15.053

Integer programming models

Introduction

1

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

2

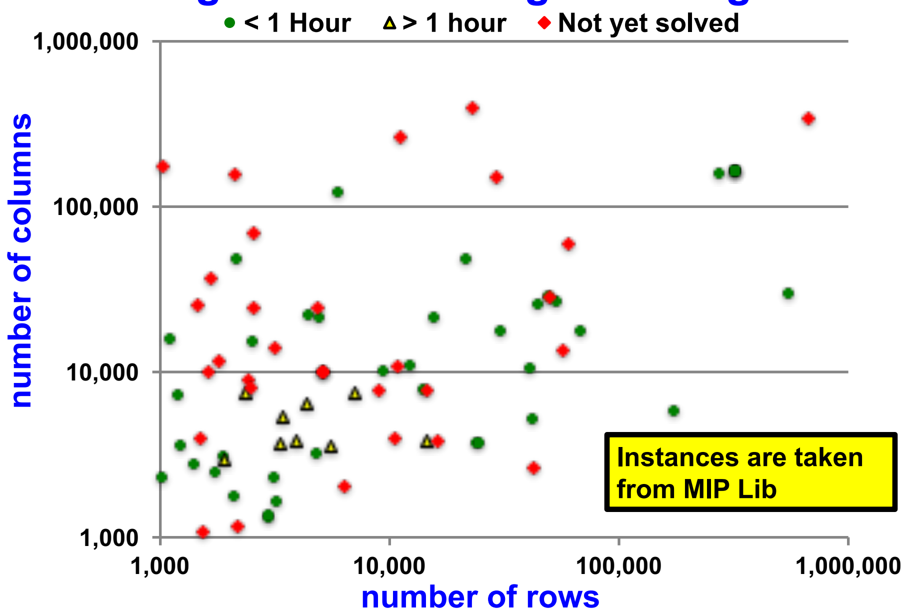
Why integer programs?

- 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.
  - More realistic
  - More flexibility
- Disadvantages
  - More difficult to model
  - Can be much more difficult to solve

On computation for IPs

- Much, much harder than solving LPs
- Some very large IPs can be solved
  - e.g., 50,000 columns 2 million non-zeros
- Very unpredictable!

Running time for solving IPs using CPLEX



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 (binary) Integer Programs — $x_j \in \{0,1\}$ for every j.

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

The Knapsack Problem

a.k.a. The Capital Budgeting Problem

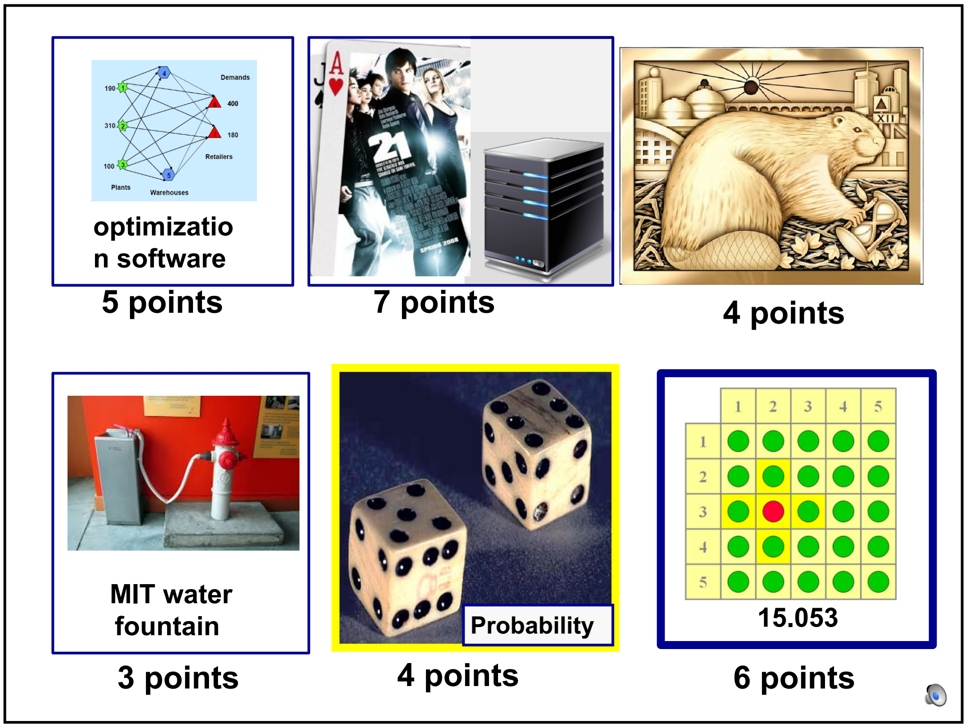
Trading for Profit

Nooz, is a fox. In fact, he is the most trusted name in fox. Nooz is a contestant on Trading for Profits. Its main slogan is

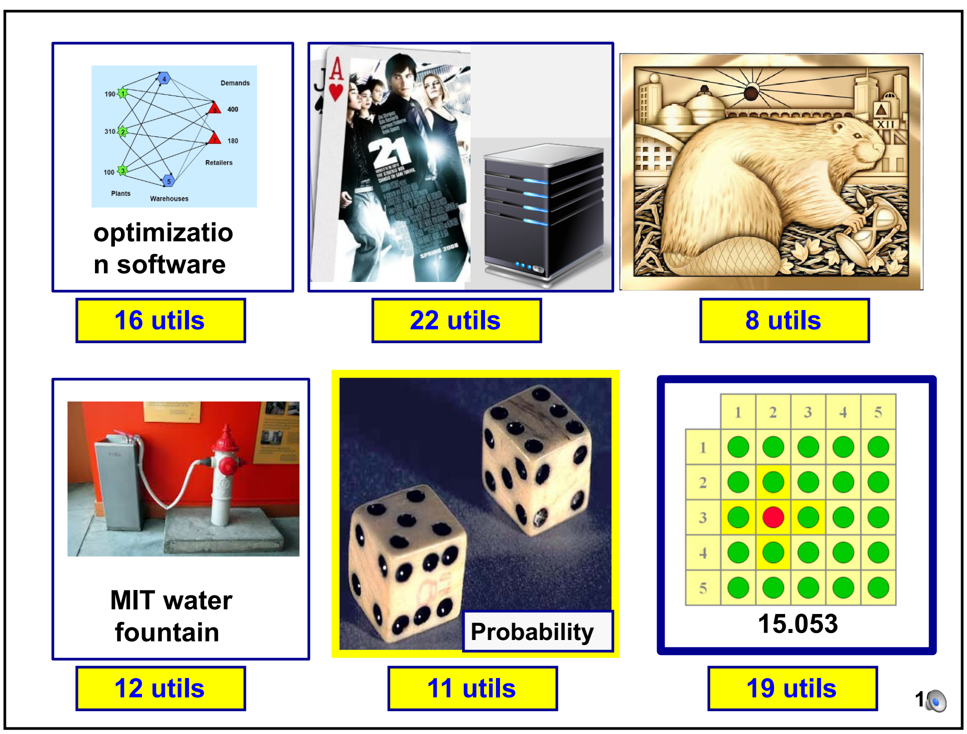
"I ♥ Trading for Profit"

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

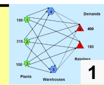
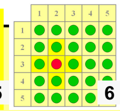
Show a picture of Nooz.



optimization software
5 points
7 points
4 points
MIT water fountain
3 points
Probability
4 points
15.053
6 points



optimization software
16 utils
22 utils
8 utils
MIT water fountain
12 utils
Probability
11 utils
15.053
19 utils
1

Prize						
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.

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

Integer Programming Formulation

Objective and Constraints?

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.

On formulating integer programs

We often have constraints and objectives that are not expressed as linear constraints plus integrality constraints. (That is, it is not currently an IP)

Examples:

- If you select Prize 1, you cannot select Prize 5.
- If Prize 1 is selected then Prize 2 must be selected.
- You must select Prize 1 or Prize 2 or both
- e.g. $y_1 \leq 7$ or $y_2 \leq 9$ or both.
- e.g., $f(y_3) = 0$ if $y_3 = 0$, and it is $10 + 5\, y_3$ if $y_3 > 0$.

On formulating integer programs

We often have constraints and objectives that are not expressed as linear constraints plus integrality constraints. (That is, it is not currently an IP)

Examples:

- If you select Prize 1, you cannot select Prize 5.
- If Prize 1 is selected then Prize 2 must be selected.
- You must select Prize 1 or Prize 2 or both
- e.g. $y_1 \leq 7$ or $y_2 \leq 9$ or both.
- e.g., $f(y_3) = 0$ if $y_3 = 0$, and it is $10 + 5\, y_3$ if $y_3 > 0$.

On formulating integer programs

We can figure out what is feasible and what is not feasible.

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

1. The feasible region for our problem and for the integer program is the "same."
2. Sometimes, we need to create new variables in order to make this happen.

Formulating logical constraints

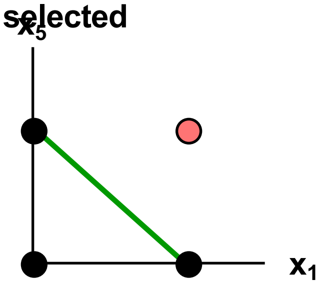
(restricted to binary variables)

If-then constraints

(1) If you select Prize 1, you must select Prize 3.

$$x_i = \begin{cases} 1 & \text{if prize i is selected} \\ 0 & \text{if prize i is not selected} \end{cases}$$

	$x_3 = 1$	$x_3 = 0$
$x_1 = 1$	T	F
$x_1 = 0$	T	T



MIP Constraint:

$x_1 + x_5 \leq 1$

If-then constraints

$$x_i = \begin{cases} 1 & \text{if prize i is selected} \\ 0 & \text{if prize i is not selected} \end{cases}$$

(1) If you select Prize 1, you must select Prize 3.

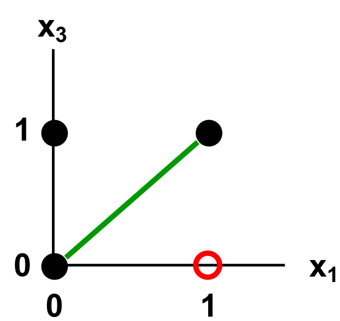
	$x_3 = 1$	$x_3 = 0$
$x_1 = 1$	T	F
$x_1 = 0$	T	T

Setting x_1 to 1 and x_3 to 0 makes this constraint unsatisfied. It is infeasible.

If-then constraints

$$x_i = \begin{cases} 1 & \text{if prize i is selected} \\ 0 & \text{if prize i is not selected} \end{cases}$$

(1) If you select Prize 1, you must select Prize 3.

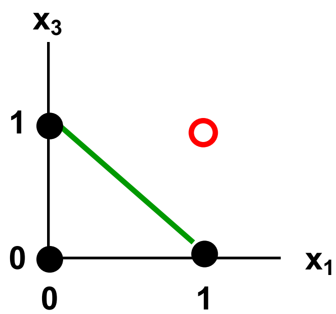


$$x_1 - x_3 \le 0$$

If-then constraints

$$x_i = \begin{cases} 1 & \text{if prize i is selected} \\ 0 & \text{if prize i is not selected} \end{cases}$$

(2) If you select Prize 1, you cannot select Prize 5.

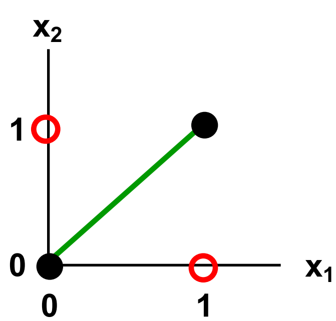


$$x_1 + x_5 \leq 1$$

"If and only if" constraints

$$x_i = \begin{cases} 1 & \text{if prize i is selected} \\ 0 & \text{if prize i is not selected} \end{cases}$$

(3) Prize 1 is selected if and only if Prize 2 is selected.

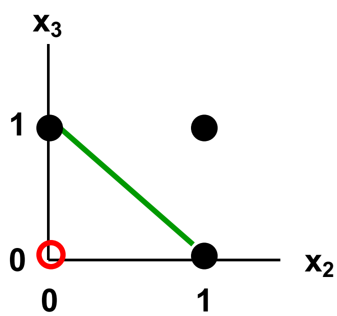


$$x_1 - x_2 = 0$$

“Or” constraints

$$x_i = \begin{cases} 1 & \text{if prize i is selected} \\ 0 & \text{if prize i is not selected} \end{cases}$$

(4) Prize 2 or Prize 3 is selected (or both)

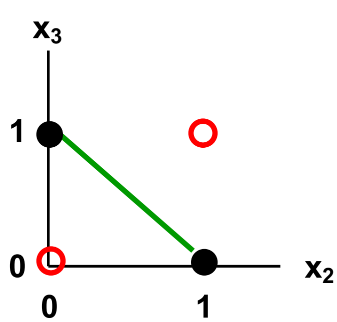


$$x_1 + x_5 \geq 1$$

“Exclusive or” constraints

$$x_i = \begin{cases} 1 & \text{if prize i is selected} \\ 0 & \text{if prize i is not selected} \end{cases}$$

(5) Prize 2 or Prize 4 is selected, but not both



$$x_1 + x_5 = 1$$

How to guarantee a constraint is redundant: intro to big M method

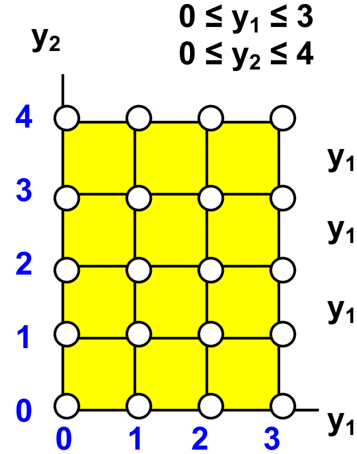
On redundant constraints

A constraint is redundant if deleting the constraint does not change the feasible region.

Suppose that

$0 \le y_1 \le 3$

$0 \le y_2 \le 4$



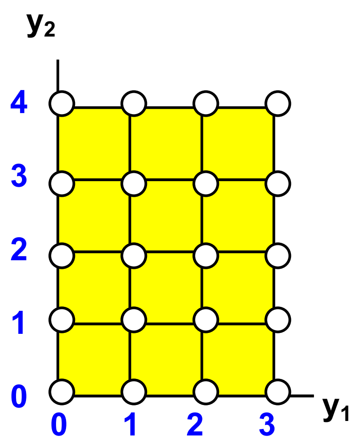
$y_1 + y_2 \le 6$ not redundant

$y_1 + y_2 \le 7$ redundant

$y_1 + y_2 \le M$ redundant for all $M \ge 7$.

On redundant constraints and "big M"

Suppose that
$0 \le y_1 \le 3$
$0 \le y_2 \le 4$



$y_1 - 2y_2 \le 3$

not redundant
($y_1 = 4$, $y_2 = 0$ is infeasible.)

$y_1 - 2y_2 \le 4$

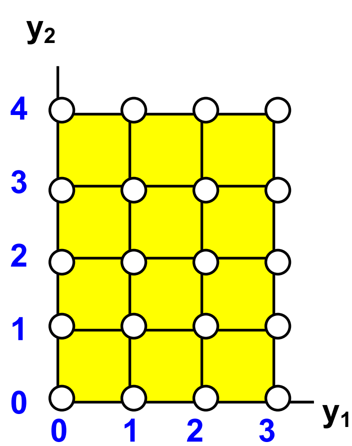
redundant

$y_1 - 2y_2 \le M$

redundant for all $M \ge 4$.

On redundant constraints and "big M"

Suppose that
$0 \le y_1 \le 3$
$0 \le y_2 \le 4$



$10 y_1 - 200 y_2 \ge 0$

not redundant

$10 y_1 - 200 y_2 \ge -799$

not redundant
$y_1 = 0$, $y_2 = 4$ is infeasible.

$10 y_1 - 200 y_2 \ge -800$

redundant

$10 y_1 - 200 y_2 \ge -M$

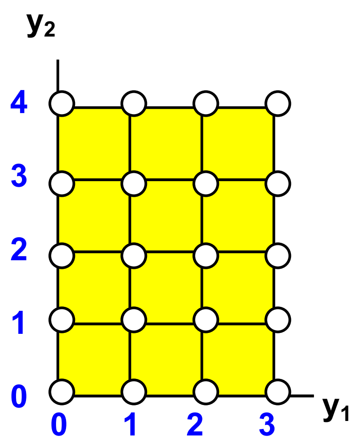
redundant for all $M \ge 800$.

On redundant constraints and "big M"

Suppose that
$0 \le y_1 \le 3$
$0 \le y_2 \le 4$

$a\, y_1 + b\, y_2 \le M$

$a\, y_1 + b\, y_2 \ge -M$



both inequalities are redundant for all sufficiently large M.

(True for all real numbers a and b whenever y_1 and y_2 are both bounded from above and below.)

On choosing the parameter M

Any redundant constraint will be OK.

But, it is often best to choose M *minimal*:

- choose M so that constraint is redundant
- with any lower value of M, it is not redundant.

We will use big M method in modeling "or constraints".

Modeling "or" constraints for mixed integer linear programming.

Logical constraints and the big M method

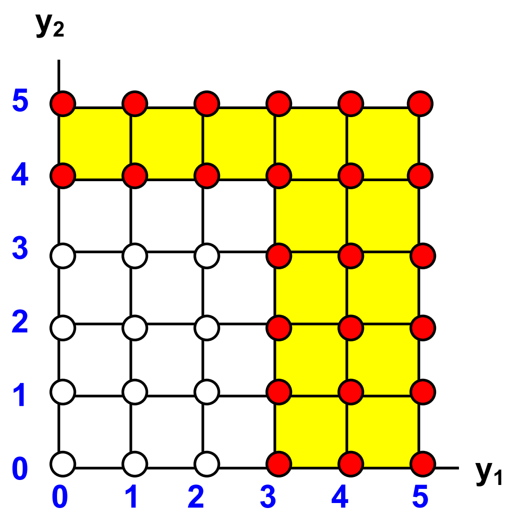
Suppose that y_1 and y_2 are linear variables and

$0 \leq y_1 \leq 5$

$0 \leq y_2 \leq 5$

How can we model:

"$y_1 \geq 3$ or $y_2 \geq 4$ or both"



The feasible region is not convex. It will not be the feasible region of any LP.

Technique: add a binary variable w

if w = 0, then $y_1 \geq 3$

if w = 1, then $y_2 \geq 4$.

An example that doesn't work

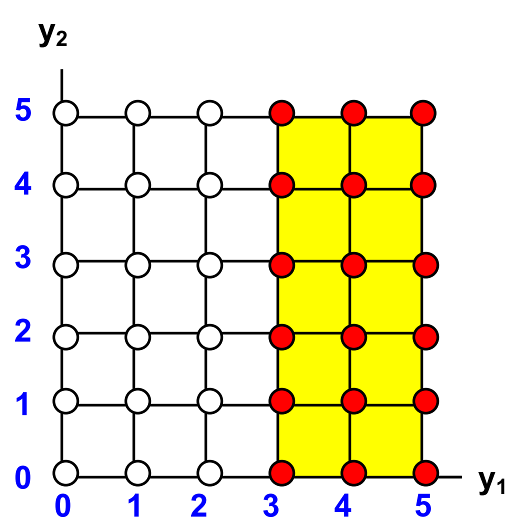
if w = 0, then $y_1 \geq 3$

if w = 1, then $y_2 \geq 4$.

$y_1 \geq 3w$

$y_2 \geq 4w$

With these constraints, y_1 is always ≥ 3.



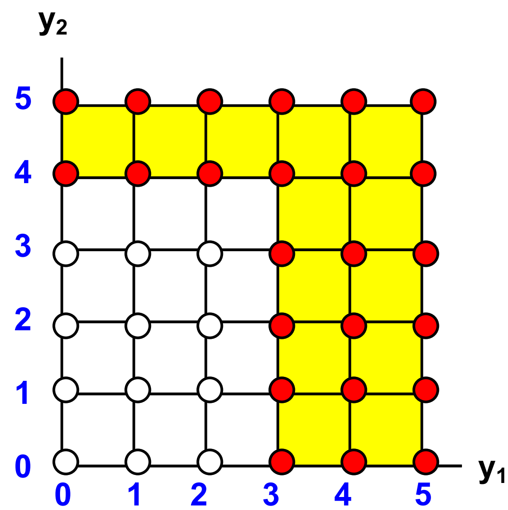
Making it work

if w = 0, then $y_1 \geq 3$

if w = 1, then $y_2 \geq 4$.

$y_1 \geq 3(1-w)$

$y_2 \geq 4w$



35

Case 1: w = 0

if w = 0, then $y_1 \geq 3$

if w = 1, then $y_2 \geq 4$.

$y_1 \geq 3(1-w)$

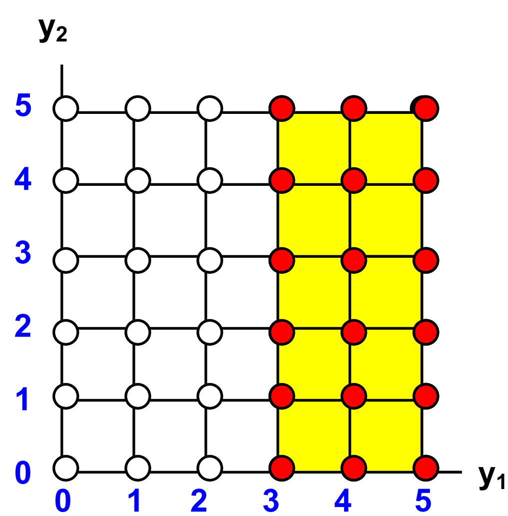
$y_2 \geq 4w$

$w = 0$

$y_1 \geq 3$

$y_2 \geq 0$

$w = 0$



36

Case 2: w = 1

if w = 0, then $y_1 \geq 3$

if w = 1, then $y_2 \geq 4$.

$y_1 \geq 3(1-w)$

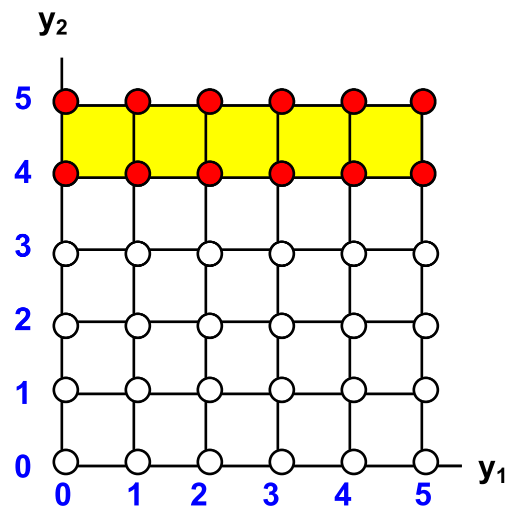
$y_2 \geq 4w$

$w = 1$

$y_1 \geq 0$ redundant

$y_2 \geq 4$

$w = 0$



Combining the two cases

if w = 0, then $y_1 \geq 3$

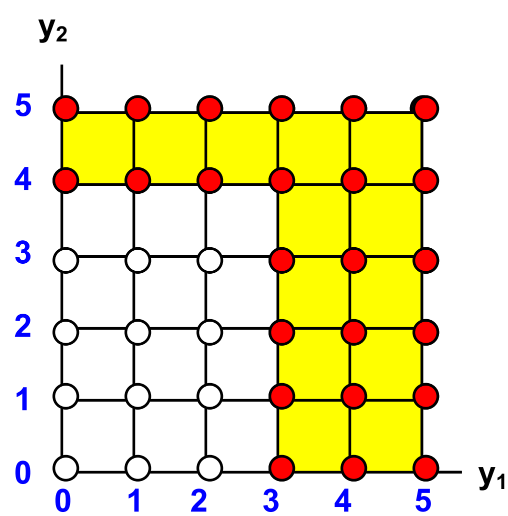
if w = 1, then $y_2 \geq 4$.

$y_1 \geq 3(1-w)$

$y_2 \geq 4w$

Either w = 0

Or w = 1



Modeling “or” constraints using the big M method

“Or constraints” using big M: a key assumption

Assumption. Variables are bounded from above and below. (Bounds may be something that is derived.)

Example 1: The variables are x_1, x_2, x_3, and

$0 \le x_1 \le 100$

$0 \le x_2 \le 125$

$0 \le x_3 \le 73$

Bounded variables

Example 2: The variables are x_1, x_2, x_3, and

$2x_1 + 5x_2 + 10x_3 \leq 20$ (1)

$x_j \geq 0$ for $j = 1, 2, 3$ (2)

(1) and (2) imply that $x_1 \leq 10$, $x_2 \leq 4$, $x_3 \leq 2$.

Example 3: The variables are x_1, x_2, x_3, and

$x_1 + 2x_3 \leq 100$ (3)

$x_2 - x_3 \leq 10$ (4)

$x_j \geq 0$ for $j = 1, 2, 3$ (5)

(3) and (5) imply that $x_1 \leq 100$, $x_3 \leq 50$

$x_3 \leq 50$ and (4) imply that $x_2 \leq 60$

Modeling "or constraints".

Assume that x_1, x_2, and x_3 are all bounded.

$x_1 + 2x_2 \geq 12$ (1)

or

$4x_2 - 10x_3 \leq 1$. (2)

Logical constraints.

Add a variable w. Enforce the following:

If $w = 0$ then $x_1 + 2x_2 \geq 12$ and (2) is redundant.

If $w = 1$, then $4x_2 - 10x_3 \leq 1$ and (1) is redundant.

$x_1 + 2x_2 \geq 12 - M(1-w)$

$4x_2 - 10x_3 \leq 1 + Mw$.

IP constraints.

where M is sufficiently large.

Some Comments on using big M method

- **The technique does require variables to be bounded.**
- **Rather than use a really large value of M, it is better to use minimal values of M.**

Modeling "or" with three choices

Modeling "if-then constraints"

Modeling "or constraints".

Assume that x_1, x_2, and x_3 are all bounded.

$x_1 \geq 12$ (1)

or

$x_2 \leq 8$ (2)

or

$-x_1 + x_3 \geq 4$ (3)

Logical constraints

$x_1 \geq 12 - M(w_1)$ (1')

$x_2 \leq 8 + M(w_2)$ (2')

$-x_1 + x_3 \geq 4 - M(w_3)$ (3')

$w_1 + w_2 + w_3 = 1$

$w_j \in \{0,1)$ for j = 1, 2, 3

Integer program

If $w_1 = 1$, then $x_1 \geq 12$

If $w_2 = 1$, then $x_2 \leq 8$

If $w_3 = 1$, then $-x_1 + x_3 \geq 4$

Modeling "if-then constraints".

Assume that x_1, x_2, and x_3 are all bounded and that x_1 is integer valued.

If $x_1 \leq 12$, then $x_2 + 5x_3 \leq 25$

is logically equivalent to

$x_1 > 12$ or $x_2 + 5x_3 \leq 25$

and also equivalent to

$x_1 \geq 13$ or $x_2 + 5x_3 \leq 25$

because x_1 is integral.

$x_1 \geq 13 - M(1-w_4)$

$x_2 + 5x_3 \leq 25 + Mw_4$

$w_4 \in \{0,1\}$

Integer program

If $w_4 = 0$, then $x_1 \geq 13$

If $w_4 = 1$, then $x_2 + 5x_3 \leq 25$.