Integer Programming

Eg:

$$\begin{aligned} \text{minimize} \quad & -x_1 - x_2 \\ \text{s.t.} \quad & -2x_1 + 4x_2 \le 1 \\ & 9x_1 - 16x_2 \le 4 \qquad \text{(IP)} \\ & x_1, x_2 \text{ integer} \\ & x_1, x_2 \ge 0. \end{aligned}$$

LP-relaxation:

$$\begin{aligned} \min \quad & -x_1 - x_2 \\ \text{st.} \quad & -2x_1 + 4x_2 \le 1 \\ & 9x_1 - 16x_2 \le 4 \\ & x_1, x_2 \ge 0 \end{aligned}$$

Soln: $x_1 = 8$, $x_2 = 4.25$
Value: -12.25.

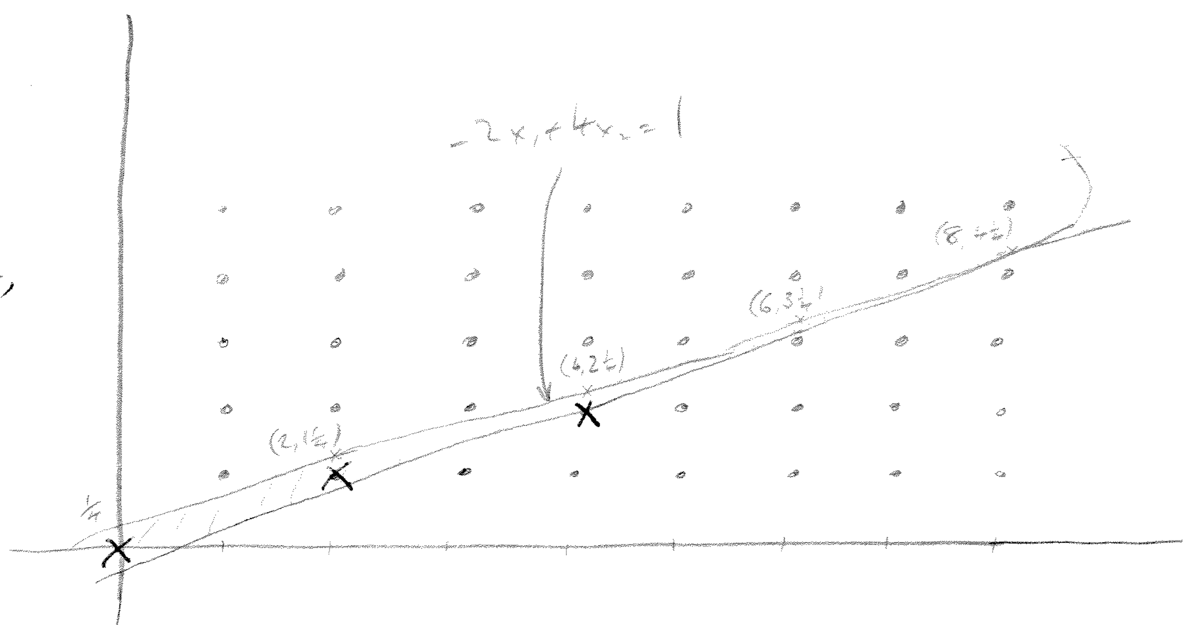
So try to get soln to (IP) by rounding this solution:

Try $x_1 = 8$, $x_2 = 4$: Infeasible in IP: $9 \times 8 - 16 \times 4 = 72 - 64 = 8 >$

Try $x_1 = 8$, $x_2 = 5$: Infeasible in IP: $-2 \times 8 + 4 \times 5 = -16 + 20 = 4 >$

In fact:

Soln to (IP) is
$x_1 = 4$, $x_2 = 2$,
value -6.



So, in general, can not get soln to IP by rounding soln to ~~LP~~ LP relaxation.

Note that we can say:

If z^*_{IP} is the minimum value for an integer program,
and z^*_{LP} is the minimum value for its LP-relaxation,
then $z^*_{IP} \geq z^*_{LP}$.

So LP relaxation provides a lower bound on the optimal objective value for the integer program.

Another solution method:
exhaustive enumeration

From ~~exam sol~~ the LP relaxation, can conclude that any feasible integer soln has $x_1 \leq 8$, $x_2 \leq 5$.

So could examine every possible combination

$x_1 = 0, 1, 2, 3, 4, 5, 6, 7, 8$

$x_2 = 0, 1, 2, 3, 4, 5$.

Impractical for even reasonable sized problems:
10 variables each with up to 6 different values gives
$6^{10} = 60,466,176$ possible combinations.

Implicit Enumeration

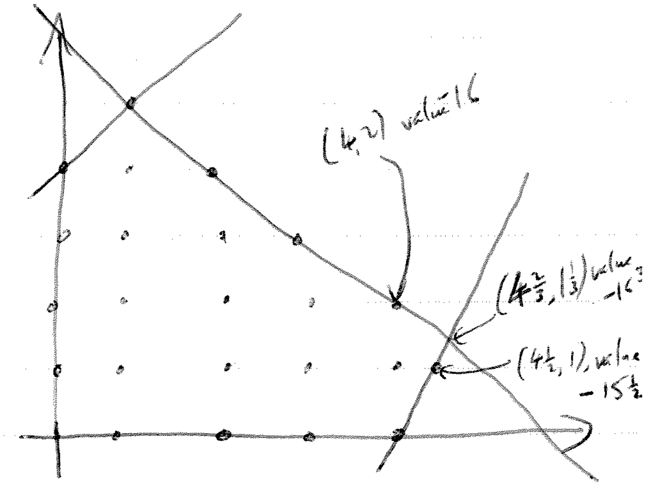
Eg: min $-3x_1 - 2x_2$

s.t. $x_1 + x_2 \leq 6$

$-x_1 + x_2 \leq 4$ (IP)

$2x_1 - x_2 \leq 8$

$x_1, x_2 \geq 0$, x_1, x_2 integer



Let $F = \{x : x_1 + x_2 \leq 6, -x_1 + x_2 \leq 4, 2x_1 - x_2 \leq 8, x_1, x_2 \geq 0\}$

Solution to LP-relaxation is $x_1 = 4\frac{2}{3}$, $x_2 = 1\frac{1}{3}$, value $-16\frac{2}{3}$.

Solution to (IP) must have either $x_2 \geq 2$, x in F

or $x_2 \leq 1$, x in F

So solve two problems:

min $-3x_1 - 2x_2$	min $-3x_1 - 2x_2$
s.t. $x_2 \geq 2$	s.t. $x_2 \leq 1$
x in F	x in F
Soln: $x_1 = 4, x_2 = 2$, value -16	Soln: $x_1 = 4\frac{1}{2}$, $x_2 = 1$, value $-15\frac{1}{2}$.

Subproblems.

Constructing subproblems is called BRANCHING

Say we BRANCHED on x_2.

Do not need to branch further because:

(i) Solution to left hand problem is integer, so ~~no subprob~~ this point solves the problem $\min\{-3x_1 - 2x_2 : x \text{ in } F, x_2 \geq 2, x \text{ integer}\}$.

(ii) Solution to right hand problem has value > value of known integer solution, so any integer point in F with $x_2 \leq 1$ must be worse than $(4,2)$.

Could have branched initially on x_1:

$\min -3x_1 - 2x_2$

s.t. $x_1 \le 4$

x in F

Soln: $x_1 = 4$, $x_2 = 2$

Value -16

No point in branching further.

$\min -3x_1 - 2x_2$

s.t. $x_1 \ge 5$

x in F

Infeasible

No point in branching further

Process is called BRANCH AND BOUND

- branch, then use reasoning to eliminate problems by using bounds.

When eliminate a subproblem from further consideration, called FATHOMING. Can happen in three ways:

(i) Soln to problem is integer

(ii) Soln to subproblem has value worse than a known integer solution

(iii) Subproblem is infeasible.

Solution by Branch and Bound

$$\begin{aligned} \min \quad & z(x) = c^T x \\ \text{s.t.} \quad & x \text{ in } F = \left\{x : Ax \left\{\begin{matrix}\geq \\ \leq \\ =\end{matrix}\right\} b,\ x \geq 0\right\}. \qquad (IP) \\ & x \text{ integer} \end{aligned}$$

Algorithm:

0. Initialize

Start by solving LP relaxation.
Find an upper bound z_u on value of (IP)
(take $z_u = +\infty$ if necessary)

1. Branch

Divide an unfathomed subproblem ~~with a fractional solution~~ into two problems, and solve the problems

2. Bound

For each new subproblem, let z be the value of the subproblem.

3. Fathom

Fathom if any of the following hold:

(i) $z \geq z_u$ (bounds)
(ii) Subproblem infeasible (infeasibility)
(iii) Subproblem has integral optimal solution (integrality)

4. If no unfathomed nodes remain, STOP.
Else, return to 1.

A Branch and Bound Example in Detail.

$$\min \quad z(x) = 5x_1 + 5x_2 - 13x_3$$

$$\text{s.t.} \quad \begin{aligned} x_1 + x_2 + x_3 &\le 6 \\ 10x_1 - 8x_2 &\le 15 \\ -6x_1 - x_2 + 9x_3 &\le 9 \\ x_i \ge 0, \quad x_i &\text{ integer} \end{aligned} \qquad (IP)$$

Let $F = \{x \geq 0 : x_1 + x_2 + x_3 \le 6, 10x_1 - 8x_2 \le 15, -6x_1 - x_2 + 9x_3 \le 9\}$

Let z_u = value of best known feasible integer solution.

Note that $x = (0, 0, 0)$ is feasible in (IP), so we can initialize $z_u = 0$.

We will branch on the variable with fractional part closest to $\frac{1}{2}$.

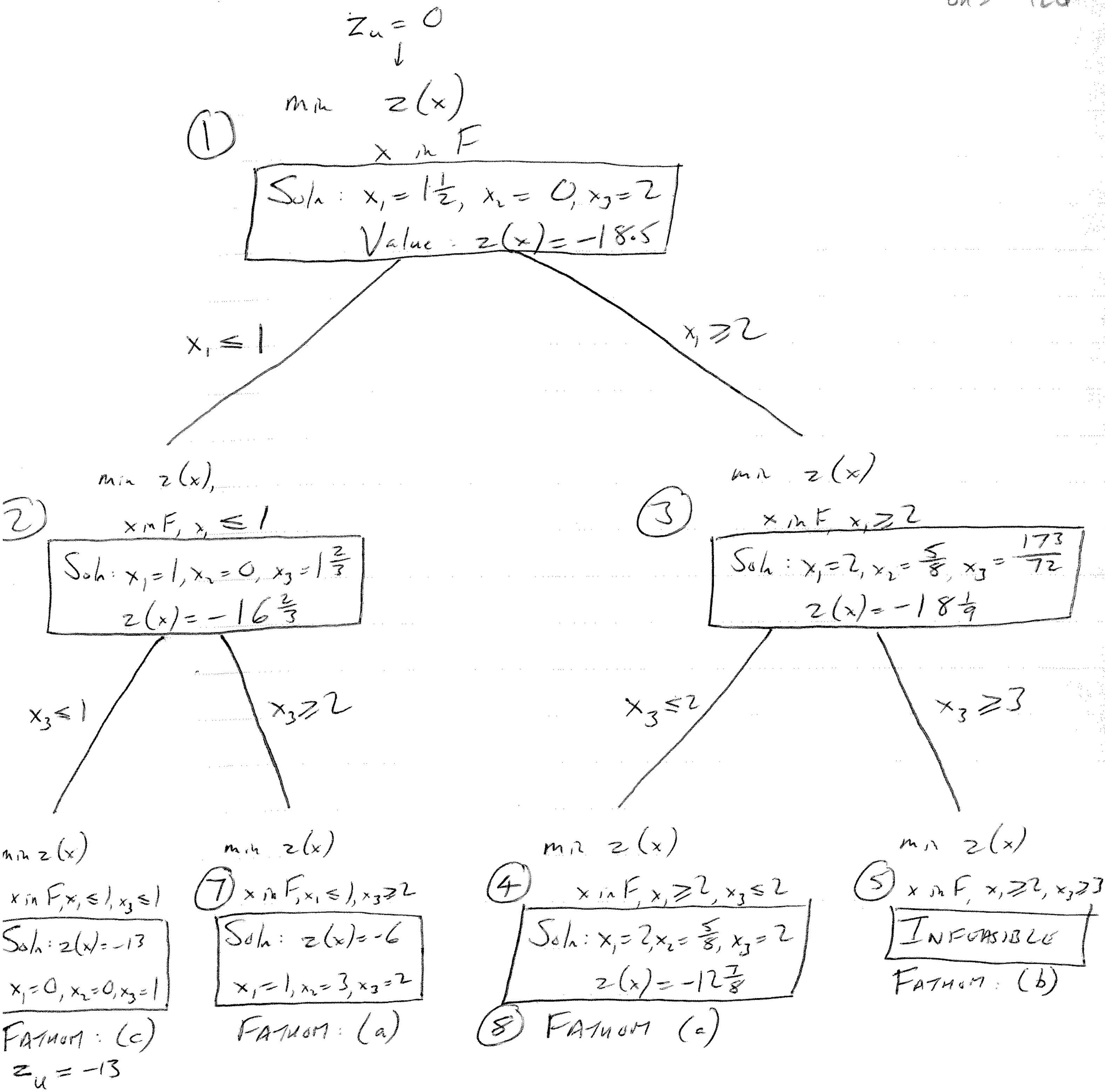
The node we branch on is the one with smallest value.

We can fathom by any of:

(a) Bounds: $z \geq z_u$.

(b) Infeasibility: Subproblem is infeasible

(c) Integrality: Subproblem has integer optimal solution with value $< z_u$.



Optimal solution: $x_1 = 0$, $x_2 = 0$, $x_3 = 1$, $z(x) = -13$.

INTEGER PROGRAMMING FORMULATION EXAMPLES

CAPITAL BUDGETING.

(Eg: building roads
- need Road A to be able to use Road B
- Roads C, D, E share same endpoints so only one of them.)

m resources available (eg money, workforce, civil engineers)
n projects being considered

r_j = revenue if project j undertaken
b_i = amount of resource i available
a_{ij} = amount of resource i required for project j.

What projects should be undertaken to maximize total revenue returned?

$$x_j = \begin{cases} 1 & \text{if project } j \text{ undertaken} \\ 0 & \text{otherwise} \end{cases}$$

$$\max \sum_{j=1}^{n} r_j x_j$$

$$\text{s.t.} \sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad i = 1, \ldots, m$$

$$x_j = 0 \text{ or } 1.$$

Additional constraints:

Eg: Project 3 can only be undertaken if Project 2 is also undertaken.

Model as $x_3 \leq x_2$, or $-x_2 + x_3 \leq 0$.

Eg: At most one of projects 1, 4, 8 can be undertaken.

Model as $x_1 + x_4 + x_8 \leq 1$.

IP 2

FACILITY LOCATION (Eg: locations for Warehouses, then ship to stores)

Facilities for distributing a product to n customers can be placed at m possible locations.

If location i is used, fixed cost F_i of setting up the location, and c_{ij} is cost of having location i send one unit to customer j.

Customer j demands d_j units of the product.

Where should facilities be located, and how should shipping be scheduled

Let $y_i = \begin{cases} 1 & \text{if a facility is placed at location } i \\ 0 & \text{otherwise.} \end{cases}$

x_{ij} = amount shipped from ~~facility~~ location i to ~~location~~ customer j.

If no facility is built at location i, need $x_{ij} = 0 \ \forall j$.

If a facility is built at location i, ~~need~~ must have $x_{ij} \leq d_j \ \forall j$.

Model as: $x_{ij} \leq y_i d_j \quad i=1,\dots,m, \ j=1,\dots,n.$

So: if $y_i = 0$ $\underline{\text{must have}}$ $x_{ij} = 0$

$\quad\ \ y_i = 1$ $\quad\quad\quad\quad\quad\ \ x_{ij} \leq d_j$ ✓.

So: $\min \underbrace{\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}}_{\text{cost of shipments}} + \underbrace{\sum_{i=1}^{m} F_i y_i}_{\text{cost of facilities}}$

$\sum_{i=1}^{m} x_{ij} = d_j \quad j=1,\dots,n$ meet customer demand

$x_{ij} \leq y_i d_j \quad i=1,\dots,m, \ j=1,\dots,n$

$x_{ij} \geq 0 \quad i=1,\dots,m, \ j=1,\dots,n$

This is a <u>mixed</u> integer program: ie, it contains both continuous and ~~int~~ integer variables.

Can aggregate the $x_{ij} \le y_i d_j$ constraints:

$$\sum_{j=1}^{n} x_{ij} \le y_i \sum_{j=1}^{n} d_j$$

If $y_i = 1$: no restriction on x_{ij}

$y_i = 0$: must have $x_{ij} = 0$ for all i.

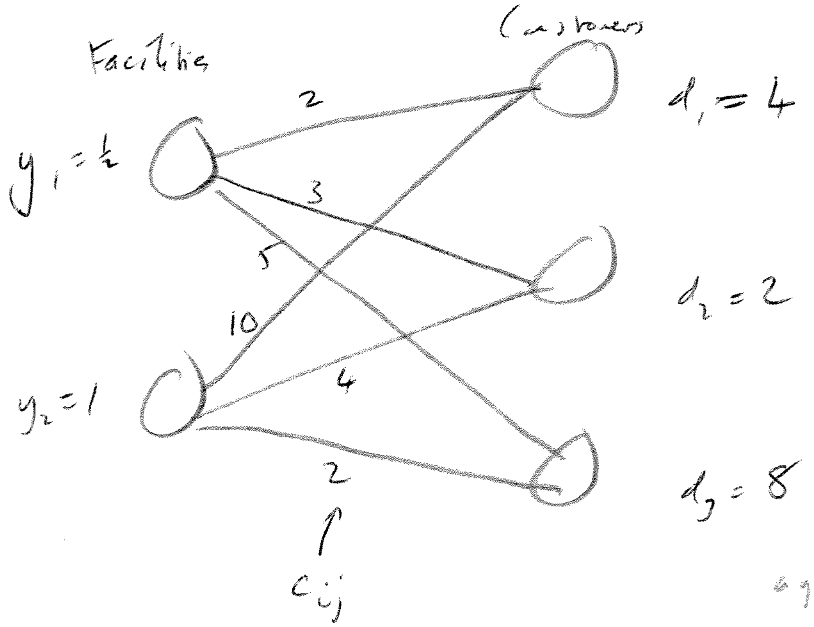
So: replace n constraints with 1 constraint.

Disadvantage: LP relaxation is weaker:

If we have $y_i = \frac{1}{2}$ then ~~in the first~~

– in the first formulation, must have each $x_{ij} \le \frac{1}{2} d_j$

– in the second formulation, could have some $x_{ij} = 0$, which allows other x_{ij} to be $> \frac{1}{2} d_j$.



Take $x_{11} = 4$, $x_{12} = 2$

$x_{23} = 8$

Satisfies the aggregated constraint, but not the disaggregated constraint.

aggregated constraint: $x_{11} + x_{12} + x_{13} \le y_1 (2+4+8) = 14 y_1$

TRAVELING SALESMAN PROBLEM (Eg: VLSI chip design, X-ray chromatography)

Traveling salesman must visit each of n cities exactly once and return to the original city.

If the cost of travel from city i to city j is c_{ij}, in what order should the cities be visited?

Looking for a tour made up of links.

Let $x_{ij} = \begin{cases} 1 & \text{if tour uses a link from city } i \text{ to city } j \\ 0 & \text{otherwise.} \end{cases}$

To visit each city once:

Must arrive at each city once, must leave each ~~other~~ city once.

So: $\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1, \ldots, n$ arrive at city j exactly once

$\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, \ldots, n$ ~~arrive at~~ leave city i exactly once.

Total cost: $\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$

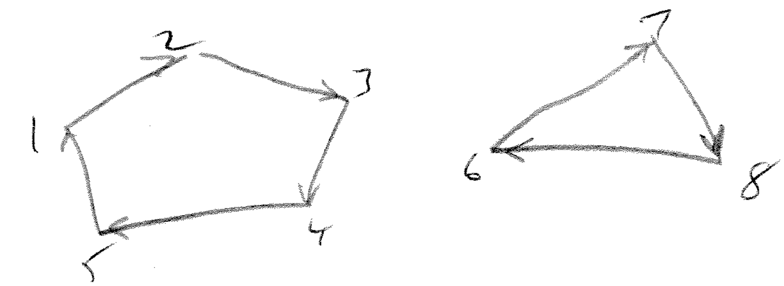
So:
$$\begin{aligned} \min \quad & \sum_i \sum_j c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_i x_{ij} = 1 \quad \forall j \\ & \sum_j x_{ij} = 1 \quad \forall i \\ & x_{ij} = 0 \text{ or } 1 \end{aligned}$$

Leave	Enter
O	O
O	O
O	O
O	O

This is an assignment problem

But: does not ensure the solution is connected.

Eg:

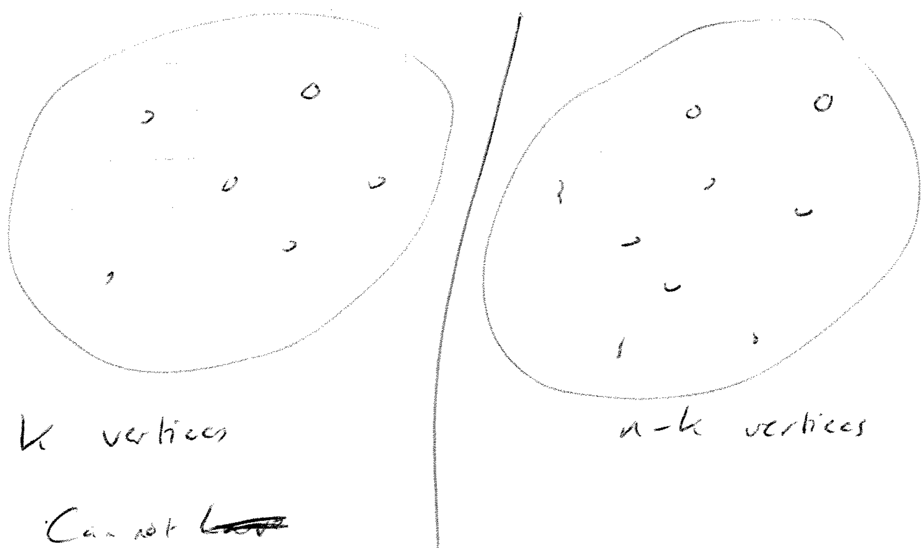


These are <u>subtours</u>.

Can use <u>subtour elimination constraints</u>:

$$x_{67} + x_{78} + x_{86} \, (+ x_{76} + x_{87} + x_{68}) \leq 2$$

Can only use two of the three edges in this triangle



Can not ~~ltoo~~ use more than k-1 edges in here

Need at least two edges going across here.

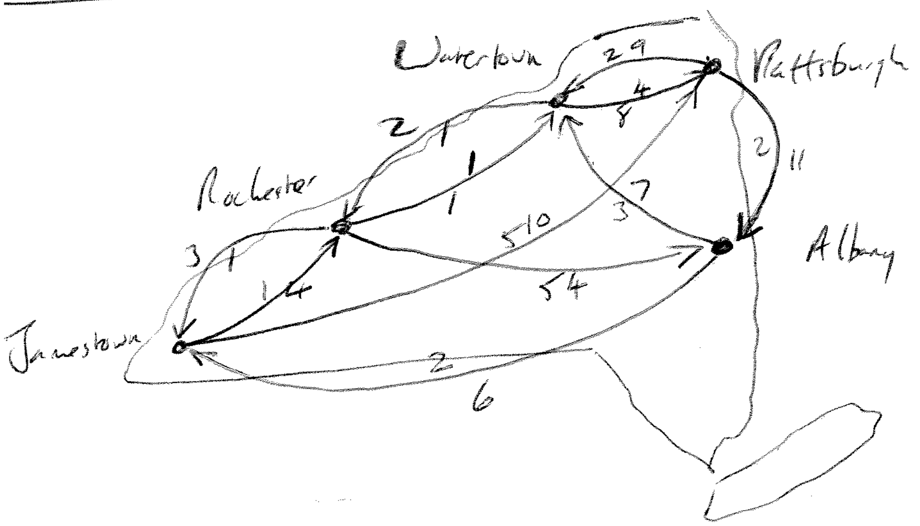
Need one such constraint for each subset of the vertices.

$\binom{n}{3} + \binom{n}{4} + \cdots + \binom{n}{n-3}$ of these constraints.

Generate them as needed as extra constraints.

Can solve problems with over ~~2000~~ 15000 cities ($\approx$ ~~4×10^6~~ edges).

Crew Scheduling



Blue: flight numbers.

Red: flight times.

Each flight must be flown at least once each day.

Four pilots:

Able, Baker live in Plattsburgh
Smith, Jones live in Jamestown.

Requirements:

- Each day, each pilot flies a sequence of flights starting and ending at his or her home base.
- No pilot is permitted to fly more than 10 hours per day
- The sequence of flights flown by a pilot must not include any flight more than once. So not, eg, 4-1-2-5-~~7~~-2-3 (although this also takes 12 hours)

Costs:

\$10 per hour flying time, \$10 per takeoff.

Problem:

Assign pilots to flights at minimum cost.

Want to assign flights to pilots.

Need to map out routes for pilots.

So: Consider all the possible routes the pilots could fly.

Eg: 9-2-3-4-1-8 can be flown by A or B at a cost of $10 \times (\underbrace{2+1+1+1+1+4}_{\text{flight hours}}) + 10 \times \underbrace{6}_{\text{number of takeoffs}} = \$160$

Also, 4-1-8-9-2-3 can be flown by S or J, also at a cost of \$160.

Start city is not important, so company does not have a preference between these.

Thus: look for circuits of length ≤ 10 hours which include Plattsburgh and/or Jamestown.

Have 12 possible routes: (2-5-7 impossible, eg, $\underbrace{11-7-2-1-8}_{\text{too long}}$ impossible)
neither Plattsburgh or Jamestown

Route number	Feasible for pilots	Flights in the route	Cost c_j
1	A, B	9, 8	80
2	A, B	11, 7, 8	120
3	A, B	9, 2, 1, 8	120
4	A, B, S, J	~~11, 6,~~ 9, 2, 3, 4, 1, 8	160
5	A, B, S, J	11, 6, 4, 1, 8	150
6	A, B, S, J	11, 6, 10	120
7	A, B, S, J	9, 2, 3, 10	130
8	S, J	4, 3	40
9	S, J	4, 5, 6	100
10	S, J	4, 1, 2, 3	80
11	S, J	4, 1, 2, 5, 6	140
12	S, J	4, 5, 7, 2, 3	150

Decision variables:

$$x_j = \begin{cases} 1 & \text{if route } j \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

Objective function:

$$\min \sum_{j=1}^{12} c_j x_j .$$

Constraints:

- Need to cover each leg.

Eg, to cover leg 6, need to fly one of routes 5, 6, 9, 11.

So need $x_5 + x_6 + x_9 + x_{11} \geq 1$.

Summarise in matrix form:

$Ax \geq e$ where $a_{ij} = \begin{cases} 1 & \text{if flight } i \text{ in route } j \\ 0 & \text{otherwise} \end{cases}$

and e = vector of all one's.

- Only have a certain number of pilots:

Can only fly two of routes 1, 2, 3. So get constraint:

$$x_1 + x_2 + x_3 \leq 2.$$

Similarly,

$x_8 + x_9 + x_{10} + x_{11} + x_{12} \leq 2$ pilots in Jamestown

Also, $\sum_{1}^{12} x_i \leq 4$ all pilots.

Real world instances have potentially millions of columns.
So use column generation – only add them in as needed.
Update ~~sim~~ solution as for "new product" sensitivity analysis.

This is an example of a covering problem.

Sabre
~~AADT~~, US Air, Delta, United ~~[illegible]~~.
... Sabre

Enforcing Logical Conditions.

eg: 1. $\begin{cases} y \text{ is } 0\text{-}1 \text{ variable.} \\ x_1, x_2 \geq 0 \text{ real} \\ x_1 \leq 3, \quad x_2 \leq 5 \\ f(x) := 2x_1 + x_2 \leq 7 \quad \text{only if } y = 0 \end{cases}$

(if $y=1$, ignore the constraint)

How do we model this?

Now, $2x_1 + x_2 \leq 6 + 5 = 11 = 7 + 4$ for all feasible x.

So: could impose the constraint

$$2x_1 + x_2 \leq 7 + 4y$$

If $y = 0$ this becomes $2x_1 + x_2 \leq 7$ - good

$y = 1$ this becomes $2x_1 + x_2 \leq 11$ - ~~becomes~~ redundant.

So constraint $2x_1 + x_2 \leq 7$ is only "turned on" when $y = 0$.

2. $\left.\begin{array}{l} 2x_1 + x_2 \leq 7 \\ x_1 + 3x_2 \leq 10 \end{array}\right\}$ at least one must hold.

Together with $0 \leq x_1 \leq 3$, $0 \leq x_2 \leq 5$.

Now, $x_1 + 3x_2 \leq 18 = 10 + 8$ for any feasible x.

So: $\left.\begin{array}{l} 2x_1 + x_2 \leq 7 + 4y_1 \\ x_1 + 3x_2 \leq 10 + 8y_2 \end{array}\right\}$ if $y_i = 0$ then constraint i is turned on

and $y_1 + y_2 \leq 1$ - at least one of $y_i = 0$.

$y_1, y_2 = 0$ or 1.

3. x must take one of the values $1.2, 2.6, 3.1, 4.5$:

Replace x by $x = 1.2y_1 + 2.6y_2 + 3.1y_3 + 4.5y_4$

with the restriction that $y_1 + y_2 + y_3 + y_4 = 1$,

$y_i = 0$ or 1.

Something to <u>not</u> do:

$y = 0$ or 1; $0 \le x_1 \le 3$, $0 \le x_2 \le 5$, $2x_1 + x_2 \le 7$ if $y = 0$.

Could model as:

$$(1-y)(2x_1 + x_2 - 7) \le 0$$

If $y = 0$ this forces $2x_1 + x_2 - 7 \le 0$ ie $2x_1 + x_2 \le 7$

$y = 1$, constraint is vacuous $(0 \le 0)$.

Problem: introduces a <u>nonlinear</u> constraint.

Integer programs with nonlinear constraints are far harder to solve than integer programs with only linear constraints.

---

4. Piecewise linear function.

SOS constraint.

IPF 11

Multicommodity Network Flow

Two formulations:

1. Edge-based variables
2. Path-based variables.

Network Design

1. Connectivity requirements.
2. Ring-based formulation.

...er Ordering

...stering

Gomory Cutting Plane

Eg: Get optimal tableau to LP relaxation:

	x_1	x_2	x_3	x_4
-6	0	0	1	5
$4\frac{1}{2}$	1	0	$1/4$	$-1/3$
$5\frac{1}{2}$	0	1	$5/4$	$13/3$

First constraint is

$$x_1 + \frac{1}{4}x_3 - \frac{1}{3}x_4 = 4\frac{1}{2}$$

Gomory cut: $\frac{1}{4}x_3 + \frac{2}{3}x_4 \geq \frac{1}{2}$ (1)

Introduce slack and add to tableau:

	x_1	x_2	x_3	x_4	x_5
-6	0	0	1	5	0
$4\frac{1}{2}$	1	0	$\frac{1}{4}$	$-\frac{1}{3}$	0
$5\frac{1}{2}$	0	1	$5/4$	$13/3$	0
$-\frac{1}{2}$	0	0	$-1/4$	$-2/3$	1
			4	$7\frac{1}{2}$	

Dual simplex pivot:

-8	0	0	0	$7/3$	4
4	1	0	0	-1	1
3	0	1	0	1	5
2	0	0	1	$8/3$	-4

Get integral solution, so also optimal to the integer program.

If $B^{-1} = \begin{bmatrix} \frac{1}{4} & -\frac{1}{3} \\ 5/4 & 13/3 \end{bmatrix}$ then $B = \frac{2}{3}\begin{bmatrix} 13/3 & \frac{1}{3} \\ -5/4 & \frac{1}{4} \end{bmatrix}$

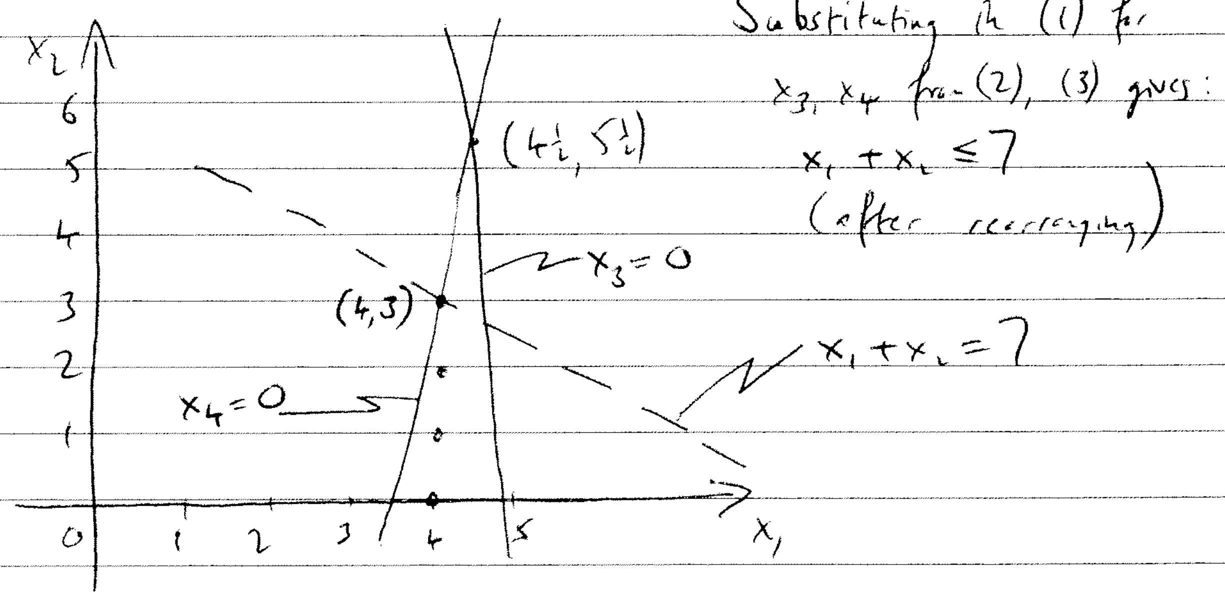
If $B^{-1}b = \begin{bmatrix} 4\frac{1}{2} \\ 5\frac{1}{2} \end{bmatrix}$ then $b = \frac{1}{3}\begin{bmatrix} 13/3 & \frac{1}{3} \\ -5/4 & \frac{1}{4} \end{bmatrix}\begin{bmatrix} 9 \\ 11 \end{bmatrix} = \begin{bmatrix} \frac{128}{9} \\ -\frac{34}{12} \end{bmatrix}$

So constraints can be written

$$\frac{26}{9}x_1 + \frac{2}{9}x_2 \le \frac{128}{9}$$
$$-\frac{5}{6}x_1 + \frac{1}{6}x_2 \le -\frac{17}{6}$$

or
$$13x_1 + x_2 \le 64$$
$$-5x_1 + x_2 \le -17$$

or
$$13x_1 + x_2 + \frac{9}{2}x_3 = 64 \qquad (2)$$
$$-5x_1 + x_2 + 6x_4 = -17 \qquad (3)$$



Substituting in (1) for x_3, x_4 from (2), (3) gives:
$x_1 + x_2 \le 7$
(after rearranging)

Can extend to mixed integer LPs.

Can use more sophisticated version of argument to get stronger cuts.

Can show that get convergence to optimal solution to IP (provided follow certain rules in adding the cuts).

Gomory cut for knapsack example

$$x_3 + 0.2 s_4 - 0.2 s_5 - 0.6 s_6 = 0.6$$

Or $(x_3 - s_5 - s_6) + \underbrace{(0.2 s_4 + 0.8 s_5 + 0.4 s_6)}_{\text{So this is } \geq 0.6} = 0.6$

Gomory cut

$$0.2 s_4 + 0.8 s_5 + 0.4 s_6 \geq 0.6$$

Equivalently

$$2 s_4 + 8 s_5 + 4 s_6 \geq 6$$

or $\quad s_4 + 4 s_5 + 2 s_6 \geq 3$

or $\quad (7 - x_1 - 3x_2 - 5x_3) + 4(1 - x_1) + 2(1 - x_2) \geq 3$

or $\quad 13 - 5x_1 - 5x_2 - 5x_3 \geq 3$

or $\quad x_1 + x_2 + x_3 \leq 2.$