

15-853: Algorithms in the Real World

Linear and Integer Programming I

- Introduction
- Geometric Interpretation
- Simplex Method

15-853

Page1

Linear and Integer Programming

Linear or Integer programming

minimize $z = c^T x$ **cost or objective function**
subject to $Ax = b$ **equalities**
 $x \geq 0$ **inequalities**
 $c \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{n \times m}$

Linear programming:

$x \in \mathbb{R}^n$ (polynomial time)

Integer programming:

$x \in \mathbb{Z}^n$ (NP-complete)

Extremely general framework, especially IP

15-853

Page2

Related Optimization Problems

Unconstrained optimization

$\min\{f(x) : x \in \mathbb{R}^n\}$

Constrained optimization

$\min\{f(x) : c_i(x) \leq 0, i \in I, c_j(x) = 0, j \in E\}$

Quadratic programming

$\min\{1/2 x^T Q x + c^T x : a_i^T x \leq b_i, i \in I, a_i^T x = b_i, j \in E\}$

Zero-One programming

$\min\{c^T x : Ax = b, x \in \{0,1\}^n, c \in \mathbb{R}^n, b \in \mathbb{R}^m\}$

Mixed Integer Programming

$\min\{c^T x : Ax = b, x \geq 0, x_i \in \mathbb{Z}, i \in I, x_r \in \mathbb{R}, r \in R\}$

15-853

Page3

How important is optimization?

- 50+ packages available
- 1300+ papers just on interior-point methods
- 100+ books in the library
- 10+ courses at most Universities
- 100s of companies
- All major airlines, delivery companies, trucking companies, manufacturers, ... make serious use of optimization.

15-853

Page4

Linear+Integer Programming Outline

Linear Programming

- General formulation and geometric interpretation
- Simplex method
- Ellipsoid method
- Interior point methods

Integer Programming

- Various reductions of NP hard problems
- Linear programming approximations
- Branch-and-bound + cutting-plane techniques
- Case study from Delta Airlines

15-853

Page5

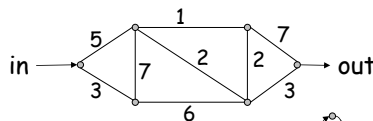
Applications of Linear Programming

1. A substep in most integer and mixed-integer linear programming (MIP) methods
2. Selecting a mix: oil mixtures, portfolio selection
3. Distribution: how much of a commodity should be distributed to different locations.
4. Allocation: how much of a resource should be allocated to different tasks
5. Network Flows

15-853

Page6

Linear Programming for Max-Flow



Create two variables per edge: x_1, x_1'

Create one equality per vertex:

$$x_1 + x_2 + x_3' = x_1' + x_2' + x_3$$

and two inequalities per edge:

$$x_1 \leq 3, x_1' \leq 3$$

add edge x_0 from out to in

maximize x_0

15-853

Page7

In Practice

In the "real world" most problems involve at least some integral constraints.

- Many resources are integral
- Can be used to model yes/no decisions (0-1 variables)

Therefore "1. A subset in integer or MIP programming" is the most common use in practice

15-853

Page8

Algorithms for Linear Programming

- **Simplex** (Dantzig 1947)
 - **Ellipsoid** (Kachian 1979)
first algorithm known to be **polynomial time**
 - **Interior Point**
first practical polynomial-time algorithms
 - **Projective method** (Karmakar 1984)
 - **Affine Method** (Dikin 1967)
 - **Log-Barrirer Methods** (Frisch 1977, Fiacco 1968, Gill et.al. 1986)
- Many of the interior point methods can be applied to nonlinear programs. Not known to be poly. time

15-853

Page9

State of the art

1 million variables

10 million nonzeros

No clear winner between Simplex and Interior Point

- Depends on the problem
- Interior point methods are subsuming more and more cases
- All major packages supply both

The truth: the sparse matrix routines, make or break both methods.

The best packages are highly sophisticated.

15-853

Page10

Comparisons, 1994

problem	Simplex (primal)	Simplex (dual)	Barrier + crossover
binpacking	29.5	62.8	560.6
distribution	18,568.0	won't run	too big
forestry	1,354.2	1,911.4	2,348.0
maintenance	57,916.3	89,890.9	3,240.8
crew	7,182.6	16,172.2	1,264.2
airfleet	71,292.5	108,015.0	37,627.3
energy	3,091.1	1,943.8	858.0
4color	45,870.2	won't run	too big

15-853

Page11

Formulations

There are many ways to formulate linear programs:

- **objective (or cost) function**
maximize $c^T x$, or
minimize $c^T x$, or
find any feasible solution
- **(in)equalities**
 $Ax \leq b$, or
 $Ax \geq b$, or
 $Ax = b$, or any combination
- **nonnegative variables**
 $x \geq 0$, or not

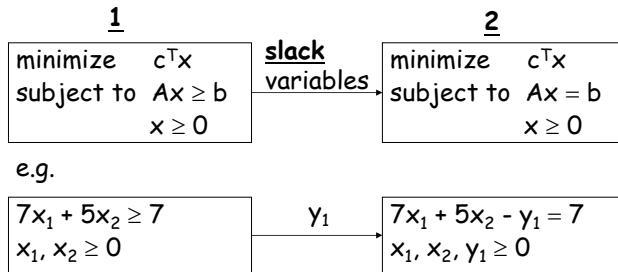
Fortunately it is pretty easy to convert among forms

15-853

Page12

Formulations

The two **most common** formulations:



More on slack variables later.

15-853

Page13

Geometric View

A **polytope** in n-dimensional space

Each inequality corresponds to a half-space.

The "feasible set" is the intersection of the half-spaces.

This corresponds to a polytope

The optimal solution is at a corner.

Simplex moves around on the surface of the polytope

Interior-Point methods move within the polytope

15-853

Page14

Geometric View

minimize:

$$z = -2x_1 - 3x_2$$

subject to:

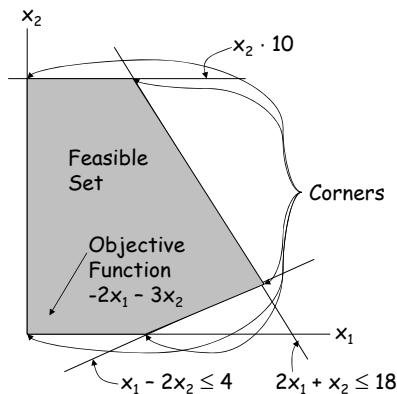
$$x_1 - 2x_2 \leq 4$$

$$2x_1 + x_2 \leq 18$$

$$x_2 \leq 10$$

$$x_1, x_2 \geq 0$$

An intersection of
5 halfspaces



15-853

Page15

Notes about higher dimensions

For n dimensions and no degeneracy

Each corner (extreme point) consists of:

- n intersecting n-1 dimensional **hyperplanes**

e.g. 3, 2d planes in 3d

- n intersecting **edges**

Each edge corresponds to moving off of one hyperplane (still constrained by n-1 of them)

Simplex will move from corner to corner along the edges

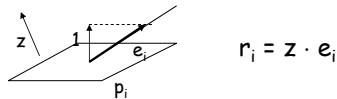
15-853

Page16

Optimality and Reduced Cost

The **Optimal** solution must include a corner.

The **Reduced cost** for a hyperplane at a corner is the cost of moving one unit away from the plane along its corresponding edge.



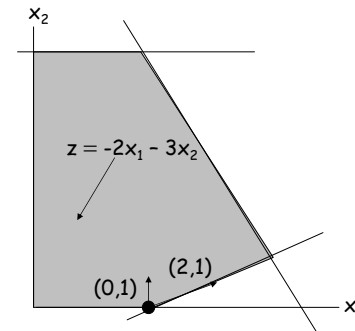
For **minimization**, if all reduced cost are non-negative, then we are at an optimal solution.

Finding the most negative reduced cost is a heuristic for choosing an edge to leave on

15-853

Page17

Reduced cost example



In the example the reduced cost of leaving the plane x_1 is $(-2, -3) \cdot (2, 1) = -7$ since moving one unit off of x_1 will move us $(2, 1)$ units along the edge. We take the dot product of this and the cost function.

15-853

Page18

Simplex Algorithm

1. Find a **corner of the feasible region**

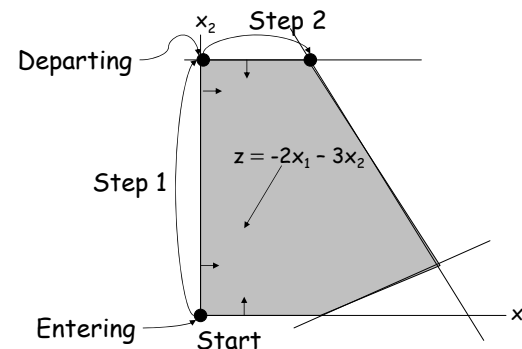
2. **Repeat**

- A. For each of the n hyperplanes intersecting at the corner, calculate its **reduced cost**
- B. If they are all non-negative, then **done**
- C. Else, pick the most negative reduced cost
This is called the **entering** plane
- D. Move along corresponding edge (i.e. leave that hyperplane) until we reach the next corner (i.e. reach another hyperplane)
The new plane is called the **departing** plane

15-853

Page19

Example



15-853

Page20

Simplifying

Problem:

- The $Ax \leq b$ constraints not symmetric with the $x \geq 0$ constraints.
We would like more symmetry.

Idea:

- Make all inequalities of the form $x \geq 0$.
Use "slack variables" to do this.

Convert into form:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array}$$

15-853

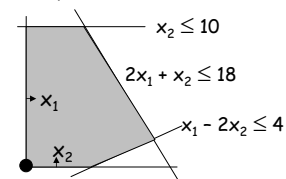
Page21

Standard and Slack Form

Standard Form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b \\ & x \geq 0 \end{array}$$

$|A| = m \times n$
i.e. m equations, n variables

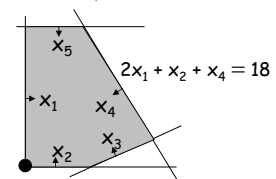


15-853

Slack Form

$$\begin{array}{ll} \text{minimize} & c^T x' \\ \text{subject to} & A'x' = b \\ & x' \geq 0 \end{array}$$

$|A'| = m \times (m+n)$
i.e. m equations, $m+n$ variables



Page22

Example, again

minimize:

$$z = -2x_1 - 3x_2$$

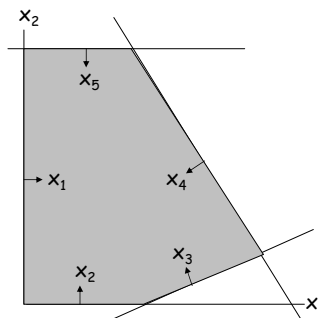
subject to:

$$x_1 - 2x_2 + x_3 = 4$$

$$2x_1 + x_2 + x_4 = 18$$

$$x_2 + x_5 = 10$$

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



The equality constraints impose a 2d plane embedded in 5d space, looking at the plane gives the figure above

15-853

Page23

Using Matrices

If before adding the slack variables A has size $m \times n$
then after it has size $m \times (n + m)$
 m can be larger or smaller than n

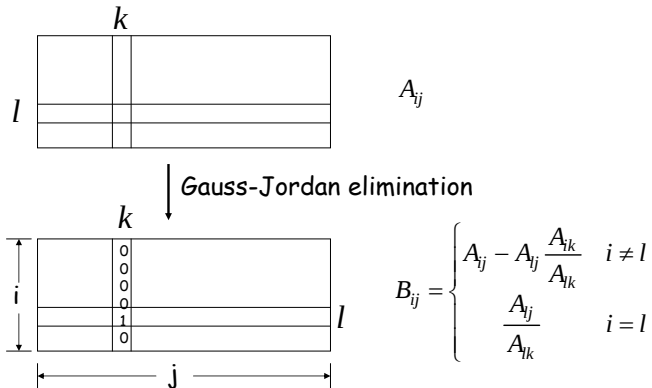
$$A = \begin{array}{c|c} \begin{array}{ccc} \leftarrow n & \leftarrow m & \\ \hline 1 & 0 & 0 \dots \\ 0 & 1 & 0 \dots \\ 0 & 0 & 1 \dots \\ \dots & & \end{array} & \begin{array}{c} \hline \\ \hline \\ \hline \\ \hline \end{array} \\ \hline \begin{array}{c} \leftarrow \text{slack vrs.} \end{array} \end{array} \quad \begin{array}{c} m \\ m \\ m \\ m \end{array}$$

Assuming rows are independent, the solution space of $Ax = b$ is a n dimensional subspace.

15-853

Page24

Gauss-Jordan Elimination



15-853

Page25

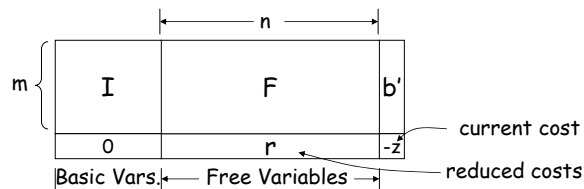
Simplex Algorithm, again

1. Find a **corner of the feasible region**
2. **Repeat**
 - A. For each of the n hyperplanes intersecting at the corner, calculate its **reduced cost**
 - B. If they are all non-negative, then **done**
 - C. Else, pick the most negative reduced cost. This is called the **entering plane**
 - D. Move along corresponding line (i.e. leave that hyperplane) until we reach the next corner (i.e. reach another hyperplane). The new plane is called the **departing plane**

15-853

Page26

Simplex Algorithm (Tableau Method)



This form is called a **Basic Solution**

- the n "**free**" variables are set to 0
- the m "**basic**" variables are set to b'

A valid solution to $Ax = b$ if reached using Gaussian Elimination

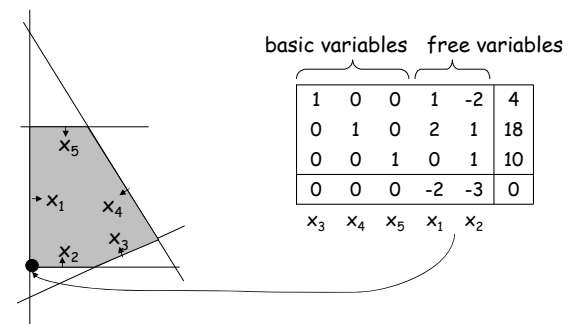
Represents n intersecting hyperplanes

If feasible (i.e. $b' \geq 0$), then the solution is called

15-853

Page27

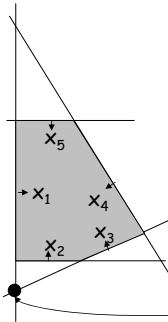
Corner



15-853

Page28

Corner



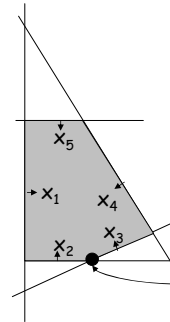
1	0	0	-.5	-1	-2
0	1	0	2.5	1	20
0	0	1	.5	1	12
0	0	0	-3.5	-3	-6

$x_2 \quad x_4 \quad x_5 \quad x_1 \quad x_3$

15-853

Page29

Corner



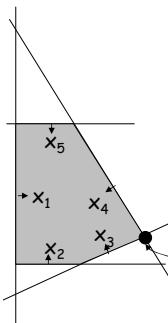
1	0	0	1	-2	4
0	1	0	-2	5	10
0	0	1	0	1	10
0	0	0	2	-7	8

$x_1 \quad x_4 \quad x_5 \quad x_3 \quad x_2$

15-853

Page30

Corner



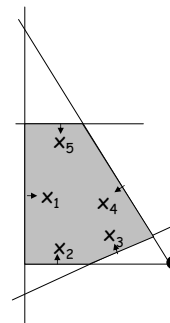
1	0	0	.2	.4	8
0	1	0	-.4	.2	2
0	0	1	.4	-.2	8
0	0	0	-.8	1.4	22

$x_1 \quad x_2 \quad x_5 \quad x_3 \quad x_4$

15-853

Page31

Corner



1	0	0	.5	-2.5	-5
0	1	0	.5	.5	9
0	0	1	0	1	10
0	0	0	1	-2	18

$x_3 \quad x_1 \quad x_5 \quad x_4 \quad x_2$

Note that in general there are $n+m$ choose m corners

15-853

Page32

Simplex Method Again

Once you have found a basic feasible solution (a corner), we can move from corner to corner by swapping columns and eliminating.

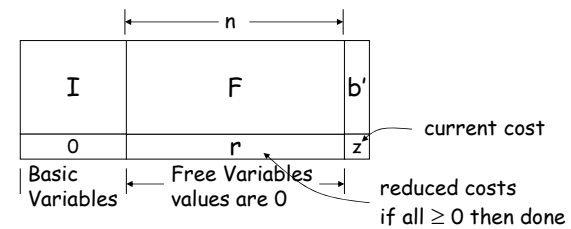
ALGORITHM

1. Find a **basic feasible solution**
2. **Repeat**
 - A. If r (reduced cost) ≥ 0 , DONE
 - B. Else, pick column with most negative r
 - C. Pick row with least positive b' /(selected column)
 - D. Swap columns
 - E. Use Gaussian elimination to restore form

Page33

Tableau Method

- A. If r are all non-negative then **done**

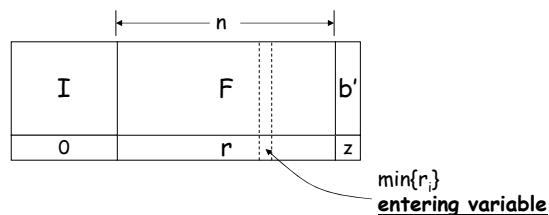


15-853

Page34

Tableau Method

- B. Else, pick the most negative reduced cost
This is called the **entering** plane

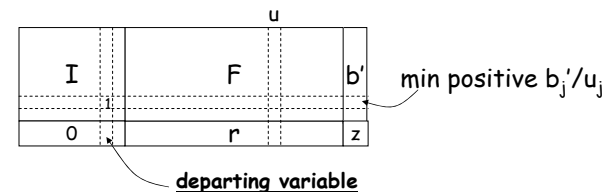


15-853

Page35

Tableau Method

- C. Move along corresponding line (i.e. leave that hyperplane) until we reach the next corner (i.e. reach another hyperplane)
The new plane is called the **departing** plane

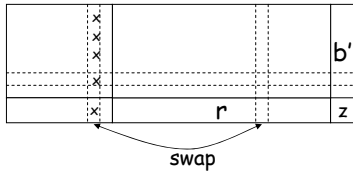


15-853

Page36

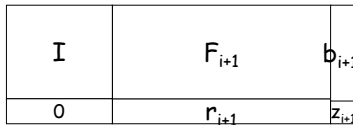
Tableau Method

D. Swap columns



No longer in proper form

E. Gauss-Jordan elimination



Back to proper form

15-853

Page37

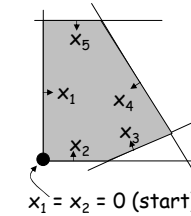
Example

x_1	x_2	x_3	x_4	x_5	
1	-2	1	0	0	4
2	1	0	1	0	18
0	1	0	0	1	10
-2	-3	0	0	0	0

$$\begin{aligned} x_1 - 2x_2 + x_3 &= 4 \\ 2x_1 + x_2 + x_4 &= 18 \\ x_2 + x_5 &= 10 \\ z &= -2x_1 - 3x_2 \end{aligned}$$

Find corner

x_3	x_4	x_5	x_1	x_2	
1	0	0	1	-2	4
0	1	0	2	1	18
0	0	1	0	1	10
0	0	0	-2	-3	0



15-853

Page38

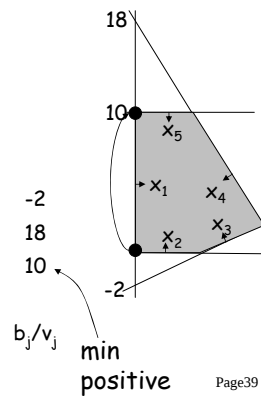
Example

1	0	0	1	-2	4
0	1	0	2	1	18
0	0	1	0	1	10
0	0	0	-2	-3	0

$x_3 \ x_4 \ x_5 \ x_1 \ x_2$

1	0	0	1	-2	4
0	1	0	2	1	18
0	0	1	0	1	10
0	0	0	-2	-3	0

$x_3 \ x_4 \ x_5 \ x_1 \ x_2$



15-853

Page39

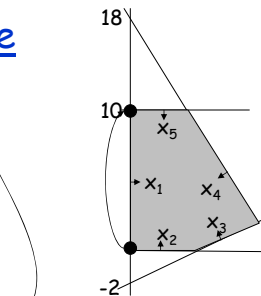
Example

1	0	-2	1	0	4
0	1	1	2	0	18
0	0	1	0	1	10
0	0	-3	-2	0	0

$x_3 \ x_4 \ x_2 \ x_1 \ x_5$

1	0	0	1	2	24
0	1	0	2	-1	8
0	0	1	0	1	10
0	0	0	-2	3	30

$x_3 \ x_4 \ x_2 \ x_1 \ x_5$



Gauss-Jordan Elimination

15-853

Page40

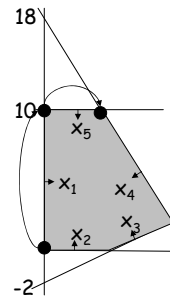
Example

1	0	0	1	2	24
0	1	0	2	-1	8
0	0	1	0	1	10
0	0	0	-2	3	30

$x_3 \quad x_4 \quad x_2 \quad x_1 \quad x_5$

1	0	0	1	2	24	24
0	1	0	2	-1	8	4
0	0	1	0	1	10	1
0	0	0	-2	3	30	

$x_3 \quad x_4 \quad x_2 \quad x_1 \quad x_5$



15-853

Page41

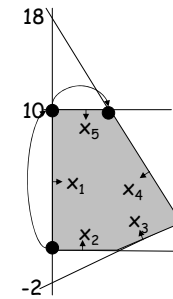
Example

1	1	0	0	2	24
0	2	0	1	-1	8
0	0	1	0	1	10
0	-2	0	0	3	30

$x_3 \quad x_1 \quad x_2 \quad x_4 \quad x_5$

1	0	0	-0.5	2.5	20
0	1	0	.5	-0.5	4
0	0	1	0	1	10
0	0	0	1	2	38

$x_3 \quad x_1 \quad x_2 \quad x_4 \quad x_5$



Gauss-Jordan Elimination

15-853

Page42

Simplex Concluding remarks

For dense matrices, takes $O(n(n+m))$ time per iteration

Can take an **exponential** number of iterations.

In practice, sparse methods are used for the iterations.

15-853

Page43

Duality

Primal (P):

maximize $z = c^T x$

subject to $Ax \leq b$

$x \geq 0$ (n equations, m variables)

Dual (D):

minimize $z = y^T b$

subject to $A^T y \geq c$

$y \geq 0$ (m equations, n variables)

Duality Theorem: if x is feasible for **P** and y is feasible for **D**, then $cx \leq yb$ and at optimality $cx = yb$.

15-853

Page44

Duality (cont.)

Optimal solution for both

feasible solutions for Dual (maximization) feasible solutions for Primal (minimization)

Quite similar to duality of Maximum Flow and Minimum Cut.

Useful in many situations.

15-853

Page45

Duality Example

Primal:

maximize:

$$z = 2x_1 + 3x_2$$

subject to:

$$x_1 - 2x_2 \leq 4$$

$$2x_1 + x_2 \leq 18$$

$$x_2 \leq 10$$

$$x_1, x_2 \geq 0$$

Dual:

minimize:

$$z = 4y_1 + 18y_2 + 10y_3$$

subject to:

$$y_1 + 2y_2 \geq 2$$

$$-2y_1 + y_2 + y_3 \geq 3$$

$$y_1, y_2, y_3 \geq 0$$

Solution to both is 38 ($x_1=4, x_2=10$), ($y_1=0, y_2=1, y_3=2$)

15-853

Page46