15.053 Thursday, April 4

- Introduction to Integer Programming
  - Integer programming models

Handouts: Lecture Notes

A 2-Variable Integer program

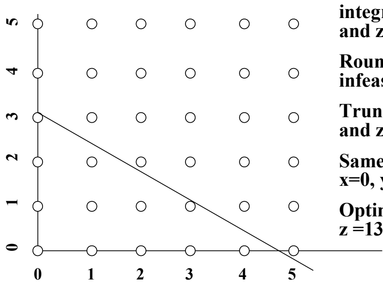
maximize $3x + 4y$

subject to $5x + 8y \leq 24$

$x, y \geq 0$ and integer

- What is the optimal solution?

The Feasible Region



Solve LP (ignore integrality) get x=24/5, y=0 and z =14 2/5.

Round, get x=5, y=0, infeasible!

Truncate, get x=4, y=0, and z =12

Same solution value at x=0, y=3.

Optimal is x=3, y=1, and z =13

Why integer programs?

- Advantages of restricting variables to take on integer values
  - More realistic
  - More flexibility
- Disadvantages
  - More difficult to model
  - Can be <u>much</u> more difficult to solve

On 0-1 variables

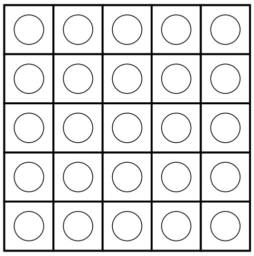
- Integer programs: linear equalities and inequalities plus constraints that say a variable must be integer valued.
- We also permit "$x_j \in \{0,1\}$." This is equivalent to

  $0 \leq x_j \leq 1$ and x_j integer.

The mystery of integer programming

- Some integer programs are easy (we can solve problems with millions of variables)
- Some integer programs are hard (even 100 variables can be challenging)
- It takes expertise and experience to know which is which
- It's an active area of research at MIT and elsewhere

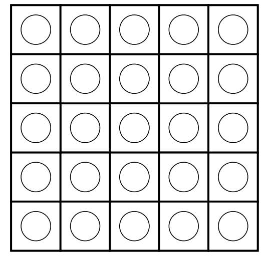
The game of fiver.



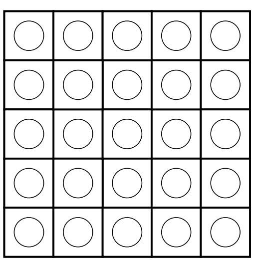
Click on a circle, and flip its color and that of adjacent colors.

Can you make all of the circles red?

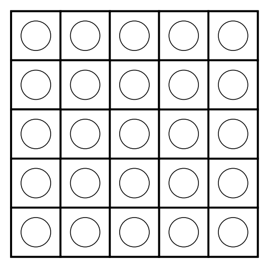
The game of fiver.



The game of fiver.

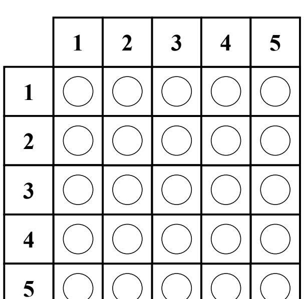


The game of fiver.



Let's write an optimization problem whose solution solves the problem in the fewest moves.

Optimizing the game of fiver.



Let x(i,j) = 1 if I click on the square in row i and column j.

x(i,j) = 0 otherwise.

Focus on the element in row 3, and column 2. To turn it red, we require that

x(2,2) + x(3,1) + x(3,2) + x(3,3) + x(4,2) is odd

Optimizing the game of fiver

- **(i, j) to be red for i = 1 to 5 and for j = 1 to 5**
- **We want to minimize the number of moves.**

Minimize $\sum_{i,j=1 \text{ to } 5} x(i,j)$

Subject to x(i, j) + x(i, j-1) + x(i, j+1) + x(i-1, j) + x(i+1, j) is odd

for i = 1 to 5, j = 1 to 5

x(i, j) is 0 or 1 for i = 1 to 5 and j = 1 to 5

x(i, j) = 0 otherwise.

- **This (with a little modification) is an integer program.**

Optimizing the game of fiver

- (i, j) to be red for i = 1 to 5 and for j = 1 to 5
- We want to minimize the number of moves.

Minimize $\Sigma_{i,j=1 \text{ to } 5}$ x(i,j)

Subject to x(i, j) + x(i, j-1) + x(i, j+1) + x(i-1, j) + x(i+1, j) - 2y(i,j) = 1

for i = 1 to 5, j = 1 to 5

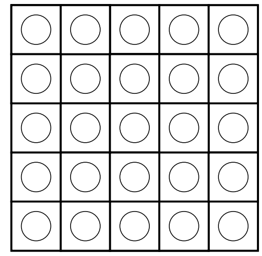
x(i, j) is 0 or 1 for i = 1 to 5 and j = 1 to 5

y(i,j) is integral

x(i, j) = 0 otherwise.

- This is an integer program.

Should I give away the solution?



Types of integer programs

- All integer programs have linear equalities and inequalities and some or all of the variables are required to be integer.
  - If all variables are required to be integer, then it is usually called a <u>pure integer program</u>.
  - If all variables are required to be 0 or 1, it is called a <u>binary integer program</u>, or a <u>0-1 integer program</u>.
  - if some variables can be fractional and others are required to be integers, it is called a <u>mixed linear integer program (MILP)</u>

Stockco Example

Stockco is considering 6 investments. The cash required from each investment as well as the NPV of the investment is given next. The cash available for the investments is $14,000. Stockco wants to maximize its NPV. What is the optimal strategy?

An investment can be selected or not. One cannot select a fraction of an investment.

Data for the Stockco Problem

Investment budget = $14,000

Investment	1	2	3	4	5	6
Cash Required (1000s)	$5	$7	$4	$3	$4	$6
NPV added (1000s)	$16	$22	$12	$8	$11	$19

Integer Programming Formulation

- What are the decision variables?

$$x_i = \begin{cases} 1, & \text{if we invest in } i = 1,\ldots,6, \\ 0, & \text{else} \end{cases}$$

- Objective and Constraints?

$$Max\ 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6$$

$$5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14$$

$$x_j \in \{0,1\} \text{ for each } j = 1 \text{ to } 6$$

Possible constraints in integer programs

- The previous constraints represent "economic indivisibilities", either a project is selected, or it is not. There is no selecting of a fraction of a project.
- Similarly, integer variables can model logical requirements (e.g., if stock 2 is selected, then so is stock 1.)

How to model "logical" constraints

- Exactly 3 stocks are selected.
- If stock 2 is selected, then so is stock 1.
- If stock 1 is selected, then stock 3 is not selected.
- Either stock 4 is selected or stock 5 is selected, but not both.

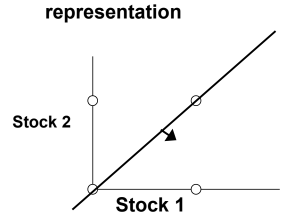
Formulating Constraints

- Exactly 3 stocks are selected

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 3$$

If stock 2 is selected then so is stock 1

A 2-dimensional representation



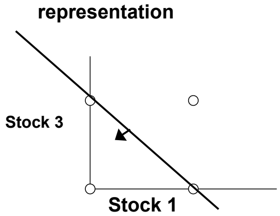
The integer programming constraint:

$$x_1 \geq x_2$$

Work with your partner for 5 minutes trying to model the other constraints.

If stock 1 is selected then stock 3 is not selected

A 2-dimensional representation

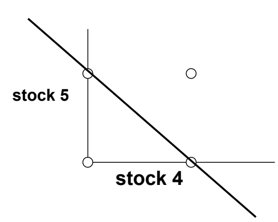


The integer programming constraint:

$$x_1 + x_3 \leq 1$$

Either stock 4 is selected or stock 5 is selected, but not both.

A 2-dimensional representation



The integer programming constraint:

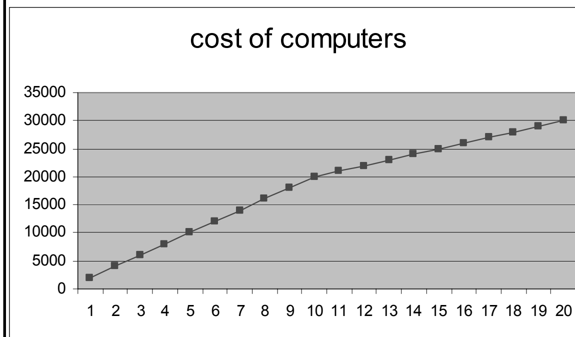
$$x_4 + x_5 = 1$$

Representing Non-linear functions

- Suppose that the cost of computers is as follows:
  - \$2000 each if you buy 1 to 10
  - \$1000 for each computer over 10
  - Suppose that at most 30 computers will be purchased

  - Let the number of computers bought be $x + y$
  - where $0 \leq x \leq 10$, and $y \geq 0$ only if $x = 10$.

  - cost is \$2000 x + \$1000 y.



Formulating using integer programming

create a variable so that $w = 1$ if $x = 10$.

cost is $2000 x + 1000 y$

subject to $0 \leq x \leq 10$

$0 \leq y$

$w \leq x/10$

$y \leq 20 w$

w binary, $x, y \geq 0$ integer

Warehouse location problem

- n warehouses
  - cost f_i of opening warehouse i
- m customers
  - customer j has a "demand" of d_j
  - unit shipping cost c_{ij} of serving customer i via warehouse j.

- Variables:
  - let $y_j = 1$ if warehouse j is opened
  - Let x_{ij} = amount of demand for customer i satisfied via warehouse j.

Suppose you knew which warehouses were open. S = set of open warehouses

- x_{ij} = demand satisfied for customer i at warehouse j
- $y_j = 1$ for j in S, $y_j = 0$ for j not in S.

minimize $\sum_{i,j} c_{ij} x_{ij} + \sum_{j \in S} f_j$

subject to:

- customers get their demand satisfied

$\sum_i x_{ij} = d_j$

- no shipments are made from an empty warehouse

$x_{ij} \leq d_j$ if $y_j = 1$
$x_{ij} = 0$ if $y_j = 0$

and $x \geq 0$

More on warehouse location

- $y_i = 1$ if warehouse i is opened
  $y_i = 0$ otherwise
- x_{ij} = flow from i to j

minimize $\sum_{i,j} c_{ij} x_{ij} + \sum_i f_i y_i$

subject to:

- customers get their demand satisfied

$\sum_i x_{ij} = d_j$

- each warehouse is either opened or it is not (no partial openings)

$0 \leq y_i \leq 1$
y_i integral for all i.

- no shipments are made from an empty warehouse

$x_{ij} \leq d_j y_i$ for all i, j

and $x \geq 0$

Two key aspects of using integrality in the model

- Costs: we include the cost of the warehouse only if it is opened.
  - $\Sigma_i \; f_i y_i$
- Constraints: We do not allow shipping from warehouse j if it is not opened.
  - $x_{ij} \leq d_j y_i$ for all i, j

More on warehouse location

- The above is a core subproblem occurring in supply chain management, and it can be enriched
  - more complex distribution system
  - capacity constraints
  - non-linear transportation costs
  - delivery times
  - multiple products
  - business rules
  - and more

Using Excel Solver to Solve Integer Programs

- Add the integrality constraints (or add that a variable is binary)
- Set the Solver Tolerance. (The tolerance is the percentage deviation from optimality allowed by solver in solving Integer Programs.)
  - The default is 5%
  - The default is way to high
  - It often finds the optimum for small problems

Some Comments on IP models

- There are often multiple ways of modeling the same integer program.
- Solvers for integer programs are ***extremely*** sensitive to the formulation. (not true for LPs)

Example

- constraint A: $2x_1 + 2x_2 + \ldots + 2x_{50} \leq 51$
- constraint B: $x_1 + x_2 + \ldots + x_{50} \leq 25$
  - assume that x is binary
- constraints C: $x_1 \leq y, x_2 \leq y, \ldots, x_{50} \leq y$ (where y is binary)
- constraint D: $x_1 + \ldots + x_{50} \leq 50\, y$

B dominates A, C dominates D
It is not obvious why, until you see the algorithms.

Summary on Integer Programming

- Dramatically improves the modeling capability
  - Economic indivisibilities
  - Logical constraints
  - Modeling nonlinearities
  - classical problems in capital budgeting and in supply chain management
- Not as easy to model
- Not as easy to solve.

The number of stocks selected is not three

Either $x_1+x_2+x_3+x_4+x_5+x_6 \geq 4$ or (1)

$x_1+x_2+x_3+x_4+x_5+x_6 \leq 2$ (2)

Add an auxiliary variable $w \in \{0,1\}$ with the following properties:

- If w = 1, then the first constraint is satisfied (A)
- If w = 0, then the second constraint is satisfied (B)

Since w is 0 or 1, at least one of the two constraints must be satisfied.

The number of stocks selected is not three (cont'd)

Add the constraint:

$x_1+x_2+x_3+x_4+x_5+x_6 \geq 4w$ (A)

So, if w=1 then $x_1+x_2+x_3+x_4+x_5+x_6 \geq 4$ (1)

Note: if w = 0, the first constraint is automatically satisfied.

$x_1+x_2+x_3+x_4+x_5+x_6 \leq 2 + 4w$ (B)

So, if w=0 then $x_1+x_2+x_3+x_4+x_5+x_6 \leq 2$ (2)

Note: if w = 1, the second constraint is automatically satisfied. (If we had written "≤ 2 + 3w", then we would incorrectly have eliminate the solution in which $x_j = 1$ for all j.)

You must select stock 1 unless the NPV of the portfolio exceeds $42,000.

If NPV < 42 then $x_1=1$.

Add the constraint: $x_1 \geq (42 - \text{NPV})/42$.

A larger denominator will also work.

Recall that NPV is

$16x_1+ 22x_2+ 12x_3+ 8x_4+ 11x_5+ 19x_6$

$42x_1 \geq 42 - (16x_1+22x_2+12x_3+8x_4+11x_5+19x_6)$