CS524: Introduction to Optimization
Lecture 18

Michael Ferris

Computer Sciences Department
University of Wisconsin-Madison

October 21, 2019

Vehicle Routing

- 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?

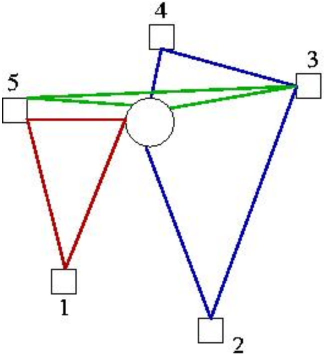
- Can you formulate it? What are your decision variables?
- What are your constraints?
- What if each customer had some "time windows" $[t_i^-, t_i^+]$ that the routes had to obey?
- What if each route had some "fixed cost" or was a nonlinear function of the distance traveled?

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	$\ldots$		
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

- Suppose we are given a set of objects $S = \{1, 2, \ldots m\}$.
- We are also given a set $\mathcal{S}$ of subsets of S.
- $S = \{1, 2, 3, 4, 5, 6\}$
  - $\mathcal{S} = \{\{1, 2\}, \{1, 3, 5\}, \{1, 2, 4, 5\}, \{4, 5\}, \{3, 6\}\}$
- The problem is to choose a set of subsets (from $\mathcal{S}$) so that all of the members of S are "covered".
- The set $C_1 = \{\{1, 2\}, \{1, 2, 4, 5\}, \{3, 6\}\}$ is a cover
- We may be interested in finding a cover of minimum cost
- Let $x_i = 1$ if the ith member of $\mathcal{S}$ is used in the cover.

Formulation (of previous example)

minimize

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

subject to

$$\begin{array}{ccccccccccc}
x_1 & + & x_2 & & x_3 & & & & & \geq & 1 \\
x_1 & & & + & x_3 & & & & & \geq & 1 \\
& & x_2 & & & & & & x_5 & \geq & 1 \\
& & & & x_3 & + & x_4 & & & \geq & 1 \\
& & x_2 & + & x_3 & & x_4 & & & \geq & 1 \\
& & & & & & & & x_5 & \geq & 1 \\
& & & & & & & & x_i & \in \{0,1\} & \forall i
\end{array}$$

More Set Covering: Kilroy County

- 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.

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

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**: $\min c^T x : Ax = 1, x \in \{0,1\}^n$

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

The Farmer's Daughter

- A traveling salesman must visit all his cities at minimum cost
- Must also start and end at home

Given

- Set of cities N
- Distances between cities c_{ij}

- This is the most famous combinatorial optimization problem
- There are lots of applications

$$x_{ij} = \begin{cases} 1 & \text{We travel from city } i \text{ to city } j \\ 0 & \text{Otherwise} \end{cases}$$

A Formulation for TSP?

- Consider the following

$$
\begin{aligned}
\min \sum_{i \in N} \sum_{j \in N} & c_{ij} x_{ij} \\
\sum_{i \in N} x_{ij} &= 1 \quad \forall j \in N \\
\sum_{j \in N} x_{ij} &= 1 \quad \forall i \in N \\
x_{ij} &\in \{0, 1\}
\end{aligned} \tag{1}
$$

- Is this a valid TSP formulation?

10 City USA Instance

```
table      dist(i,j)      "distances"
           Atlanta Chicago   Denver Houston LosAngeles Miami NewYork SanFrancisco Seattle WashingtonDC
Atlanta      0       587      1212    701       1936     604     748       2139       2182     543
Chicago      587      0       920     940       1745     1188     713       1858       1737      597
Denver       1212     920      0      879        831     1726    1631        949       1021     1494
Houston      701     940      879      0        1372     968    1420       1645       1891     1220
LosAngeles   1936     1745     831     1374        0      2339    2451        347        959     2300
Miami        604     1188     1726     968       2339     0      1092       2594       2734     923
NewYork      748      713     1631     1420      2451     1092     0         2571       2408     205
SanFrancisco 2139    1858     949     1645       347     2594    2571         0         678     2442
Seattle      2182     1737     1021    1891       959     2734    2408        678         0      2329
WashingtonDC  543     597     1494    1220       2300     923      205       2442        2329     0 ;
```

Subtours

- How do we get rid of those pesky subtours?
- For every set S of cities except the whole set, we can use at most $|S| - 1$ edges.

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subseteq N, 2 \leq |S| \leq |N| - 1$$

- There are exponentially many "subtour elimination" constraints

Quiz Time!

1018517988167243043134222844204689080525734196832968125318070224677190649881668353091698688.

- Is this...
  (a) The number of gifts that I have bought my wife?
  (b) The number of subatomic particles in the universe?
  (c) The number of "subtour elimination constraints" when $|N| = 299$.
  (d) All of the above?
  (e) None of the above?

Answer Time!

- The answer is (e). (a)–(c) are all too small (as far as I know) `:-)`. (It is (c), for $|N| = 300$).
- "Exponential" is really big.
- Yet people have solved TSP's with $|N| > 80,000$!
- How? You ask...
- Branch and cut! (`www.tsp.gatech.edu`)
  - Solve the problem without all of those subtours, and then just add them as we need them
- Finding the "add we need them" subtours is called the *separation problem*
- Let's finish solving our example...
  - See tsp1.gms in GAMS model library for fancier code: doing it dynamically (Other examples: tsp2,tsp3,tsp4,tsp5,swath)

Miller Tucker Zemlin

- There is a weaker, but more compact formulation that should suffice to eliminate subtours for smallish instances.
- We can derive it from the Slide of Trix.
- Idea: Let $u_i, i \in N$ be the relative positive of node i in the tour.
- $1 \leq u_i \leq n = |N| \quad \forall i \in N$
- Just model implication (assuming $x_{ii} = 0$ is forced):

$$x_{ij} = 1 \Rightarrow u_j \geq u_i + 1 (i.e. u_i - u_j \leq -1) \ \ (\forall i \neq 1, \forall j \neq 1)$$

$$\delta = 1 \Rightarrow \sum_{j \in N} a_j x_j \leq b \Leftrightarrow \sum_{j \in N} a_j x_j + M\delta \leq M + b.$$

- $M = n - 1 + 1 = n$
- Gives inequalities

$$u_i - u_j + n x_{ij} \leq n - 1 \quad \forall i \neq 1, \forall j \neq 1$$

Stronger version

To exclude subtours, one can use extra variables $u_i (i = 1, \ldots, n)$, and the constraints

$$\begin{aligned} & u_1 = 1, \\ & 2 \leq u_i \leq n, \quad \forall i \neq 1, \\ & u_i - u_j + 1 \leq (n-1)(1 - x_{ij}), \quad \forall i \neq 1, \forall j \neq 1. \end{aligned} \tag{2}$$

We call the last inequality in (2) an arc-constraint. The formulation consisting of (1) and (2) is called the MillerTuckerZemlin (MTZ) formulation of the TSP. It indeed excludes subtours, as (1) the arc-constraint for (i, j) forces $u_j \geq u_i + 1$, when $x_{ij} = 1$; (2) if a feasible solution of (1)-(2) contained more than one subtour, then at least one of these would not contain node 1, and along this subtour the u_i values would have to increase to infinity. This argument, with the bounds on the u_i variables, also implies that the only feasible value of u_i is the position of node i in the tour.