15.053/8 **March 14, 2013**

Introduction to Integer Programming

– Integer programming models

Quotes of the Day

“Somebody who thinks logically is a nice contrast to the real world.”
-- The Law of Thumb

“Take some more tea,” the March Hare said to Alice, very earnestly.

“I’ve had nothing yet,” Alice replied in an offended tone, “so I can’t take more.”

“You mean you can’t take less,” said the Hatter. “It’s very easy to take more than nothing.”
-- Lewis Carroll in *Alice in Wonderland*

Combinatorial optimization problems

- **INPUT**: A description of the data for an instance of the problem

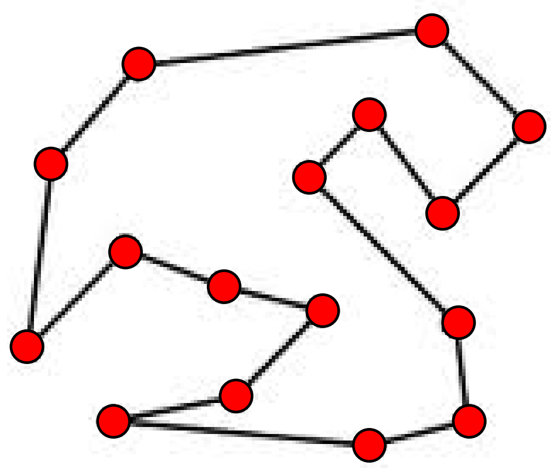
- **FEASIBLE SOLUTIONS**: there is a way of determining from the input whether a given solution x' (assignment of values to decision variables) is feasible. Typically in combinatorial optimization problems there is a finite number of possible solutions.

- **OBJECTIVE FUNCTION**: For each feasible solution x' there is an associated objective f(x').

Minimization problem. Find a feasible solution x* that minimizes f( ) among all feasible solutions.

Example 1: Traveling Salesman Problem

- **INPUT**: a set N of n points in the plane
- **FEASIBLE SOLUTION:** a tour that passes through each point exactly once.
- **OBJECTIVE**: minimize the length of the tour.



Example 2: Balanced Partition

- **INPUT**: A set of positive integers $a_1, \ldots, a_n$
- **FEASIBLE SOLUTION**: a partition of {1, 2, … n} into two disjoint sets S and T.
  - $S \cap T = \emptyset$, $S \cup T = \{1, \ldots, n\}$
- **OBJECTIVE** : minimize $\left| \sum_{i \in S} a_i - \sum_{i \in T} a_i \right|$

Example: 7, 10, 13, 17, 20, 22
These numbers sum to 89

The best split is {10, 13, 22} and {7, 17, 20}.

Example 3. Exam Scheduling

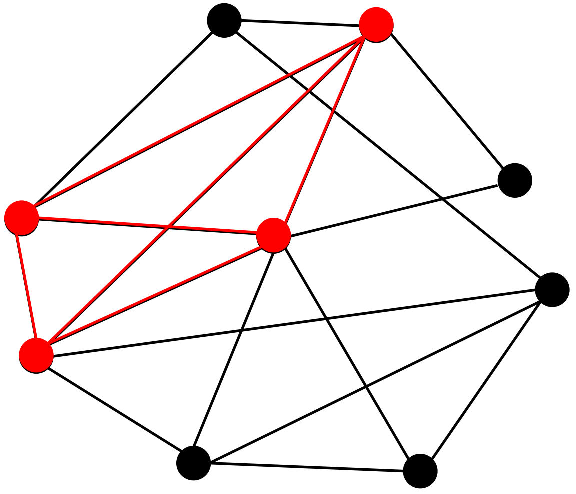
- **INPUT**: a list of subjects with a final exam; class lists for each of these subjects, and a list of times that final exams can be scheduled.
  Let a_{ij} denote the number of students that are taking subjects i and j.

- **FEASIBLE SOLUTION**: An assignment of subjects to exam periods

- **OBJECTIVE**: minimize
  $\sum \{a_{ij} : i \text{ and } j \text{ are scheduled at the same time}\}$

Example 4: Maximum Clique Problem

- **INPUT**: a friendship network G = (N, A). If persons i and j are friends, then (i, j) ∈ A.
- **FEASIBLE SOLUTION**: a set S of people such that every pair of persons in S are friends.
- **OBJECTIVE**: maximize |S|



Example 5: Integer programming

- **INPUT**: a set of variables $x_1, \ldots, x_n$ and a set of linear inequalities and equalities, and a subset of variables that is required to be integer.
- **FEASIBLE SOLUTION**: a solution x' that satisfies all of the inequalities and equalities as well as the integrality requirements.
- **OBJECTIVE**: maximize $\sum_i c_i x_i$

Example: maximize $3x + 4y$

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

$x, y \geq 0$ and integer

Which of the following is false?

1. The Traveling Salesman Problem is a combinatorial optimization problem.
2. Integer Programming is a combinatorial optimization problem.
3. Every instance of a combinatorial optimization problem has data, a method for determining which solutions are feasible, and an objective function value for each feasible solution.
4. Warren G. Harding was the greatest American President.

The advantages of integer programs

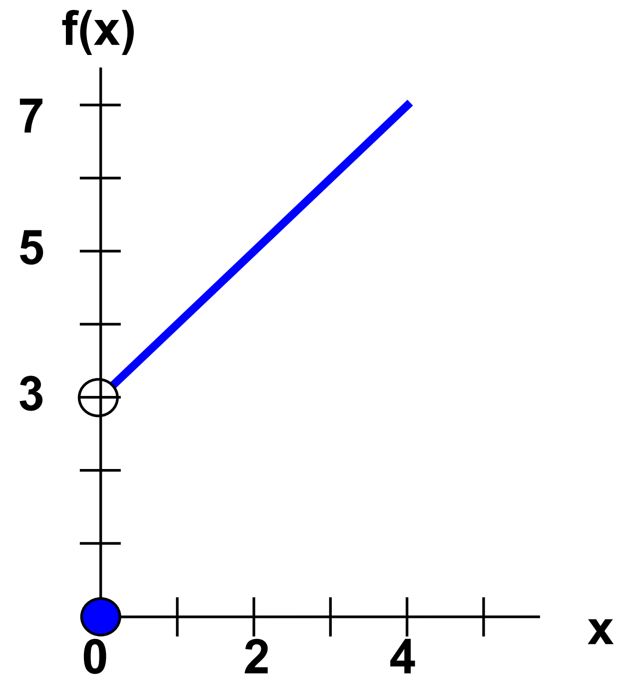
- **Rule of thumb: integer programming can model any of the variables and constraints that you really want to put into an LP, but can't.**
- **Binary variables**
  - **x_i = 1 if we decide to do project i (else, it is 0)**
  - **x_i = 1 if node i is selected in the graph (else 0)**
  - **x_{ij} = 1 if we carry out task j after task i (else, 0)**
  - **x_{it} = 1 if we take subject i in semester t (else, 0)**

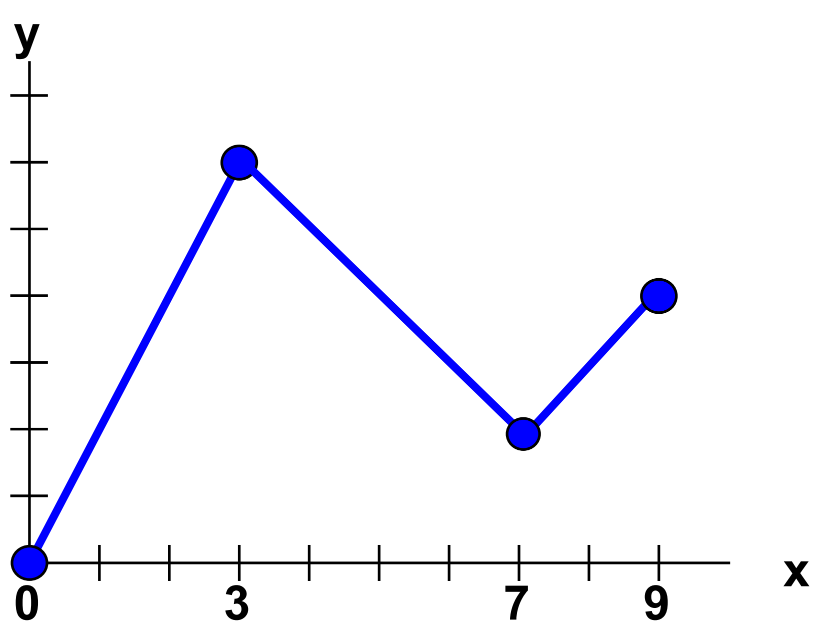
Examples of constraints

- If project i is selected, then project j is not selected.
- If $x_1 > 0$, then $x_1 \geq 10$.
- $x_3 \geq 5$ or $x_4 \geq 8$.
- x_1, x_2, x_3, x_4, x_5, are all different integers in {1, 2, 3, 4, 5}
- x is divisible by 7
- x is either 1 or 2 or 4 or 8 or 32

Nonlinear functions can be modeled using integer programming

$$f(x) = \begin{cases} 3 + x & \text{if } x > 0 \\ 0 & \text{if } x = 0 \end{cases}$$



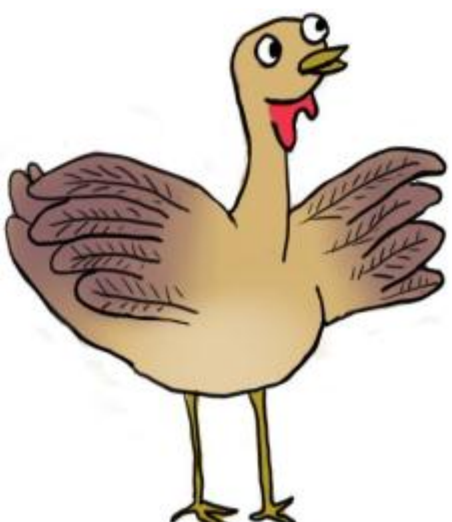


y = 2 x   if $0 \leq x \leq 3$

y = 9 – x   if $3 \leq x \leq 7$

y = -5 + x   if $7 \leq x \leq 9$

You mean, that you can write all of those constraints in an integer program. That’s so easy.



No. That’s not what we mean! We mean that we can take any of these constraints, and there is a way of creating integer programming constraints that are mathematically equivalent. It’s not so easy at first, but it gets easier after you see some examples.

We’ll show you how to do this one step at a time. But first, we’ll review what we mean by integer programs.

Integer Programs

Integer programs: a linear program plus the additional constraints that some or all of the variables must be integer valued.

We also permit "$x_j \in \{0,1\}$," or equivalently, "x_j is binary"

This is a shortcut for writing the constraints:

$$0 \le x_j \le 1 \text{ and } x_j \text{ integer}.$$

Types of Integer Programs

Mixed integer linear programs (MILPs or MIPs)

$x_j \geq 0$ and integer for some or all j.

Pure Integer Programs

$x_j \geq 0$ and integer for every j.

0-1 Integer Programs

$x_j \in \{0,1\}$ for every j.

Note, pure integer programming instances that are unbounded can have an infinite number of solutions. But they have a finite number of solutions if the variables are bounded.

Goals of lectures on Integer Programming.

- Lectures 1 and 2
  - Introduce integer programming
  - Techniques (or tricks) for formulating combinatorial optimization problems as IPs
- Lectures 3 and 4.
  - How integer programs are solved (and why they are hard to solve).
    - Rely on solving LPs fast
    - Branch and bound and cutting planes
- Lecture 5. Review and modeling practice

A 2-Variable Integer program

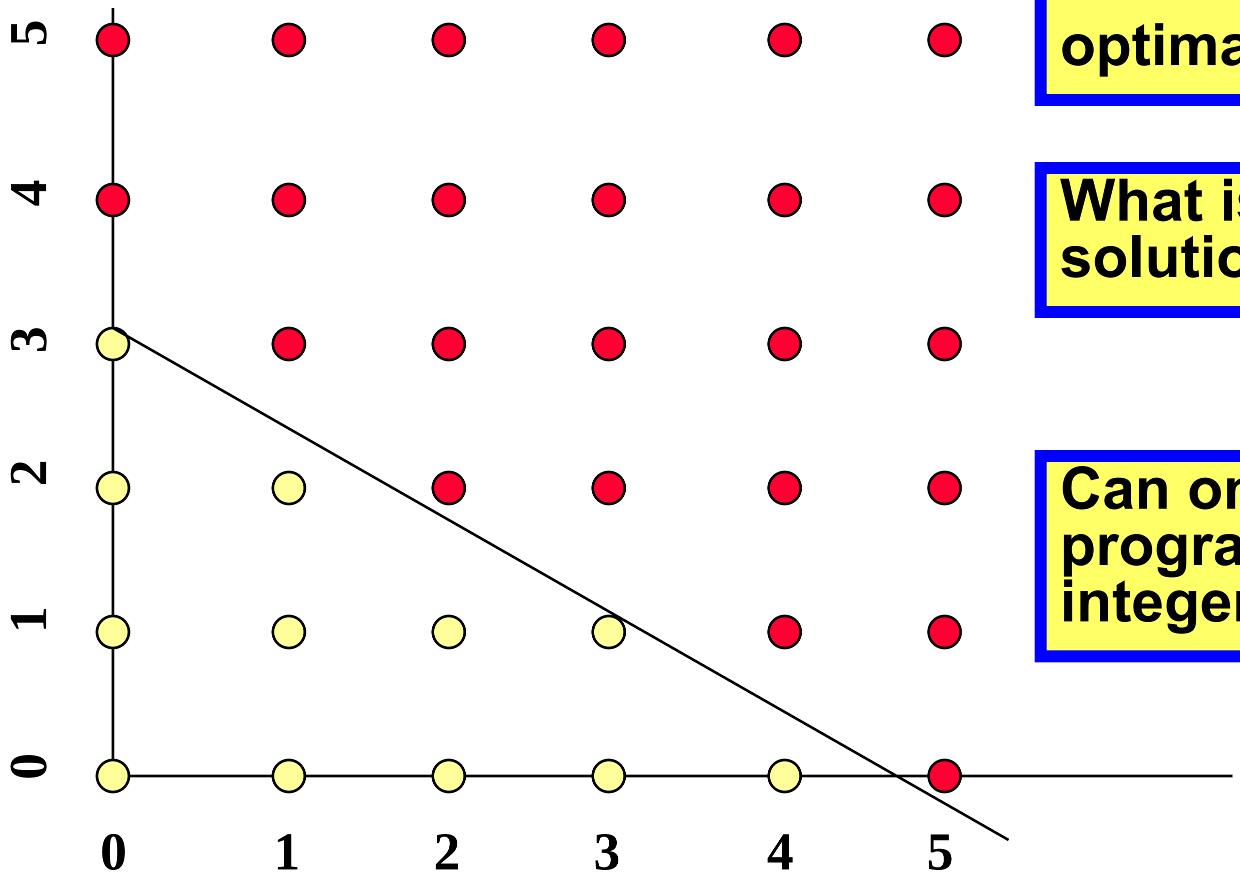
maximize $3x + 4y$

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

$x, y \geq 0$ and integer

- What is the optimal solution?

The Feasible Region

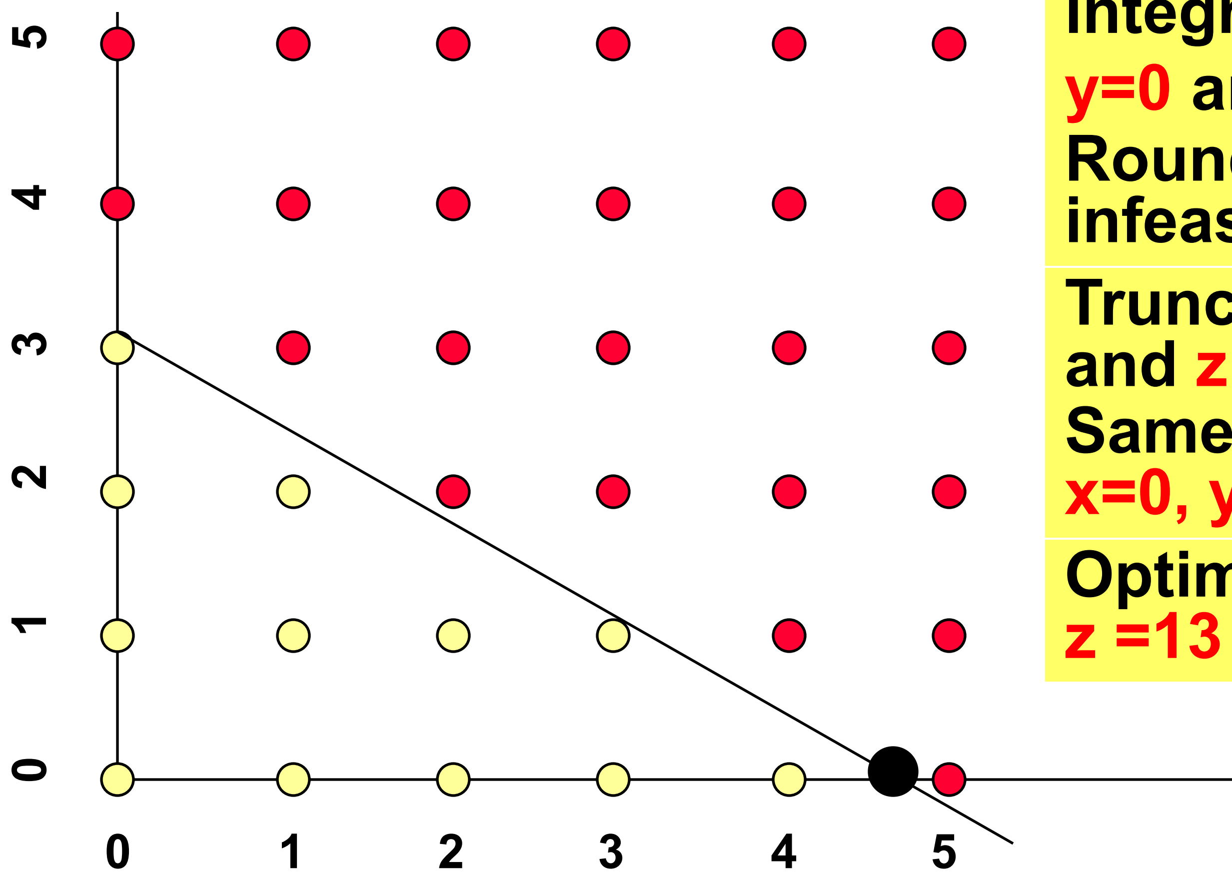


Question: What is the optimal integer solution?

What is the optimal linear solution?

Can one use linear programming to solve the integer program?

A rounding technique that sometimes is useful, and sometimes not.

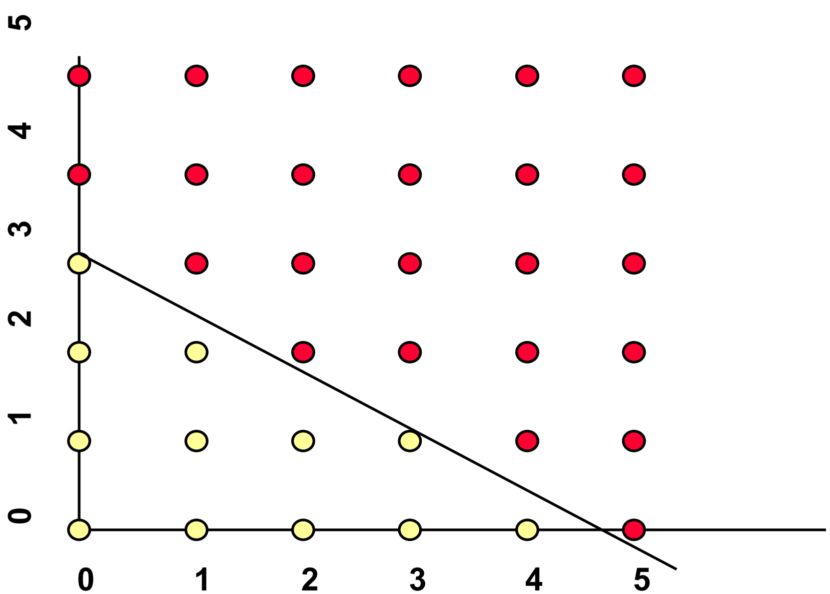


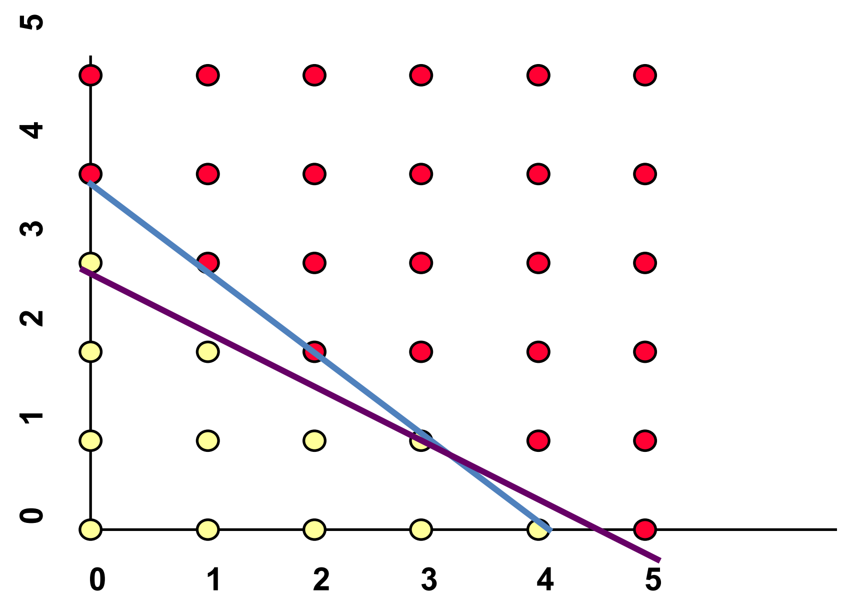
Solve LP (ignore integrality) get x=24/5, y=0 and z =14 2/5.
Round, get x=5, y=0, infeasible!

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

Consider the feasible regions for the two integer programs on this slide.





max $3x + 4y$

s.t. $5x + 8y \leq 24$

$x, y \geq 0$ and integer

max $3x + 4y$

s.t. $x + y \leq 4$

$2x + 3y \leq 9$

$x, y \geq 0$ and integer

Which of the following is false for the two integer programs on the previous slide?

1. The two models are the same in that they have the same feasible regions and the same objective function.

2. Model 1 will be solved faster because it has fewer constraints.

3. If we removed the integrality constraints from both models, they would become two different linear programs.

4. Model 1 has the fewest number of constraints for an IP with this feasible region.

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**

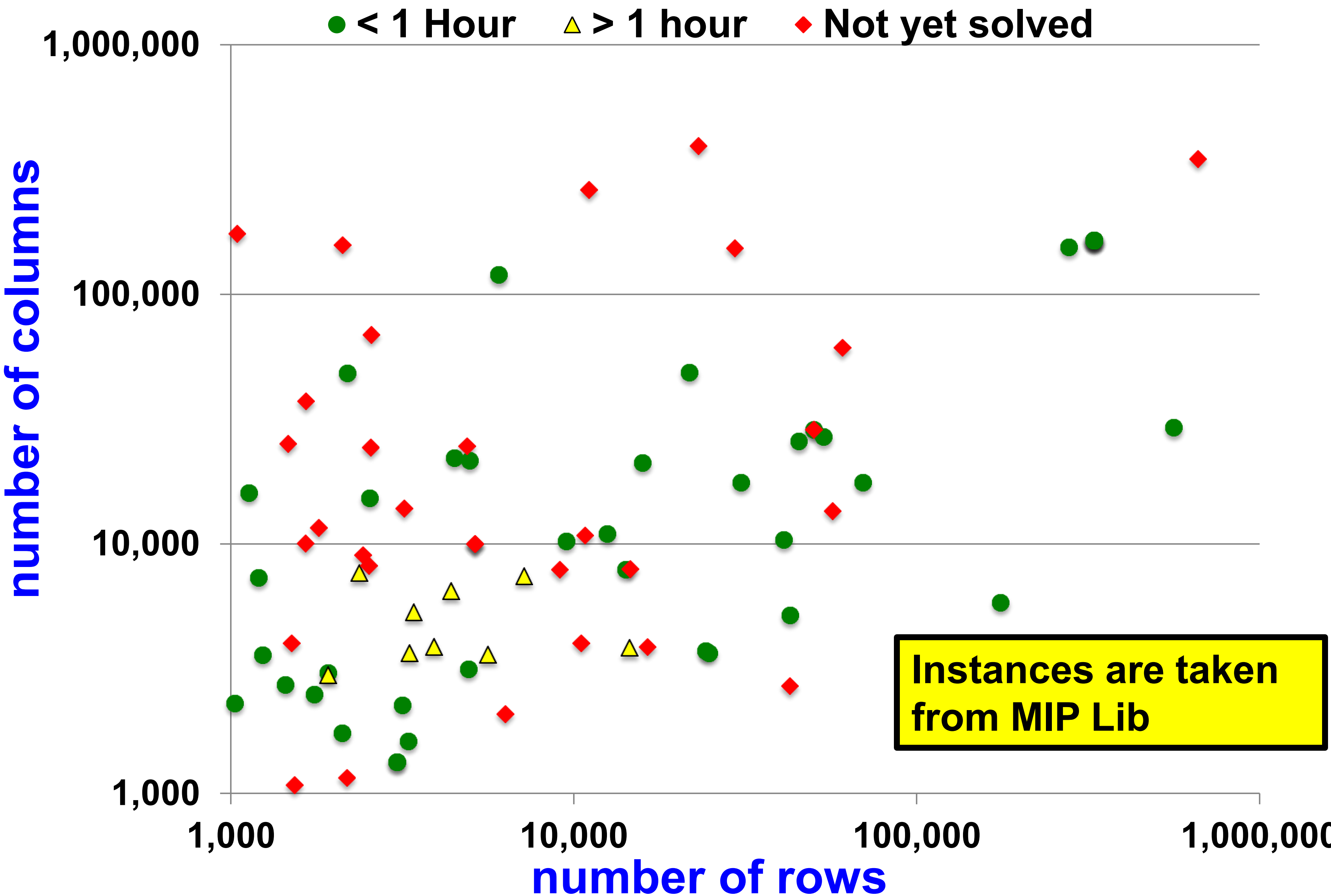
On computation for IPs

- **Much, much harder than solving LPs**

- **Very good solvers can solve large problems**
  - **e.g., 50,000 columns   2 million non-zeros**

- **Hard to predict what will be solved quickly and what will take a long time.**

Running time to optimality (CPLEX)



Mental Break

On formulating integer programs

Consider an instance of a combinatorial optimization problem (COP).

When we form the integer program (IP), we usually want the following:

1. If x is feasible for the COP, then x is feasible for the IP.
2. If x is feasible for the IP, then x is feasible for the COP.
3. If x is feasible, then its objective function value is the same for both the IP and COP.

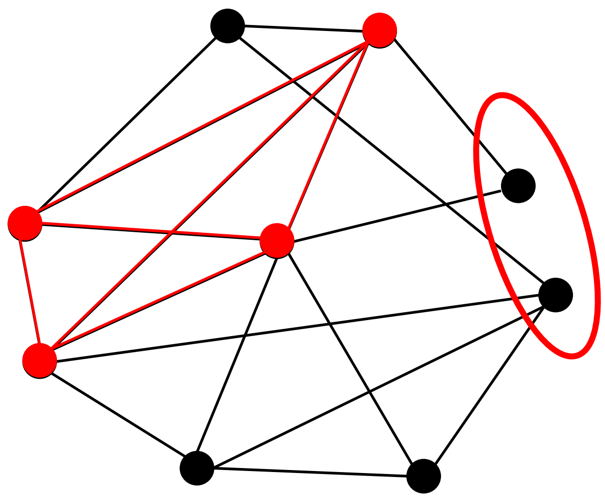
Note: We often need to add variables to the COP (especially 0-1 variables), when formulating integer programs.

Example : Maximum Clique Problem

INPUT: a friendship network G = (N, A). If persons i and j are friends, then (i, j) ∈ A.

FEASIBLE SOLUTION: a set S of people such that every pair of persons in S are friends.

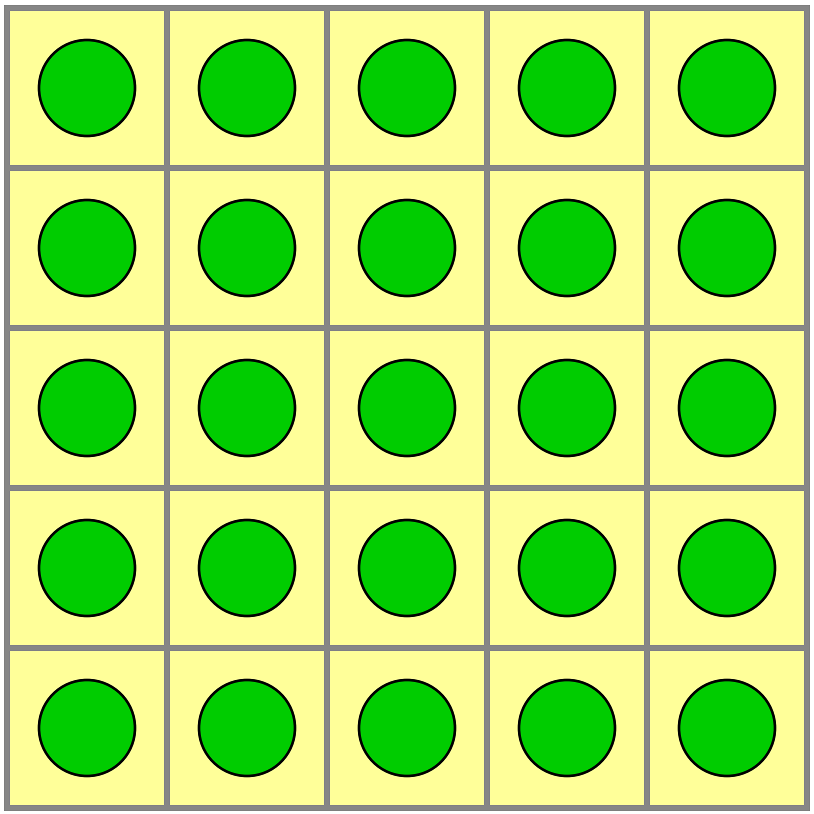
OBJECTIVE: maximize |S|



Decision variables

$$x_i = \begin{cases} 1 & \text{if node } i \text{ is selected.} \\ 0 & \text{otherwise.} \end{cases}$$

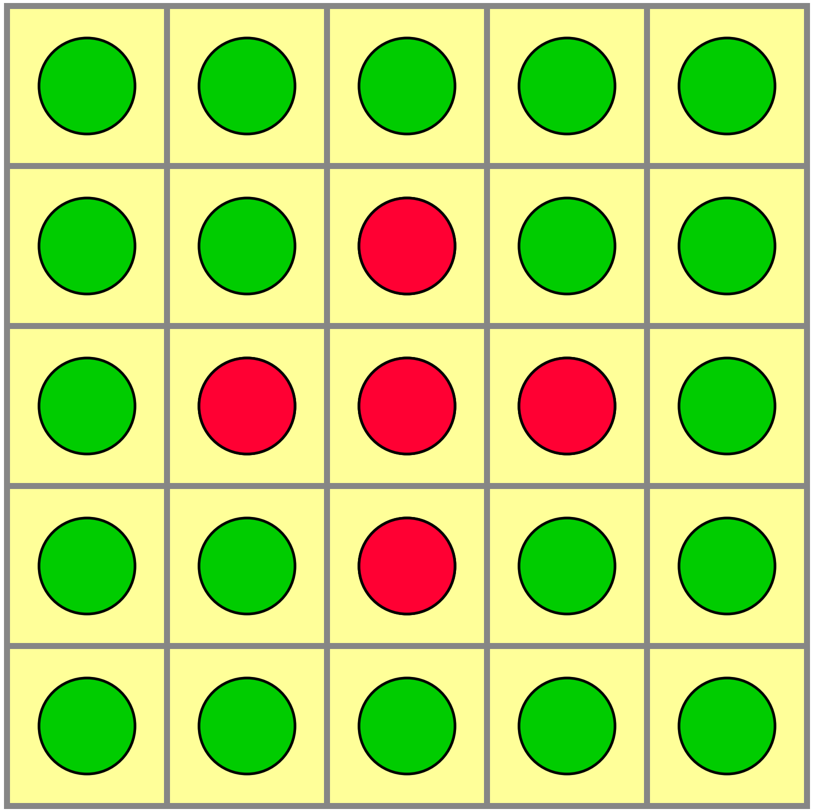
The Game of Fiver.



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

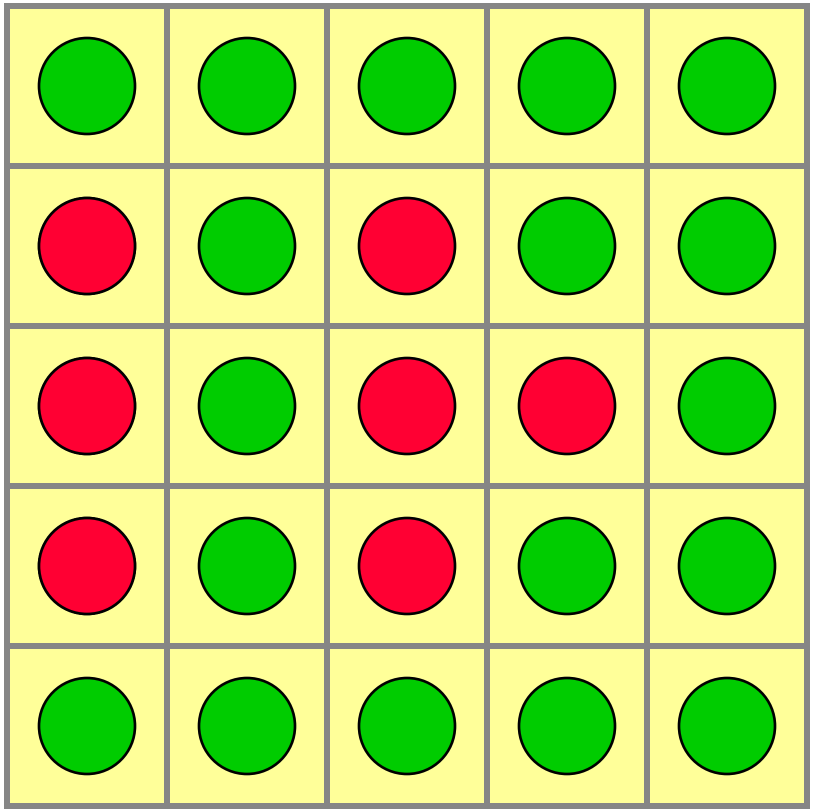
Can you make all of the circles red?

The game of fiver.



Click on (3, 3)

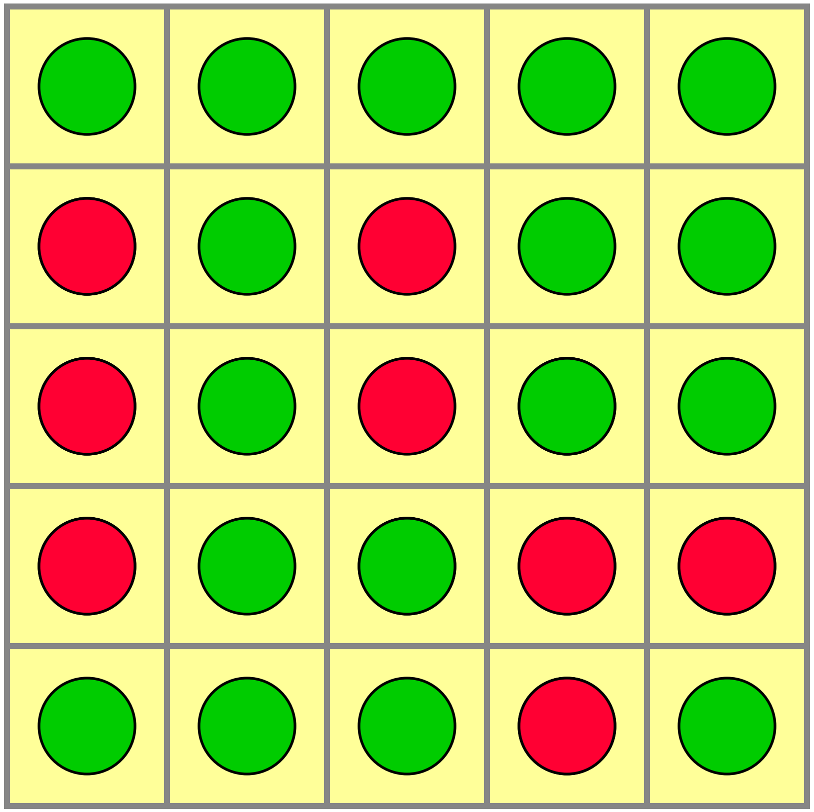
The game of fiver.



Click on (3, 1)

Click on (4, 4)

The game of fiver.



Next: an optimization problem whose solution solves the problem in the fewest moves.

On forming Integer programs

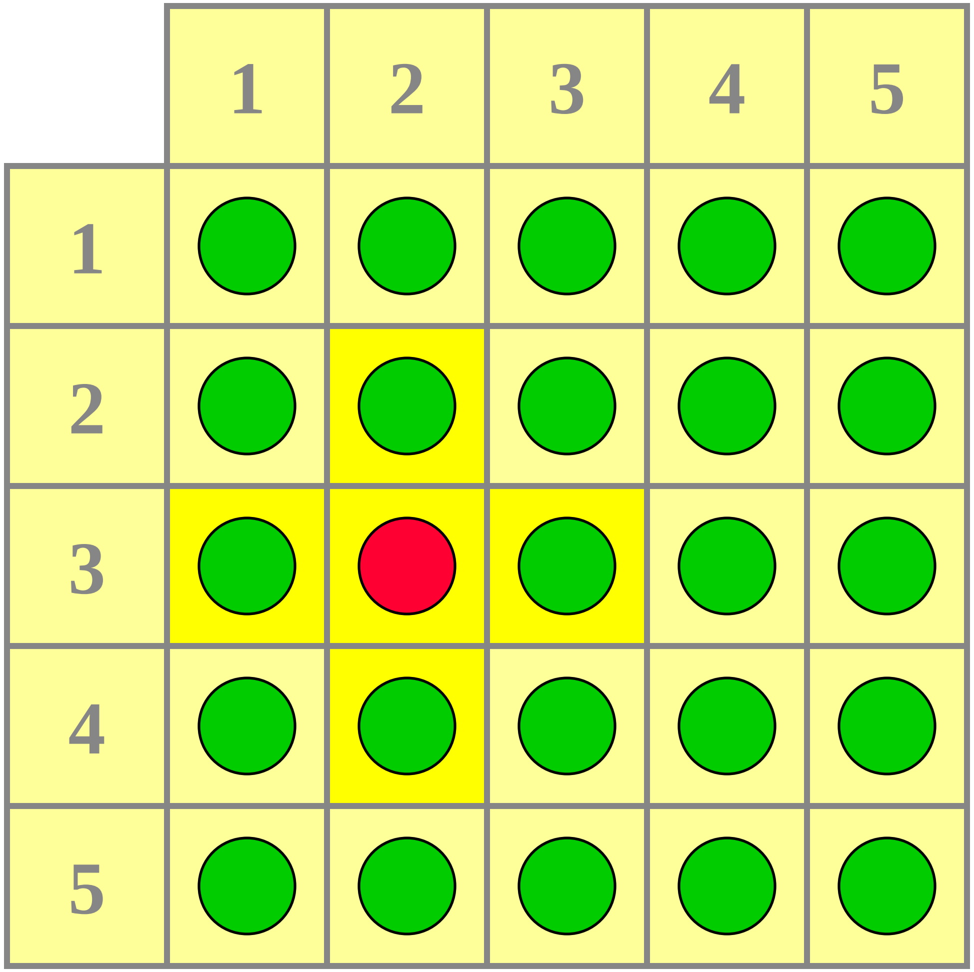
1. First select the set of decision variables.

   It turns out that timing does not matter in this game. All that matters is what square are clicked on.

2. Write the objective.

3. Write the constraints. If it is easier to express it using non-linear constraints, or logical constraints, then do this first.

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.

Let's write the formulation

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

x(i,j) = 0 otherwise.

Optimizing the game of fiver

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

s.t. $x(i, j) + x(i, j-1) + x(i, j+1) + x(i-1, j) + x(i+1, j)$ **is odd for all** $i, j = 1$ **to** 5.

$x(i, j)$ **is 0 or 1 for all** $i, j = 1$ **to** 5.

$x(i, j) = 0$ **otherwise.**

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

Optimizing the game of fiver

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

s.t. $x(i, j) + x(i, j-1) + x(i, j+1) + x(i-1, j) + x(i+1, j) - 2\, y(i, j) = 1$ for all $i, j = 1$ to 5.

$x(i, j)$ is 0 or 1 for all $i, j = 1$ to 5.

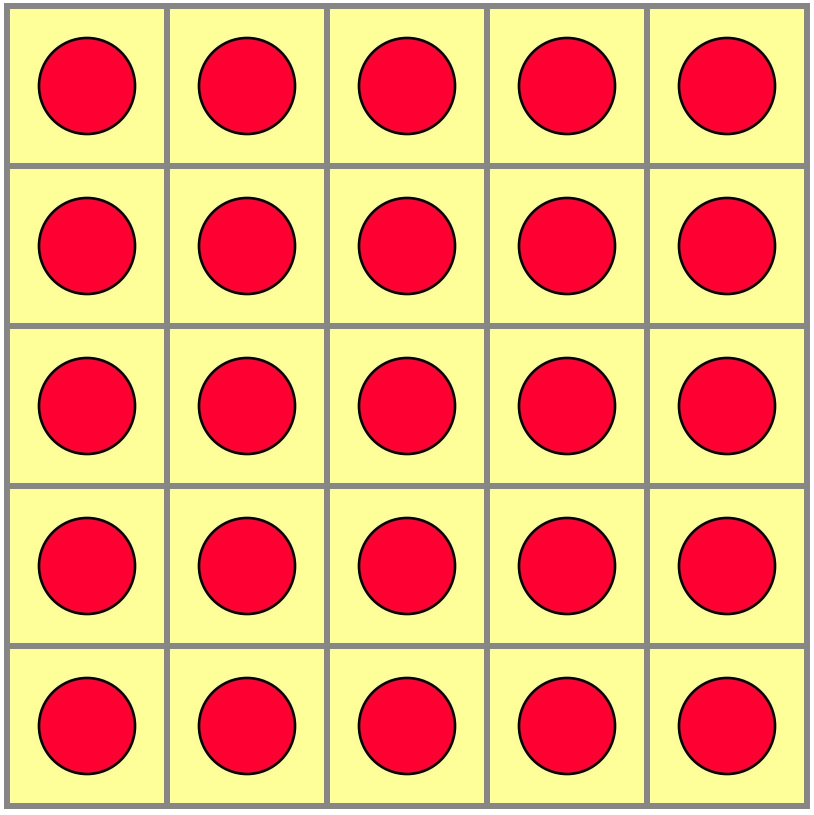
$x(i, j) = 0$ otherwise

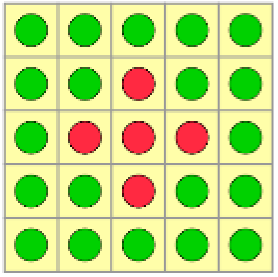
$0 \leq y(i, j) \leq 2$; $y(i, j)$ integer for $i, j = 1$ to 5.

This <u>is</u> an integer program.

x is odd if there is an integer y such that $x - 2y = 1$.

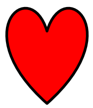
Should I give away the solution?





Trading for Profit

Nooz is a contestant on Trading for Profits. Its main slogan is

"I  Trading for Profit"

Nooz has just won 14 IHTFP points. We now join the quiz show to see what the 14 points are worth.

iPad	Data server	MIT 'Brass Rat' ring
5 points	**7 points**	**4 points**
$500 gift certificate to Au Bon Pain	Tutoring in 6.041 *Probabilistic Systems Analysis*	Dinner with Prof. Orlin and the 15.053 TAs
3 points	**4 points**	**6 points**

iPad	Data server	MIT 'Brass Rat' ring	$500 gift certificate to Au Bon Pain
5 points	**7 points**	**4 points**	**3 points**
16 utils	**22 utils**	**12 utils**	**8 utils**

Tutoring in 6.041 *Probabilistic Systems Analysis*	Dinner with Prof. Orlin and the 15.053 TAs
4 points	**6 points**
11 utils	**19 utils**

Nooz determines what each prize is worth to him. He measure everything in "utils" on a scale from 1 to 25.

Prize	iPad **1**	server **2**	Brass Rat **3**	Au Bon Pain **4**	6.041 tutoring **5**	15.053 dinner **6**
Points	**5**	**7**	**4**	**3**	**4**	**6**
Utility	**16**	**22**	**12**	**8**	**11**	**19**

Budget: 14 IHTFP points.

Write Nooz's problem as an integer program.

$$\textbf{Let } x_i = \begin{cases} 1 & \textbf{if prize i is selected} \\ 0 & \textbf{otherwise} \end{cases}$$

Integer Programming Formulation

Objective and Constraints?

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

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

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

We will solve this problem in two lectures.

Knapsack or Capital Budgeting

- You have n items to choose from to put into your knapsack.
- Item i has weight w_i, and it has value c_i.
- The maximum weight your knapsack (or you) can hold is b.
- Formulate the knapsack problem.

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

Using Excel Solver to Solve Integer Programs

- Add the integrality constraints (or add that a variable is binary)
- Set the Solver Tolerance. (Integer optimality %)
(The tolerance is the percentage deviation from optimality allowed by solver in solving Integer Programs.)
  - The default used to be 5%.
  - A 5% default is way too 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)**

Summary on Integer Programming

- Dramatically improves the modeling capability
  - Economic indivisibilities
  - Logical constraints
  - capital budgeting
  - games
- Not as easy to model
- Not as easy to solve.
- Next lecture: more IP formulations