
<td>4823062</td>
</tr>
</tbody>
</table>

**LP Progress: 1988 – 2002**

- **Algorithms**
  - Primal simplex in 1988 *versus* Best(primal,dual,barrier) today *2360x*

- **Machines** *800x*

**Net:** Algorithm * Machine ~ 1 900 000x


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