IEM 5013: Introduction to Optimization
Integer Linear Optimization Formulations

Dr. Baski Balasundaram

Professor
School of Industrial Engineering & Management
Oklahoma State University
Stillwater, OK
baski@okstate.edu

September 7, 2023

Chapters

Types of integer linear optimization (ILO) problems

- When all the variables are continuous we have an LO problem;
- When some variables are continuous and some are restricted to be integers, we have a mixed integer linear optimization (MILO) problem;
- When all the variables are required to be integers, we have a pure ILO problem;
- A special case is when the integer restriction limits the variables to binary $\{0, 1\}$ values, called a binary ILO problem.

Some well-known problems

- Capital budgeting problem aka knapsack problem
- Facility location problem
- Traveling salesperson problem
- Vehicle routing problem
- Job-shop and flow-shop scheduling problems
- Assembly line balancing problem
- Fixed-charge network flow problem
- ... and many others can be modeled as MILO problems, in particular, with the use of binary variables.

Knapsack problem

Capital budgeting. Suppose a company wishes to undertake from n independent projects subject to a total budget b, those that maximize cumulative profits. The initial investment required to initiate project i is given by c_i and the profit (assumed to be net present worth over the life of the project) is p_i. A project cannot be partially undertaken, it is either selected or rejected. Formulate this as an ILO problem.

Knapsack problem. Help a burglar select a subset of n items, each of size c_i, that fits in a knapsack of size b and maximizes the total value of the stolen items– each item is worth p_i.

Knapsack problem

Decision Variables.

$x_i = 1$ if item i is chosen, and 0 otherwise, $i = 1, \ldots, n$.

$$\max \sum_{i=1}^{n} p_i x_i$$

subject to:

$$\begin{aligned} \sum_{i=1}^{n} c_i x_i &\leq b \\ x_i &\in \{0,1\}, \quad i = 1, \ldots, n \end{aligned}$$

Other types of "side" constraints:

- At most k items can be chosen: $\sum_{i=1}^{n} x_i \leq k$
- If item i is chosen, then item j must be chosen: $x_i \leq x_j$
- If item i and item j are chosen, then item k must be chosen: $x_i x_j \leq x_k$, when linearized results in $x_i + x_j - 1 \leq x_k$

Assignment problem

There are n people available to perform n jobs. Each person is to be assigned to exactly one job. Each person charges different amounts perform each tasks. The cost of person i to carry out job j is c_{ij} for $i = 1, \ldots, n$ and $j = 1, \ldots, n$. Formulate an ILO model to find the minimum cost assignment.

Assignment problem

Decision Variables.

$x_{ij} = 1$ if person i is assigned to job j, and 0 otherwise, $i = 1, \ldots, n$, $j = 1, \ldots, n$.

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\begin{aligned}
\text{subject to: } \sum_{j=1}^{n} x_{ij} &= 1, \quad \forall i = 1, \ldots, n \\
\sum_{i=1}^{n} x_{ij} &= 1, \quad \forall j = 1, \ldots, n \\
x_{ij} &\in \{0,1\}, \quad \forall i, j = 1, \ldots, n
\end{aligned}$$

Note: We can, in this very special case, replace the binary restriction with $0 \leq x_{ij} \leq 1$, solve the resulting LP by the simplex method and still get binary solutions!

Warehouse location problem

Given a set of n sites for locating warehouses and a set of m clients that need to be served, we wish to select k out of the n sites to build the warehouses so that the demand d_j of customer j is satisfied, and the total cost is a minimum.

There is a fixed initial investment o_i for opening a warehouse at site i, there is a unit operating cost p_i at warehouse i for every unit shipped out, and there is a unit transportation cost c_{ij} for every unit transported from site i to site j.

A warehouse at site i also has a capacity limit u_i on the number of units it can ship out. Formulate an ILO model of this problem.

Warehouse location problem

Decision variables: $y_i = 1$ if site i is chosen to build a warehouse and 0 otherwise, $i = 1, \ldots, n$.
x_{ij} is the quantity shipped from the warehouse in site i to client j, for $i = 1, \ldots, n,\ j = 1, \ldots, m$.

$$\min \sum_{i=1}^{n} o_i y_i + \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} + \sum_{i=1}^{n} p_i \sum_{j=1}^{m} x_{ij}$$

$$\text{subject to:} \quad \sum_{i=1}^{n} y_i \leq k$$

$$\sum_{i=1}^{n} x_{ij} \geq d_j \qquad \forall j = 1, \ldots, m$$

$$\sum_{j=1}^{m} x_{ij} \leq u_i y_i \qquad \forall i = 1, \ldots, n$$

$$y_i \in \{0, 1\} \qquad \forall i = 1, \ldots, n$$

$$x_{ij} \geq 0 \qquad \forall i = 1, \ldots, n,\ j = 1, \ldots, m$$

The p-median problem

Given a set of n points and the distance d_{ij} between any two points i and j, find a set of p medians such that the total distance from any non-median point to its nearest median is minimized. Formulate an ILO model of this problem.

The p-median problem

Decision variables:

$x_{ij} = 1$ if point i is assigned to median j and 0 otherwise, $i = 1, \ldots, n,\ j = 1, \ldots, n,\ j \neq i$.

$x_{jj} = 1$ of j is a median and 0 otherwise, $j = 1, \ldots, n$.

$$\min \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} d_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{jj} = p$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i = 1, \ldots, n$$

$$x_{ij} \leq x_{jj} \qquad \forall i, j = 1, \ldots, n, i \neq j$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j = 1, \ldots, n$$

Flow-shop scheduling

Consider a workshop with m machines and n jobs to complete. The processing time of job i on machine j is given by p_{ij}. Every job goes through the machines in a fixed order (assume 1,2,...,m).

A machine can process at most one job at a time, once started a job must be finished and no job cannot preempt another job ahead of it in the sequence. In other words, the sequence in which the jobs enter machine 1, will be the sequence in which they enter every other machine. Formulate an ILO model to find the optimal sequence of jobs that minimizes total completion time.

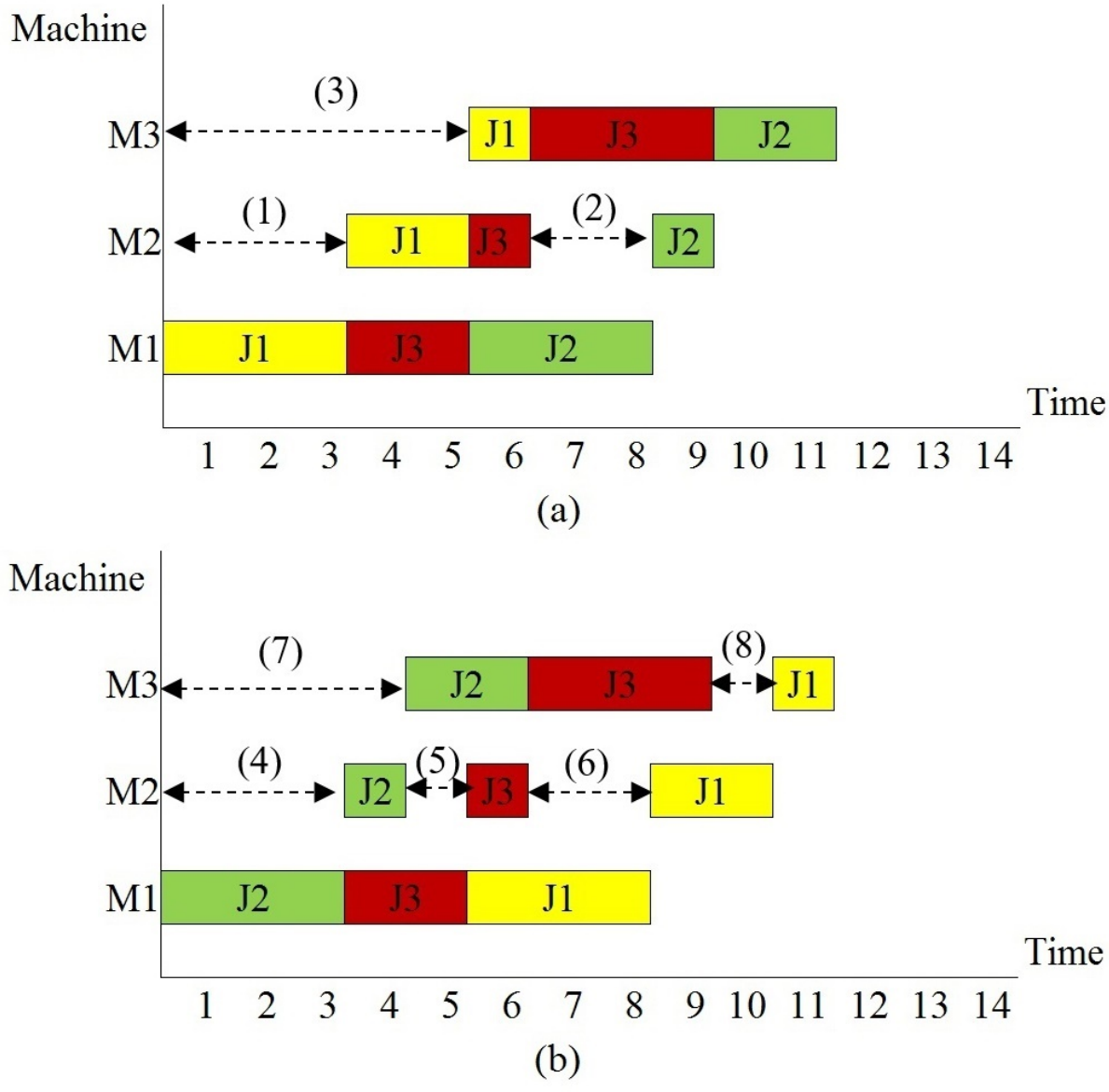


Image source: Utama et al. (2019) IJTech Vol 10, No 2.

Flow-shop scheduling

Indices: $i = 1, \ldots, n$ for job number; $j = 1, \ldots, n$ for the position in the sequence; $k = 1, \ldots, m$ for machine number

Decision variables: $z_{ij} = 1$ if job i is scheduled in position j and 0 otherwise; s_{jk} is the start time of job in position j on machine k

$$\min s_{nm} + \sum_{i=1}^{n} p_{im} z_{in}$$

s.t.

$$\sum_{i=1}^{n} z_{ij} = 1, \; j = 1, \ldots, n$$

$$\sum_{j=1}^{n} z_{ij} = 1, \; i = 1, \ldots, n$$

$$s_{11} = 0$$

$$s_{1,k+1} = s_{1,k} + \sum_{i=1}^{n} p_{ik} z_{i1}, \; k = 1, \ldots, m-1$$

$$s_{j+1,1} = s_{j,1} + \sum_{i=1}^{n} p_{i1} z_{ij}, \; j = 1, \ldots, n-1$$

Flow-shop scheduling

Consider s_{jk} for $j, k \geq 2$:

- Either the job in position j waits for machine k to finish processing the job in position $j-1$,
- Or machine k waits for the job in position j to finish on machine $k-1$.
- So the start time of job in position j on machine k is at least as large as the maximum of the following two quantities:
  - the end time of the job in position $j-1$ on machine k, and
  - the end time of the job in position j on machine $k-1$.

$$
\begin{aligned}
s_{jk} &\geq s_{j-1,k} + \sum_{i=1}^{n} p_{ik} z_{i,j-1},\ j = 2,\ldots,n,\ k = 2,\ldots,m \\
s_{jk} &\geq s_{j,k-1} + \sum_{i=1}^{n} p_{i,k-1} z_{ij},\ j = 2,\ldots,n,\ k = 2,\ldots,m \\
s_{jk} &\geq 0,\ j = 1,\ldots,n,\ k = 1,\ldots,m \\
z_{ij} &\in \{0,1\},\ i,j = 1,\ldots,n
\end{aligned}
$$

Job-shop scheduling

Consider a workshop that has to complete 3 jobs A,B,C on 4 machines. Each job must be processed on the machines in a particular order. Job A goes through machines 1, 3, 4; Job B goes through machines 1, 2, 4; Job C goes through machines 2,3.

Processing times of Job A,B,C on machine i are denoted by a_i, b_i, c_i respectively. Jobs cannot be split and machine cannot be interfered with. Formulate an ILO model that minimizes the time by which all jobs are completed.

Job-shop scheduling

Decision variables: Let x_{Ai}, x_{Bi}, x_{Ci} denote the start time of jobs A, B, C respectively on machine $i = 1, 2, 3, 4$. Let y denote the completion time of all the jobs.

A : 1, 3, 4
B : 1, 2, 4
C : 2, 3

$$\min y$$

subject to:

$$\begin{aligned}
y &\geq x_{A4} + a_4 \\
y &\geq x_{B4} + b_4 \\
y &\geq x_{C3} + c_3 \\
x_{A3} &\geq x_{A1} + a_1 \\
x_{A4} &\geq x_{A3} + a_3 \\
x_{B2} &\geq x_{B1} + b_1 \\
x_{B4} &\geq x_{B2} + b_2 \\
x_{C3} &\geq x_{C2} + c_2
\end{aligned}$$

These constraints make sure the job is started on a machine after it is finished on the previous machine. But nothing prevents two jobs being scheduled at the same time on a machine. For instance, $x_{A1} = x_{B1} = 0$ is permitted. Machine conflicts need to be resolved.

Job-shop scheduling

Machine preemption constraints: First identify all jobs that visit any given machine, for example on machine 1, A and B need to be scheduled.

Machine 1: A before B **OR** B before A

We need,
"$x_{B1} \geq x_{A1} + a_1$" **OR** "$x_{A1} \geq x_{B1} + b_1$"

We model such disjunctive constraints as follows: $\delta^1_{AB} = 1$ if A precedes B on machine 1, and 0 otherwise.

$$x_{A1} + a_1 - x_{B1} \leq M(1 - \delta^1_{AB})$$

$$x_{B1} + b_1 - x_{A1} \leq M\delta^1_{AB}$$

Job-shop scheduling

$$
\begin{aligned}
MC1: x_{A1} + a_1 - x_{B1} &\leq M(1-\delta^1_{AB}) \\
MC1: x_{B1} + b_1 - x_{A1} &\leq M\delta^1_{AB} \\
MC2: x_{B2} + b_2 - x_{C2} &\leq M(1-\delta^2_{BC}) \\
MC2: x_{C2} + c_2 - x_{B2} &\leq M\delta^2_{BC} \\
MC3: x_{A3} + a_3 - x_{C3} &\leq M(1-\delta^3_{AC}) \\
MC3: x_{C3} + c_3 - x_{A3} &\leq M\delta^3_{AC} \\
MC4: x_{A4} + a_4 - x_{B4} &\leq M(1-\delta^4_{AB}) \\
MC4: x_{B4} + b_4 - x_{A4} &\leq M\delta^4_{AB} \\
x_{Ai}, x_{Bi}, x_{Ci} &\geq 0, \quad i = 1,2,3,4 \\
\delta^1_{AB}, \delta^2_{BC}, \delta^3_{AC}, \delta^4_{AB} &\in \{0,1\}
\end{aligned}
$$

Any thoughts on how to set the value of the big-M?

What if:

A : 1, 3, 4

B : 1, 2, 4

C : 1, 2, 3

Assembly line balancing

In an assembly line that assembles multiple components into a final product, there are multiple machines that are grouped into stations where each station is handled by a single worker.

The product has to visit the machines while obeying the precedence requirement. Since the stations are arranged in a straight line, this amounts to ensuring that a predecessor of a particular machine is not assigned to a station after this machine.

Suppose you are given 10 machines, the processing time of the job carried out by each machine, and precedence requirements. Formulate an ILO model to minimize the line cycle time (maximum of station cycle times) while creating no more than 3 stations?

Another type of line balancing problem is to assign the jobs to a minimum number of stations such that the station cycle time in each station is at most 15. How does the formulation change in this case?

Assembly line balancing

Job	Proc. Time	Pred.
1	3	-
2	5	-
3	2	2
4	6	1,3
5	9	2
6	3	4,5
7	4	-
8	1	-
9	3	-
10	5	-

Assembly line balancing

- This is a system that repeatedly performs the same type of job, such as assembling components into a finished product, that uses all the machines in the assembly line.
- The problem here is to group the n machines into stations. Each station has a cycle time which is the sum of the processing times of machines in that station. The line cycle time is the largest of the station cycle times.
- The stations are assumed to be arranged in a straight line and each station is run by one person. The precedence requirements state that if machine i must precede machine j then i must be assigned to the same station as j, or to a station that precedes it.
- We are interested in two objectives- the line cycle time and the number of stations created.
- Type I problem minimizes the line cycle time while limiting the number of stations created to be at most K.
- Type II problem is to minimize the number of stations created while limiting to the line cycle time to at most T.

Type I: ALB

Parameters: K denotes the maximum number of stations allowed; p_i is the processing time on machine i.
Decision variables: $x_{ij} = 1$ if machine i is assigned to station j, $i = 1, \ldots, n,\ j = 1, \ldots, K$ and 0 otherwise.
T denotes line cycle time.

$$\min T$$

s.t.

$$T \geq \sum_{i=1}^{n} p_i x_{ij}, \quad \forall j = 1, \ldots, K$$

$$\sum_{j=1}^{K} x_{ij} = 1, \quad \forall i = 1, \ldots, n$$

$$\sum_{j=1}^{K} j x_{uj} \leq \sum_{j=1}^{K} j x_{vj} \quad \forall u, v \text{ such that } u \text{ precedes } v$$

$$x_{ij} \in \{0, 1\} \quad \forall i = 1, \ldots, n,\ j = 1, \ldots, K$$

Type II: ALB

Parameters: T denotes the maximum line cycle time allowed; p_i is the processing time on machine i.

Decision variables: $x_{ij} = 1$ if machine i is assigned to station j, $i, j = 1, \ldots, n$ and 0 o.w.

$y_j = 1$ if station j is created and 0 otherwise, $j = 1, \ldots, n$.

$$\min \sum_{j=1}^{n} y_j$$

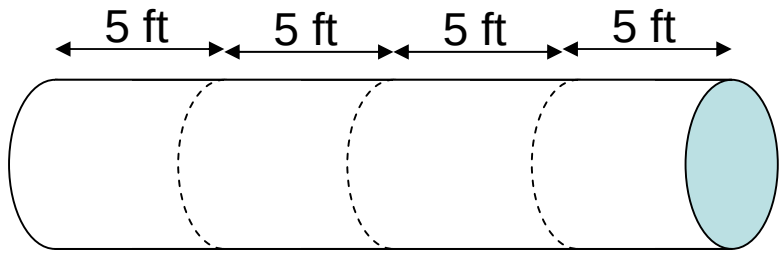
$$\begin{aligned}
\text{s.t.} \quad \sum_{i=1}^{n} p_i x_{ij} &\leq T, \quad \forall j = 1, \ldots, n \\
\sum_{j=1}^{n} x_{ij} &= 1, \quad \forall i = 1, \ldots, n \\
\sum_{j=1}^{n} j x_{uj} &\leq \sum_{j=1}^{n} j x_{vj} \quad \forall u, v \mid u \text{ precedes } v \\
x_{ij} &\leq y_j, \quad \forall i, j = 1, \ldots, n \\
x_{ij} &\in \{0,1\}, \quad \forall i, j = 1, \ldots, n \\
y_j &\in \{0,1\}, \quad \forall j = 1, \ldots, n
\end{aligned}$$

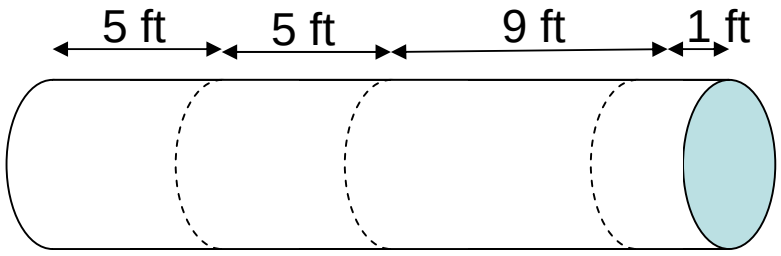
Cutting Stock Problem

A paper company produces paper rolls with a standard width of 20 feet each and a standard length of L feet. Special customer orders with different widths are produced by cutting the stock rolls. Typical daily demands for these non-standard width rolls are summarized below.

Order	Desired width (ft)	Demand (no. of rolls)
1	5	150
2	7	200
3	9	300

These orders are filled by cutting the stock roll to the desired widths using different knife settings. For instance, they could set the knives to cut a standard roll into 4 pieces, each 5 ft wide. Alternately, they could cut a stock piece into 4 pieces, two 5 ft wide, one 9 ft wide, and the remaining 1 ft wide piece is waste, called trim loss (see figure).





The company is free to choose any number of different knife settings, and needs to decide the number of stock pieces to be cut according to each knife setting, so that the cumulative trim loss, plus any order type (1,2, or 3) produced in excess of the demand, is minimized. Formulate an IP model for the company's problem.

Cutting Stock Problem

Index sets:

$J = \{1, 2, 3, 4, 5, 6\}$ index set for knife settings (patterns) information given in the table below

Pattern No.	1	2	3	4	5	6
No. of 5'	4	1	0	2	2	0
No. of 7'	0	2	0	1	0	1
No. of 9'	0	0	2	0	1	1
waste of pattern no. $j \in J$	0	1	2	3	1	4

Each pattern vector captures the no. of 5' rolls produced in the first component, no. of 7' rolls produced in the second component, and number of 9' rolls produced in the third component.

Cutting Stock Problem

Parameters:

w_j denotes trim-loss of knife setting (pattern) $j \in J$.

Decision Variables:

x_j denotes the number of stocks cut according to knife setting $j \in J$

$$\min \sum_{j \in J} w_j x_j + 5s_1 + 7s_2 + 9s_3$$

subject to:

$$\begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix} x_1 + \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} x_2 + \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} x_3 + \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} x_4 + \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} x_5 + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} x_6 - \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 150 \\ 200 \\ 300 \end{bmatrix}$$

$$x_j \geq 0 \text{ and integer}, \forall j \in J$$

$$s_1, s_2, s_3 \geq 0$$

Why, for the first time, have we written this particular model columnwise?

Traveling salesperson problem (TSP)

Given a set of n cities and pairwise distances $c_{ij}, i, j = 1, \ldots, n$, the TSP seeks to find a "tour" of the n cities with minimum total distance in which the salesperson visits every city exactly once starting from their home city. Formulate an ILO model of this problem.

Traveling salesperson problem (TSP)

Decision Variables:

$x_{ij} = 1$ if city j is visited immediately after city i in the tour and 0 otherwise, $i, j = 1, \ldots, n$ and $i \neq j$

$$\min \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} c_{ij} x_{ij}$$

subject to:

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} = 1 \qquad \forall i = 1, \ldots, n$$

$$\sum_{\substack{i=1 \\ j \neq i}}^{n} x_{ij} = 1 \qquad \forall j = 1, \ldots, n$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j = 1, \ldots, n \text{ such that } i \neq j$$

Is that it, or do we need more constraints?

Traveling salesperson problem (TSP)

Subtour Elimination Constraints (SEC): Any one of the following groups of constraints would do the job. (Note that U is a strict subset of the cities, i.e., U is never equal to all the cities.)

- Cut Covering.

$$\sum_{i \in U} \sum_{j \notin U} x_{ij} \geq 1, \quad \forall U \subset \{1, \ldots, n\} \text{ and } U \neq \emptyset$$

- Cycle Prevention.

$$\sum_{i \in U} \sum_{\substack{j \in U \\ i \neq j}} x_{ij} \leq |U| - 1, \quad \forall U \subset \{1, \ldots, n\} \text{ and } U \neq \emptyset$$

Review Miller-Tucker-Zemlin SEC from the textbook!

Multiple TSP

The multiple traveling salesperson problem (MTSP) is a generalization of the TSP in which more than one salesperson is routed. Given a set of n cities including a home-base (depot) at city-1 where the k salespersons are located, and transportation cost c_{ij} between cities i and j, the objective of the MTSP is to determine a set of routes for the k salespersons so as to minimize the total cost of the k routes. The cost metric can represent cost, distance, or time. The requirements on the set of routes are:

- All of the routes must start and end at the home-base (depot).
- Each city must be visited exactly once **by only one salesperson**.

The MTSP is aka the Uncapacitated Vehicle Routing Problem (VRP).

MTSP

$$\min \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} c_{ij} x_{ij}$$

$$\text{subject to:} \quad \sum_{j=2}^{n} x_{1,j} = k$$

$$\sum_{i=2}^{n} x_{i,1} = k$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} = 1 \qquad \forall i = 2, \ldots, n$$

$$\sum_{\substack{i=1 \\ j \neq i}}^{n} x_{ij} = 1 \qquad \forall j = 2, \ldots, n$$

$$\sum_{i \in U} \sum_{j \notin U} x_{ij} \geq 1 \qquad \forall U \subseteq \{2, 3, \ldots, n\} \text{ and } U \neq \emptyset$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j = 1, \ldots, n \text{ such that } i \neq j$$

Capacitated Vehicle Routing Problem (CVRP)

The CVRP consists of n locations (a depot and a set of $n-1$ customers), c_{ij} specifying the distance (or some other cost) to travel between each pair of locations, a quantity q_i that specifies the demand for customer i, and the maximum capacity Q of the quantity that each of the k vehicles can carry.

A feasible solution to the CVRP consists of a set of routes that begin and end at the depot, such that each customer is visited on exactly one route and the total demand by the customers assigned to a route does not exceed the vehicle capacity Q.

An optimal solution for CVRP is a feasible solution that minimizes the total combined distance of the routes.

CVRP

$$\min \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} c_{ij} x_{ij}$$

subject to:

$$\sum_{j=2}^{n} x_{1,j} = k$$

$$\sum_{i=2}^{n} x_{i,1} = k$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} = 1 \qquad \forall i = 2, \ldots, n$$

$$\sum_{\substack{i=1 \\ j \neq i}}^{n} x_{ij} = 1 \qquad \forall j = 2, \ldots, n$$

$$\sum_{i \in U} \sum_{j \notin U} x_{ij} \geq \left\lceil \sum_{i \in U} q_i / Q \right\rceil \qquad \forall U \subseteq \{2, 3, \ldots, n\} \text{ and } U \neq \emptyset$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j = 1, \ldots, n \text{ such that } i \neq j$$

Some closing remarks ...

- There are numerous variations of vehicle routing that are studied due to the wide application of the model and the variety of practical considerations.
- There are other ways to formulate CVRP and MTSP using flow variables, route variables, etc.
- There are many more classical integer/combinatorial/network optimization models we haven't looked at ...
  - Set covering, partitioning, and packing
  - Graph problems—matching, clustering, partitioning, covering, cliques, coloring, domination, ...
  - Network design problems—Steiner trees, spanning trees, ...
  - Reformulating nonconvex piecewise linear functions
  - Scheduling, staffing, and time-tabling problems
  - ...