CS 524: Introduction to Optimization
Lecture 18 : Set Covering

Michael Ferris

Computer Sciences Department
University of Wisconsin-Madison

October 16, 2023

Set Covering `simpsetcov.gms`

- Suppose we are given a set of objects $S = \{a, b, \ldots f\}$.
- We are also given a set $\mathcal{C}$ of subsets of S.
- $S = \{a, b, c, d, e, f\}$
  - $\mathcal{C} = \{\{a, b\}, \{a, c, e\}, \{a, b, d, e\}, \{d, e\}, \{c, f\}\}$
- The problem is to choose a set of subsets (from $\mathcal{C}$) so that all of the members of S are "covered".
- The set $C_1 = \{\{a, b\}, \{a, b, d, e\}, \{c, f\}\}$ is a cover
- We may be interested in finding a cover of minimum cost
- Let $x_j = 1$ if the jth subset of $\mathcal{C}$ is used in the cover, so C_1 corresponds to $x = (1, 0, 1, 0, 1)$

Formulation of example

minimize

$$x_1 + x_2 + x_3 + x_4 + x_5$$

subject to

$$\begin{array}{ccccccccccccc}
x_1 & + & x_2 & & x_3 & & & & & \geq & 1 & (a) \\
x_1 & & & + & x_3 & & & & & \geq & 1 & (b) \\
 & & x_2 & & & & & x_5 & & \geq & 1 & (c) \\
 & & & & x_3 & + & x_4 & & & \geq & 1 & (d) \\
 & & x_2 & + & x_3 & & x_4 & & & \geq & 1 & (e) \\
 & & & & & & & x_5 & & \geq & 1 & (f) \\
 & & & & & & & x_j & \in \{0,1\} & \forall j & &
\end{array}$$

More Set Covering: Kilroy County `kilroy.gms`

- There are 6 cities in Kilroy County.
- The county must determine where to build fire stations to serve these cities. They want to build the stations in some of the cities, and to build the minimum number of stations needed to ensure that at least one station is within 15 minutes driving time of each city.
- Can we formulate an integer program whose solution gives the minimum number of fire stations and their locations.

Kilroy county model

$$S = \{\{1,2\},\{1,2,6\},\{3,4\},\{3,4,5\},\{4,5,6\},\{2,5,6\}\}$$

$$\begin{aligned}
&\underset{x}{\text{minimize}} \quad x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\
&\text{subject to:} \quad \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\
&\qquad\qquad x_i \in \{0,1\} \quad \forall i
\end{aligned}$$

- Optimal solution: $x_2 = x_4 = 1$ (two fire stations)

Driving Distances

	1	2	3	4	5	6
1	0	10	20	30	30	20
2	10	0	25	35	20	10
3	20	25	0	15	30	20
4	30	35	15	0	15	25
5	30	20	30	15	0	14
6	20	10	20	25	14	0

Variables

- $x_j = 1$ if build in city j

Vehicle Routing `vr.gms`

- You have a fleet of k trucks, each with a capacity Q
- You have a set of customers $N = \{0, 1, 2, \ldots n\}$
- 0 is a special "depot" node
- Each customer has a demand b_i, $i \in N \setminus \{0\}$
- How do route trucks to meet customer demand at minimum cost?
  - A truck must visit every customer
  - The sum of the demands on the route visiting the customer must be $\leq Q$

Can you do it?

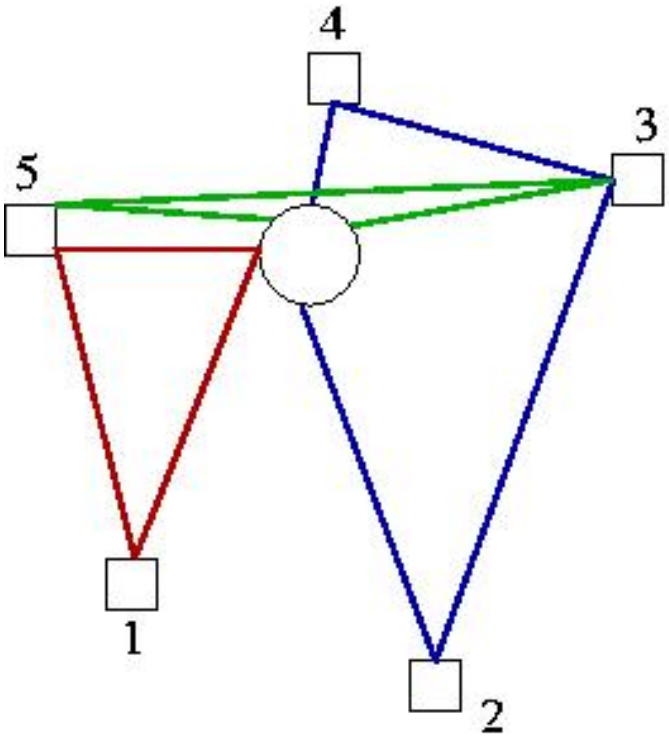
- Can you formulate it? What are your decision variables?
- What are your constraints?
- What if each customer had some "time windows" $[t_i^-, t_i^+]$ that the routes had to obey?
- What if each route had some "fixed cost" or was a nonlinear function of the distance traveled?

A New Formulation Idea

- Let there be a variable for every possible route $r \in R$

$$x_r = \begin{cases} 1 & \text{Travel on route } r \\ 0 & \text{Otherwise} \end{cases}$$

Vehicle Routing

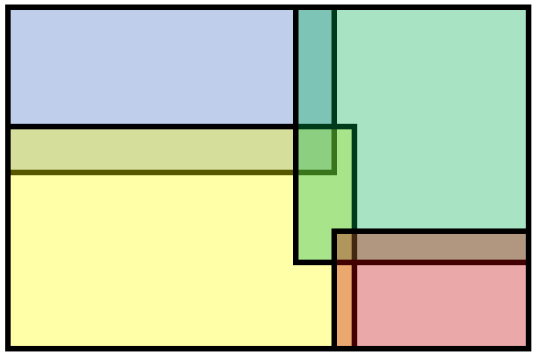


	x_1	x_2	x_3	$\cdots$		
Customer 1 :	1	0	0	$\vdots$	$\geq$	1
Customer 2 :	0	1	0	$\vdots$	$\geq$	1
Customer 3 :	0	1	1	$\vdots$	$\geq$	1
Customer 4 :	0	1	0	$\vdots$	$\geq$	1
Customer 5 :	1	0	1	$\vdots$	$\geq$	1

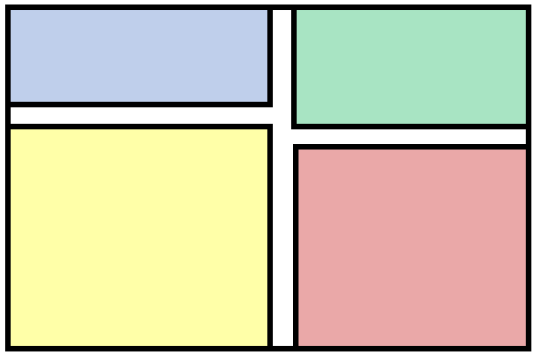
Set Covering/Packing/Partitioning

- If A is a matrix consisting only of 0's and 1's, then
- **Set Covering**: $\min c^T x : Ax \geq 1, x \in \{0,1\}^n$
- **Set Packing**: $\max c^T x : Ax \leq 1, x \in \{0,1\}^n$
- **Set Partitioning**: $\min c^T x : Ax = 1, x \in \{0,1\}^n$

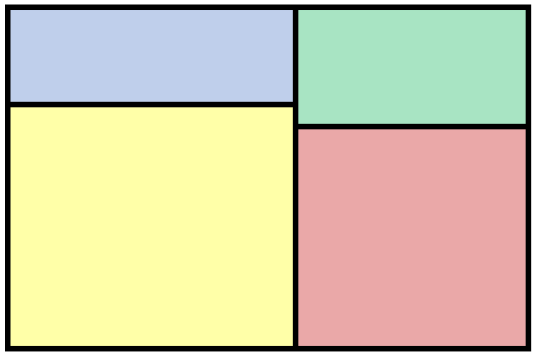
Set covering



Set packing



Set partitioning



Another Example – Flight Crew Scheduling

- Determine a minimum cost set of pilots to use so that all flights are "covered"
- The rules that constitute a feasible trip (called a pairing) for a set of pilots are complicated
- The amount of money that a pilot is paid for his pairing is also a complicated function of flying time, time away from home, and a minimum trip guarantee.
- How do they do it?
  - The trick is to just list all (or most) of the feasible pairings and their costs

---

Set Covering

- Rows: Flight Legs
- Columns: "Pairings": Subsets of flight legs, starting and ending in home base, that obey FAA flying regulations

A cutting stock problem

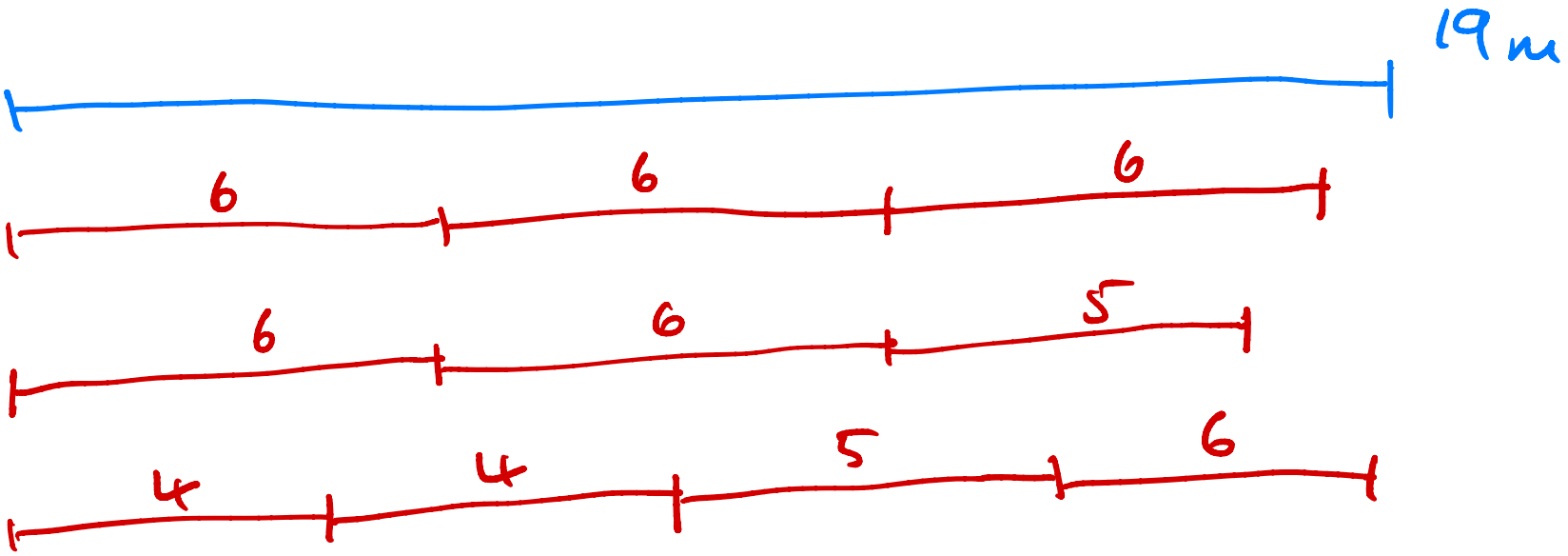
A plumber stocks standard lengths of pipe, all of length 19 m.

An order arrives for:

- 12 lengths of 4m
- 15 lengths of 5m
- 22 lengths of 6m

How should these lengths be cut from standard stock pipes so as to minimize the number of standard pipes used?

How should we model this?


19 m
6
6
6
6
6
5
4
4
5
6

Another cutting example

One possible model:

- Let N be an upper bound on the number of standard pipes. One possible upper bound is $N = 16$ (3 pipes per standard).
- Let $z_j = 1$ if we end up cutting standard pipe j.
- Let x_{ij} = number of pipes of length i to cut from pipe j.
- If $x_{ij} > 0$ then $z_j = 1$ (fixed cost).
- Obvious upper bounds: $x_{1j} \leq 4$, $x_{2j} \leq 3$, and $x_{3j} \leq 3$.

Another cutting example: Model 1

$$\min_{x,z} \quad \sum_{j=1}^{N} z_j \qquad \text{(total number of pipes cut)}$$

$$\text{s.t.} \quad 4x_{1j} + 5x_{2j} + 6x_{3j} \leq 19 \quad \text{for } j = 1, \ldots, N \qquad \text{(cuts are feasible)}$$

$$\sum_{j=1}^{N} x_{1j} \geq 12, \quad \sum_{j=1}^{N} x_{2j} \geq 15, \quad \sum_{j=1}^{N} x_{3j} \geq 22 \qquad \text{(orders must be met)}$$

$$x_{1j} \leq 4z_j, \quad x_{2j} \leq 3z_j, \quad x_{3j} \leq 3z_j \quad \forall j \qquad \text{(fixed cost constraints)}$$

$$x_{ij} \geq 0 \text{ integer}, \quad z_j \in \{0, 1\}$$

- A lot of symmetry. We can break the symmetry by using for example the constraint: $z_1 \geq z_2 \geq \cdots \geq z_N$.
- But should we? Symmetry was good for Sudoku, even though we had to use more variables...

Moving forward

Downsides of the first model:

- very large space with a lot of redundancy
- solve time scales poorly with problem size
- (scaling the demand should just scale the solution!)

Observations:

- The optimal solution will consist of patterns, such as $(5+6+6)$ or $(4+4+5+6)$.
- Even if there are many possible patterns, the optimal solution will only use a few different ones.
- Can we take advantage of these facts?

Enumerate all the possibilities!

- This enumeration can be tedious, but the problem is easy to solve once we have figured it out (essentially a set covering problem)
- Not a feasible method if we have many (100s) of patterns.
- If we use fewer columns (fewer patterns), the solution will still be an upper bound on the optimal one.
- This approach is also useful for problems such as flight crew scheduling: list the possible crew pairings with their associated costs, and solve the set cover problem.

Model

- There are only 9 different ways to cut the pipe:

$6+6+6$	$6+6+5$	$6+6+4$
$6+5+5$	$6+5+4+4$	$5+5+5+4$
$5+5+4+4$	$5+4+4+4$	$4+4+4+4$

- Each cut pattern is a column of this matrix:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 2 & 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 2 & 1 & 3 & 2 & 1 & 0 \\ 3 & 2 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\min_x \sum_{j=1}^{9} x_j \qquad \text{(how many times each pattern is used)}$$

$$\text{s.t. } Ax \geq \begin{bmatrix} 12 \\ 15 \\ 22 \end{bmatrix} \qquad \text{(orders must be met)}$$

$$x_J \geq 0 \qquad \text{(and integer)}$$

Larger instances

- If we had much larger pieces of standard pipe, there would be a very large number of possible patterns.
- No longer feasible to enumerate them all
- We shouldn't have to! Each column represents a potential vertex. Most will lie inside the feasible polyhedron.
- Column generation strategy:
  - Pick a small number of columns to start
  - Solve the LP relaxation
  - Use the dual to help pick a new column
  - Add the column to the A matrix and repeat

Examples: `cutstock.gms`

- To implement the column generation strategy, see `cutstock.gms` file and run on three data sets using `gams cutstock --data=1` (2 or 3)
- Example data above corresponds to `--data=3`

Column generation

1. Start with columns $A = [a_1, \ldots, a_k]$
2. Solve RMIP: minimize $\sum_j x_j$ subject to $Ax \geq d$.
3. Obtain dual variable λ^* to the $Ax \geq d$ constraint.
4. If we add a new column $\tilde{a}$, cost will drop by $\tilde{a}^T \lambda^*$.
   - Suppose $\tilde{a}^T = [y_1\ y_2\ y_3]^T$ is the new column.
   - Solve MIP: maximize $\tilde{a}^T \lambda^*$ subject to $4y_1 + 5y_2 + 6y_3 \leq 19$.
   - Add new column to A: $a_{k+1} = \tilde{a}^*$
5. Repeat from step 2.

Column generation can be an effective strategy when it's impossible to enumerate all possible columns. Only columns that are deemed likely to help reduce the cost are added.