Linear Programming

It's never been programming, and now it's not even linear!

William J. Martin

Department of Mathematical Sciences
and
Department of Computer Science
Worcester Polytechnic Institute

WPI Math Meet, October 19, 2011

Appreciative Graduates

It's always nice to see the impact you've had on your former students:

Appreciative Graduates

It's always nice to see the impact you've had on your former students:

Math Diversions

Q: How much time does it save to cut diagonally across a square parking lot, park, or lawn?

Q: For beginning protractor users: Zoom in on Google maps and figure out the system for numbering runways at airports.

Q: What's wrong with this McGraw-Hill pre-calculus text?

> *To find horizontal asymptotes of a rational function $y = f(x)$, first solve for x and locate those values of y where this function is undefined.*

Linear Programming in the Sci-Fi Literature

'I don't want to bore you', Harvey said, 'but you should understand that these heaps of wire can practically think — linear programming — which means that instead of going through all the alternatives they have a hunch which is the right one.'

– Billion Dollar Brain, by Len Deighton

Sheriff Greg Bartlett

- Morgan County, Alabama, jail holds about 300
- state food allowance is \$1.75 per inmate, per day
- county sheriffs like Bartlett are in charge of diet, food procurement, kitchen, etc.

Sheriff Greg Bartlett

- Morgan County, Alabama, jail holds about 300
- state food allowance is $1.75 per inmate, per day
- county sheriffs like Bartlett are in charge of diet, food procurement, kitchen, etc.
- any allowance not spent on food goes directly to the sheriff

Sheriff Greg Bartlett

- food allowance \$1.75 per inmate, per day. Unspent money goes to sheriff
- federal agents arrested the sheriff Wed Jan 7th 2009
- judge sent him to jail (for one day) until he proposed a new food plan

Sheriff Greg Bartlett

- food allowance \$1.75 per inmate, per day. Unspent money goes to sheriff
- federal agents arrested the sheriff Wed Jan 7th 2009
- judge sent him to jail (for one day) until he proposed a new food plan
- Bartlett had pocketed \$212,000 in surplus meal monies in just 3 years
- $1.75 \times 300 \times 365 = \$191,625$ per year allowance

What Sheriff Bartlett missed

- diet problems are standard applications of linear optimization

What Sheriff Bartlett missed

- diet problems are standard applications of linear optimization
- **variables:** quantity of each food to include in one person's daily diet
- **constraints:** meet FDA nutritional requirements (protein, vitamins, fibre, ...)

What Sheriff Bartlett missed

- diet problems are standard applications of linear optimization
- **variables:** quantity of each food to include in one person's daily diet
- **constraints:** meet FDA nutritional requirements (protein, vitamins, fibre, ...)
- E.g., each person should get at least 2000 calories per day

What Sheriff Bartlett missed

- diet problems are standard applications of linear optimization
- **variables:** quantity of each food to include in one person's daily diet
- **constraints:** meet FDA nutritional requirements (protein, vitamins, fibre, . . .)
- E.g., each person should get at least 2000 calories per day
- **objective:** minimize cost

Photo Credit: trusty guides

Photo Credit: stockfood.com

What Sheriff Bartlett missed

- **variables:** quantity of each food to include
- **constraints:** meet FDA nutritional requirements
- **objective:** minimize cost

What Sheriff Bartlett missed

- **variables:** quantity of each food to include
- **constraints:** meet FDA nutritional requirements
- **objective:** minimize cost
- Mike Trick (Carnegie Mellon) used linear programming, found

What Sheriff Bartlett missed

- **variables:** quantity of each food to include
- **constraints:** meet FDA nutritional requirements
- **objective:** minimize cost
- Mike Trick (Carnegie Mellon) used linear programming, found
- The good sheriff could have met all dietary requirements on just \$0.96 per inmate per day:

What Sheriff Bartlett missed

- **variables:** quantity of each food to include
- **constraints:** meet FDA nutritional requirements
- **objective:** minimize cost
- Mike Trick (Carnegie Mellon) used linear programming, found
- The good sheriff could have met all dietary requirements on just $0.96 per inmate per day:
  - 0.24 servings of raw carrots (2 cents per inmate per day)
  - 3.60 servings of peanut butter (25 cents)
  - 4.82 servings of popcorn (19 cents)
  - 3.54 servings of baked potato (21 cents)
  - 2.17 servings of skim milk (28 cents)

What Sheriff Bartlett missed

- **variables:** quantity of each food to include
- **constraints:** meet FDA nutritional requirements
- **objective:** minimize cost
- Mike Trick (Carnegie Mellon) used linear programming, found
- The good sheriff could have met all dietary requirements on just \$0.96 per inmate per day:
  - 0.24 servings of raw carrots (2 cents per inmate per day)
  - 3.60 servings of peanut butter (25 cents)
  - 4.82 servings of popcorn (19 cents)
  - 3.54 servings of baked potato (21 cents)
  - 2.17 servings of skim milk (28 cents)

Isn't math wonderful!

Shameless Digressions

Shameless Digressions

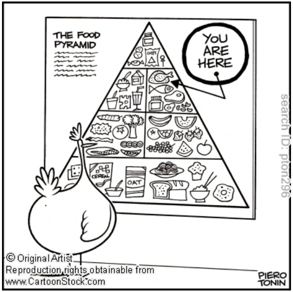


Photo Credit: cartoon stock.com

Here's a web interface to linear programming software that allows you to solve your own diet problems:

`http://www.neos-guide.org/NEOS/index.php/Diet_Problem_Demo`

A Few Simple Messages

- math majors get jobs (Jobs Rated 2011: top four professions are . . .)
- math is still alive, new theorems every year, many unsolved problems
- for example, linear programming is cool, powerful, versatile, changing
- In 1970, a study estimated that 25% of all computing was devoted to LP

Solving Systems of Linear Equations

A system of linear equations (3 equations, 3 unknowns):

$$\begin{array}{rcrcrcr} 3x & + & 5y & & & = & 18 \\ 2x & & & - & 5z & = & 2 \\ & & 2y & + & 3z & = & 6 \end{array}$$

Solving Systems of Linear Equations

A system of linear equations (3 equations, 3 unknowns):

$$\begin{array}{rcrcrcr} 3x & + & 5y & & & = & 18 \\ 2x & & & - & 5z & = & 2 \\ & & 2y & + & 3z & = & 6 \end{array}$$

The solution set is a line.

Solving Systems of Linear Equations

A system of linear equations (3 equations, 3 unknowns):

$$\begin{array}{rcrcrcr} 3x & + & 5y & & & = & 18 \\ 2x & & & - & 5z & = & 2 \\ & & 2y & + & 3z & = & 6 \end{array}$$

The solution set is a line. Let's parametrize it by t:

$$x = -4 + 5t, \qquad y = 6 - 3t, \qquad z = -2 + 2t.$$

Solving Systems of Linear Equations

A system of linear equations (3 equations, 3 unknowns):

$$\begin{array}{rcrcrcr} 3x & + & 5y & & & = & 18 \\ 2x & & & - & 5z & = & 2 \\ & & 2y & + & 3z & = & 6 \end{array}$$

The solution set is a line. Let's parametrize it by t:

$$x = -4 + 5t, \qquad y = 6 - 3t, \qquad z = -2 + 2t.$$

Today's Question: Can we find a solution with all entries non-negative?

Solving Systems of Linear Equations

A system of linear equations (3 equations, 3 unknowns):

$$\begin{array}{rcrcrcr} 3x & + & 5y & & & = & 18 \\ 2x & & & - & 5z & = & 2 \\ & & 2y & + & 3z & = & 6 \end{array}$$

The solution set is a line. Let's parametrize it by t:

$$x = -4 + 5t, \qquad y = 6 - 3t, \qquad z = -2 + 2t.$$

Today's Question: Can we find a solution with all entries non-negative?

Yes: For $t = 1$, $(x, y, z) = (1, 3, 0)$;
for $t = 2$, $(x, y, z) = (6, 0, 2)$.

Solving Systems of Linear Equations

A system of linear equations (3 equations, 3 unknowns):

$$\begin{array}{rcrcrcr} 3x & + & 5y & & & = & 18 \\ 2x & & & - & 5z & = & 2 \\ & & 2y & + & 3z & = & 6 \end{array}$$

The solution set is a line. Let's parametrize it by t:

$$x = -4 + 5t, \qquad y = 6 - 3t, \qquad z = -2 + 2t.$$

Today's Question: Can we find a solution with all entries non-negative?

Yes: For $t = 1$, $(x, y, z) = (1, 3, 0)$;
for $t = 2$, $(x, y, z) = (6, 0, 2)$.

Any more solutions?

Solving Systems of Linear Equations

A system of linear equations (3 equations, 3 unknowns):

$$\begin{array}{rcrcrcr} 3x & + & 5y & & & = & 18 \\ 2x & & & - & 5z & = & 2 \\ & & 2y & + & 3z & = & 6 \end{array}$$

The solution set is a line. Let's parametrize it by t:

$$x = -4 + 5t, \qquad y = 6 - 3t, \qquad z = -2 + 2t.$$

Today's Question: Can we find a solution with all entries non-negative?

Yes: For $t = 1$, $(x, y, z) = (1, 3, 0)$;
for $t = 2$, $(x, y, z) = (6, 0, 2)$.

Any more solutions? Yes!

All solutions with $1 \leq t \leq 2$ are non-negative vectors. (Convexity!)

The Language of Linear Algebra

Let's revisit our system of 3 equations in 3 unknowns:

$$\begin{array}{rcrcrcr} 3x_1 & + & 5x_2 & & & = & 18 \\ 2x_1 & & & - & 5x_3 & = & 2 \\ & & 2x_2 & + & 3x_3 & = & 6 \end{array}$$

This is written $\mathbf{Ax} = \mathbf{b}$ for

$$\mathbf{A} = \begin{bmatrix} 3 & 5 & 0 \\ 2 & 0 & -5 \\ 0 & 2 & 3 \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 18 \\ 2 \\ 6 \end{bmatrix}.$$

The Basic Problem

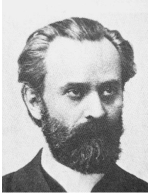
Given a matrix $\mathbf{A}$ with m rows and n columns and a vector $\mathbf{b}$ with m rows, we have the solution set of all vectors $\mathbf{x}$ of length n satisfying

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

Linear Programming: Find a solution with no negative entries.

This is equivalent to the more familiar formulation of maximizing some linear profit function subject to some collection of constraints which are linear equations or inequalities.

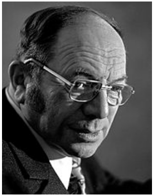
Farkas' Lemma



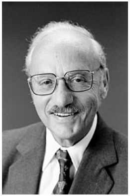
Gyula Farkas (1847-1930)

Theorem (1902): Given matrix $\mathbf{A}$ and vector $\mathbf{b}$
EITHER
there is a non-negative vector $\mathbf{x}$ such that $\mathbf{Ax} = \mathbf{b}$
OR
there is a vector $\mathbf{y}$ such that $\mathbf{y}^\top \mathbf{A} \geq \mathbf{0}$ and yet $\mathbf{y}^\top \mathbf{b} < 0$
NOT BOTH.

History of Linear Programming



Leonid Kantorovich (1912-1986)

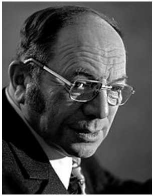


George Dantzig (1914-2005)

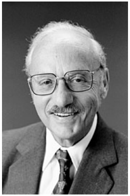
John von Neumann (1903-1957)

- Operations Research
- Linear Programming (1939-1947)
- Simplex Algorithm (1947)
- Duality Theorem (1950)

History of Linear Programming



Leonid Kantorovich (1912-1986)



George Dantzig (1914-2005)

John von Neumann (1903-1957)

- (Military) **Operations** Research
- Linear **Program**ming (a **program** is a plan of action)
- Simplex Algorithm
- Duality Theorem (rediscovery of Farkas' Lemma)

Polynomial Time Algorithms are "Fast"

- We say an algorithm runs in "polynomial time", if there exists a polynomial $f(n)$ such that the algorithm takes at most $f(n)$ steps on n pieces of input for all $n \geq 0$

Polynomial Time Algorithms are "Fast"

- We say an algorithm runs in "polynomial time", if there exists a polynomial $f(n)$ such that the algorithm takes at most $f(n)$ steps on n pieces of input for all $n \geq 0$
- Technically, "pieces" means "bits", but we can say "variables x constraints" or even "variables + constraints"

Polynomial Time Algorithms are "Fast"

- We say an algorithm runs in "polynomial time", if there exists a polynomial $f(n)$ such that the algorithm takes at most $f(n)$ steps on n pieces of input for all $n \geq 0$
- Technically, "pieces" means "bits", but we can say "variables x constraints" or even "variables + constraints"
- E.g., sorting, shortest path are polynomial time; graph coloring, traveling salesman are probably exponential

Polynomial Time Algorithms are "Fast"

- We say an algorithm runs in "polynomial time", if there exists a polynomial $f(n)$ such that the algorithm takes at most $f(n)$ steps on n pieces of input for all $n \geq 0$
- Technically, "pieces" means "bits", but we can say "variables x constraints" or even "variables + constraints"
- E.g., sorting, shortest path are polynomial time; graph coloring, traveling salesman are probably exponential
- **Q:** If you can verify a YES answer in polynomial time and you can verify a NO answer in polynomial time, can't you decide whether the answer is YES or NO in polynomial time? (I.e., is $P = NP \cap coNP$?)
- Big question in the 1970s: Can linear programming problems be solved in polynomial time?

Ellipsoid Method

Leonid Khachian, 1979 in Moscow, provided the first polynomial time algorithm for linear programming.

Ellipsoid Method

Leonid Khachian, 1979 in Moscow, provided the first polynomial time algorithm for linear programming.

- Ellipsoid Method solves our version $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$
- originally developed by Shor, Nemirovski & Yudin in Russia in the early 1970s

Ellipsoid Method

Leonid Khachian, 1979 in Moscow, provided the first polynomial time algorithm for linear programming.

- Ellipsoid Method solves our version $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$
- originally developed by Shor, Nemirovski & Yudin in Russia in the early 1970s
- Start with an ellipsoid that contains the region (if non-empty)

Ellipsoid Method

Leonid Khachian, 1979 in Moscow, provided the first polynomial time algorithm for linear programming.

- Ellipsoid Method solves our version $\mathbf{Ax} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$
- originally developed by Shor, Nemirovski & Yudin in Russia in the early 1970s
- Start with an ellipsoid that contains the region (if non-empty)
- If the center of the ellipsoid is feasible, DONE

Ellipsoid Method

Leonid Khachian, 1979 in Moscow, provided the first polynomial time algorithm for linear programming.

- Ellipsoid Method solves our version $\mathbf{Ax} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$
- originally developed by Shor, Nemirovski & Yudin in Russia in the early 1970s
- Start with an ellipsoid that contains the region (if non-empty)
- If the center of the ellipsoid is feasible, DONE
- Else, find a separating hyperplane

Ellipsoid Method

Leonid Khachian, 1979 in Moscow, provided the first polynomial time algorithm for linear programming.

- Ellipsoid Method solves our version $\mathbf{Ax} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$
- originally developed by Shor, Nemirovski & Yudin in Russia in the early 1970s
- Start with an ellipsoid that contains the region (if non-empty)
- If the center of the ellipsoid is feasible, DONE
- Else, find a separating hyperplane
- Now find a smaller ellipsoid that contains the half-ellipsoid that contains the feasible region. Iterate.

Ellipsoid Method

Leonid Khachian, 1979 in Moscow, provided the first polynomial time algorithm for linear programming.

- Ellipsoid Method solves our version $\mathbf{Ax} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$
- originally developed by Shor, Nemirovski & Yudin in Russia in the early 1970s
- Start with an ellipsoid that contains the region (if non-empty)
- If the center of the ellipsoid is feasible, DONE
- Else, find a separating hyperplane
- Now find a smaller ellipsoid that contains the half-ellipsoid that contains the feasible region. Iterate.
- **The Bad News:** high complexity ($n^4 \log \frac{R}{r}$ iterations each requiring $O(n^2)$ steps)

Ellipsoid Method

Leonid Khachian, 1979 in Moscow, provided the first polynomial time algorithm for linear programming.

- Ellipsoid Method solves our version $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$
- originally developed by Shor, Nemirovski & Yudin in Russia in the early 1970s
- Start with an ellipsoid that contains the region (if non-empty)
- If the center of the ellipsoid is feasible, DONE
- Else, find a separating hyperplane
- Now find a smaller ellipsoid that contains the half-ellipsoid that contains the feasible region. Iterate.
- **The Bad News:** high complexity ($n^4 \log \frac{R}{r}$ iterations each requiring $O(n^2)$ steps)
- **Worse News:** numerically unstable in higher dimensions

Karmarkar's Algorithm

- Khachiyan's result generated great interest in efficient algorithms
- Narendra Karmarkar (PhD, Berkeley 1983, w/ Richard Karp)

Karmarkar's Algorithm

- Khachiyan's result generated great interest in efficient algorithms
- Narendra Karmarkar (PhD, Berkeley 1983, w/ Richard Karp)
- ideas from non-linear optimization applied to linear problem
- log-barrier method, Newton-type method with projective (non-linear) scaling

Karmarkar's Algorithm

- Khachiyan's result generated great interest in efficient algorithms
- Narendra Karmarkar (PhD, Berkeley 1983, w/ Richard Karp)
- ideas from non-linear optimization applied to linear problem
- log-barrier method, Newton-type method with projective (non-linear) scaling
- running time (according to wikipedia) is $O(n^{7/2}\ L^2\ \log L\ \log\log L)$ for a problem with n variables and L bits of input
- Very complicated, but similar to Affine Scaling Method, which is easy to teach

Patents on Mathematics?

- When Karmarkar published his algorithm, he was employed at Bell Labs, Murray Hill, NJ

Patents on Mathematics?

- When Karmarkar published his algorithm, he was employed at Bell Labs, Murray Hill, NJ
- AT&T tried to patent the algorithm,

Patents on Mathematics?

- When Karmarkar published his algorithm, he was employed at Bell Labs, Murray Hill, NJ
- AT&T tried to patent the algorithm, succeeded (patent expired in 2006)
- there was a publicly available version —provably polynomial-time — and a faster secret version

Patents on Mathematics?

- When Karmarkar published his algorithm, he was employed at Bell Labs, Murray Hill, NJ
- AT&T tried to patent the algorithm, succeeded (patent expired in 2006)
- there was a publicly available version —provably polynomial-time — and a faster secret version
- AT&T's KORBX computer implemented Karmarkar's algorithm for linear and integer programming

Patents on Mathematics?

- When Karmarkar published his algorithm, he was employed at Bell Labs, Murray Hill, NJ
- AT&T tried to patent the algorithm, succeeded (patent expired in 2006)
- there was a publicly available version —provably polynomial-time — and a faster secret version
- AT&T's KORBX computer implemented Karmarkar's algorithm for linear and integer programming
- Price: \$8.9 million.

Patents on Mathematics?

- When Karmarkar published his algorithm, he was employed at Bell Labs, Murray Hill, NJ
- AT&T tried to patent the algorithm, succeeded (patent expired in 2006)
- there was a publicly available version —provably polynomial-time — and a faster secret version
- AT&T's KORBX computer implemented Karmarkar's algorithm for linear and integer programming
- Price: $8.9 million. Number of computers sold: **2**

Patents on Mathematics?

- When Karmarkar published his algorithm, he was employed at Bell Labs, Murray Hill, NJ
- AT&T tried to patent the algorithm, succeeded (patent expired in 2006)
- there was a publicly available version —provably polynomial-time — and a faster secret version
- AT&T's KORBX computer implemented Karmarkar's algorithm for linear and integer programming
- Price: $8.9 million. Number of computers sold: **2**
- Anecdote: AT&T applied this algorithm to optimize their Pacific basin network
- I was told that this saved the company $20 million per year in operating costs

A Flurry of Activity to this Day

- Interior point methods have been a very active area of research since Karmarkar's result
- We now have polynomial time algorithms for many types of "conic programming" problems
- Most importantly, we have efficient algorithms for **semidefinite programming**
- SDP encompasses LP, but also includes special types of quadratic constraints
- many applications in finance, graph theory, control theory, ...

Game Theory

Suppose we play Rock/Paper/Scissors many many times, with our opponent carefully watching our behavior.

Game Theory

Suppose we play Rock/Paper/Scissors many many times, with our opponent carefully watching our behavior.

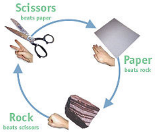
Our optimal strategy is to play **R**, **S**, **P** at random, each with 1/3 probability.

But what if the payoff for winning with Rock or Scissors is \$1 and the payoff for winning with Paper is \$2?

Q: What is the optimal strategy now?

Asymmetric Rock/Paper/Scissors

Setting: We play Rock/Paper/Scissors many times. The payoff for winning with Rock or Scissors is \$1;
the payoff for winning with Paper is \$2.

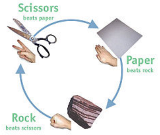


$$
\begin{array}{lrcrcrcl}
\textbf{maximize} & & & & & w & & \\
 & & & 2x_P & - & x_S & \geq & w \\
 & -2x_R & & & + & x_S & \geq & w \\
 & x_R & - & x_P & & & \geq & w \\
 & x_R & + & x_P & + & x_S & = & 1 \\
 & x_R, & & x_P, & & x_S & \geq & 0
\end{array}
$$

Optimal Solution:

Asymmetric Rock/Paper/Scissors

Setting: We play Rock/Paper/Scissors many times. The payoff for winning with Rock or Scissors is \$1;
the payoff for winning with Paper is \$2.



$$
\begin{array}{lrcrcrcl}
\textbf{maximize} & & & & & w & & \\
 & & & 2x_P & - & x_S & \geq & w \\
 & -2x_R & & & + & x_S & \geq & w \\
 & x_R & - & x_P & & & \geq & w \\
 & x_R & + & x_P & + & x_S & = & 1 \\
 & x_R, & & x_P, & & x_S & \geq & 0
\end{array}
$$

Optimal Solution: $x_R = x_P = 1/4, \quad x_S = 1/2$

LP really is Artificial Intelligence

Hanshin Expressway, Osaka Japan

- very expensive toll road through a congested area
- Light sensors every 500m measure traffic volume
- A computer solves a linear programming problem every 15 minutes
- Decides which on-ramps to open and close
- Objective: to maximize number of vehicles subject to no traffic jams

The End

Good luck to all your Math Meet competitors!