Lecture 13 Mixed-Integer Linear Programming Applications

Can Li

ChE 597: Computational Optimization
Purdue University

Outline of this lecture

- Traveling salesman problem
- Uncapacitated facility location problem (k-medoids clustering)
- Uncapacitated Lot-Sizing (ULS)
- Scheduling problems
- Unit commitment
- Superstructure optimization for chemical process synthesis

The Traveling Salesman Problem (TSP)

- The TSP is a classic challenge in Operations Research, involving a salesman needing to visit n cities once and return to the start to minimize travel time.
- Applicable in various sectors, including logistics, manufacturing, and storage, underscoring its widespread relevance.

Defining the Variables and Objective

- Let $x_{ij} = 1$ if the salesman travels directly from city i to city j, and $x_{ij} = 0$ otherwise.
- Self-visits are excluded, i.e., x_{ii} is undefined for all i.
- The objective is to minimize the total travel time, formalized as:

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

- This function calculates the shortest possible route that visits each city once and returns to the starting point.
- If $c_{ij} = c_{ji}$: symmetric TSP. If not, asymmetric TSP.

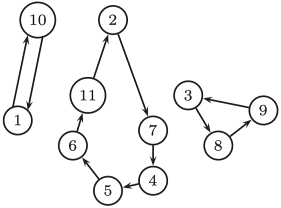
Constraints in the TSP

- Each city must be left and arrived at exactly once, ensuring every city is visited in the tour. These constraints are like assignment constraints.

$$\sum_{j:j\neq i} x_{ij} = 1 \qquad \text{for } i = 1, \ldots, n.$$

$$\sum_{i:i\neq j} x_{ij} = 1 \qquad \text{for } j = 1, \ldots, n.$$

- The following figure with subtours also satisfy the assignment constraints but is not a valid TSP tour.



Subtour Elimination Constraints

Objective

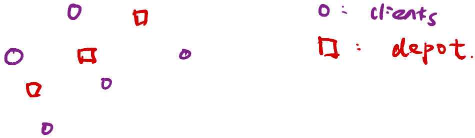
To prevent the formation of subtours, ensuring the salesman completes a single, continuous tour visiting all cities.

Constraint Formula

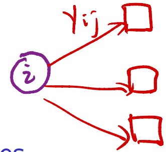
$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad \text{for } S \subset N, 2 \leq |S| \leq n - 1$$

- S is a subset of cities within the total set of cities N.
- This constraint limits the number of trips within a subset S to at most $|S| - 1$, effectively preventing any closed loops that don't include every city.
- It applies to all possible subsets of cities, except the trivial cases where $|S| < 2$ or $S = N$.
- It suffices to enforce for $S \subset N, 2 \leq |S| \leq \frac{1}{2}n$

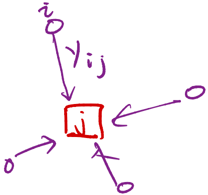
Uncapacitated Facility Location (UFL)



- The UFL problem involves determining optimal depot openings and client servicing to minimize fixed and transportation costs.
- Considers a set N of potential depots and a set M of clients, with associated fixed costs f_j for depots and transportation costs c_{ij} from depot j to client i.



Formulation



Variables

- $x_j = 1$ if depot j is used, $x_j = 0$ otherwise.
- y_{ij} represents the fraction of client i's demand satisfied from depot j.

Constraints

- Demand Satisfaction: $\sum_{j=1}^{n} y_{ij} = 1$ for each client $i = 1, \ldots, m$.
- Depot-Client Link: $\sum_{i \in M} y_{ij} \leq m x_j$, ensuring depots only serve clients if opened. $y_{ij} \leq x_j \quad \forall i \in M$
- Variables: $y_{ij} \geq 0$ for demand fractions, $x_j \in \{0, 1\}$ for depot openings.

Formulation

Objective Function

Minimize total fixed and transportation costs.

$$\min \sum_{i \in M} \sum_{j \in N} c_{ij} y_{ij} + \sum_{j \in N} f_j x_j$$

- This combines both depot operational costs and client delivery costs for an optimal solution.

Connections with *k*-medoids clustering

The UFL problem and *k*-medoids clustering share a similar structure, where both aim to choose locations (depots or medoids) that minimize the total distance or cost to serve a set of points (clients). In *k*-medoids, each cluster is represented by one of the actual data points (medoids), analogous to selecting depots in UFL.

Alternative formulation

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} y_{ij} + \sum_{j=1}^{n} f_j x_j$$

$$\sum_{j=1}^{n} y_{ij} = 1 \text{ for } i \in M$$

$$y_{ij} \leq x_j \text{ for } i \in M, j \in N$$

$$y_{ij} \geq 0 \text{ for } i \in M, j \in N, \quad x_j \in \{0,1\} \text{ for } j \in N.$$

- Larger number of constraints
- Tighter than the big-M formulation.

Introduction to Uncapacitated Lot-Sizing (ULS)

- The ULS problem involves deciding on a production plan for a single product over an n-period horizon.
- Parameters
  1. fixed production costs (f_t)
  2. unit production costs (p_t)
  3. unit storage costs (h_t)
  4. demand (d_t)
  5. production capacity (C)

Model Formulation and Variables

- Variables:
  - y_t: Amount produced in period t.
  - s_t: Stock at the end of period t.
  - $x_t = 1$ if production occurs in t, 0 otherwise.
- Balance the trade-off of storage v.s. production cost.

Objective and Constraints

Objective Function

Minimize total production, storage, and fixed costs:

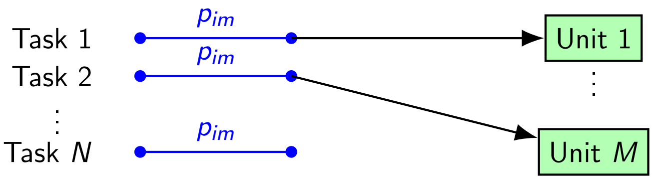
$$\min \sum_{t=1}^{n} p_t y_t + \sum_{t=1}^{n} h_t s_t + \sum_{t=1}^{n} f_t x_t$$

Constraints

- Balance equation: $s_{t-1} + y_t = d_t + s_t$ for $t = 1, \ldots, n$.
- Production limit: $y_t \leq C x_t$ for $t = 1, \ldots, n$, can be tightened if $s_n = 0$.
- Initial and variable constraints: $s_0 = 0$, $s_t, y_t \geq 0$, $x_t \in \{0, 1\}$.

Scheduling: Single-stage parallel units

- Given N tasks (i) and M units/machines (m)
- Tasks (Jobs/Orders) = fixed processing times p_{im}
- Find how to assign tasks j to machine m to minimize completion time (makespan) MS.



MILP Formulation: Objective and Notation

Objective

Minimize the makespan (MS), which is the total time to complete all tasks. The makespan is defined as:

$$\min \quad MS$$

Notation

- x_{im}: Binary variable, 1 if task i is assigned to unit m.
- p_{im}: Parameter, processing time of task i on unit m.

MILP Formulation: Constraints and Their Explanations

Constraints

- **Total Processing Time Constraint:** The total processing time for tasks assigned to a unit m must not exceed the makespan (MS).

$$\sum_i p_{im} x_{im} \leq MS \quad \forall m$$

The amount of time to complete the jobs on machine m.

- **Task Assignment Constraint:** Each task i must be assigned to exactly one unit m.

$$\sum_m x_{im} = 1 \quad \forall i$$

- **Non-Negativity and Binary Constraints:** The makespan (MS) must be non-negative, and the task assignment variables x_{im} are binary.

$$MS \geq 0, \, x_{im} \in \{0, 1\}$$

Jobs with start and due dates

- Given N tasks (i) and M units (m)
- Tasks (Jobs/Orders) = fixed processing times p_{im}
- Jobs have their release dates and due dates, i.e., a job cannot start before its release date and has to be completed before its due date.
- Find how to assign tasks j to machine m to minimize the production cost while satisfying the release and due dates constraints.

Task Scheduling Model

Decision Variables

- $x_{im} = \begin{cases} 1 & \text{if task } i \text{ is assigned to unit } m \\ 0 & \text{otherwise} \end{cases}$
- ts_i = start time of task i
- $y_{ii'} = 1$ if task i before task i' on given unit; 0 otherwise

Objective

Minimize the cost of processing tasks:

$$\min \sum_{i \in I} \sum_{m \in M} c_{im} x_{im}$$

Constraints

- **Earliest Start:** Start times must respect the earliest start constraints.
  $ts_i \geq r_i$ for all $i \in I$. r_i : release time
- **Latest Start:** Tasks must start early enough to finish before their deadlines.
  $ts_i \leq d_i - \sum_{m \in M} p_{im} x_{im}$ for all $i \in I$.
- **Assign to One Unit:** Each task is assigned to exactly one unit.
  $\sum_{m \in M} x_{im} = 1$ for all $i \in I$.
- **Redundant Constraint:** This ensures processing times are within a feasible range. This constraint tighten the LP relaxation.
  $\sum_{i \in I} p_{im} x_{im} \leq \max\{d_i\} - \min\{r_i\}$ for all $m \in M$.

Task Sequencing and Timing Constraints

$(x_{im} \wedge x_{i'm}) \Rightarrow (y_{ii'} \vee y_{i'i})$

$\underbrace{(x_{im} \wedge x_{i'm})}$ both i and i' are assigned to machine m.

$A \Rightarrow B$

$\neg A \vee B$

$\neg(x_{im} \wedge x_{i'm}) \vee (y_{ii'} \vee y_{i'i})$

$\Downarrow$ De-Morgan

$(\neg x_{im} \vee \neg x_{i'm}) \vee (y_{ii'} \vee y_{i'i})$

$1 - x_{im} + 1 - x_{im} + y_{ii'} + y_{i'i} \geq 1$.

- Precedence constraint: If x_{im} AND $x_{i'm}$ true then $y_{ii'}$ OR $y_{i'i}$ are true.
  $y_{ii'} + y_{i'i} \geq x_{im} + x_{i'm} - 1$ for all i, i' in I, $i > i'$, m in M
- Sequencing constraint: If $y_{ii'} = 1$ then $ts_{i'} \geq ts_i + p_{im}$.
  $ts_{i'} \geq ts_i + \underbrace{\sum_{m \in M} p_{im} x_{im}}_{\text{processing time of job } i.} - M(1 - y_{ii'})$ for all i, i' in I, $i \neq i'$.

Unit Commitment Problem

- The Unit Commitment (UC) problem involves determining the start-up and shut-down schedules of power generating units to meet demand over a planning horizon while minimizing the total operational cost, subject to system constraints.
- Solved by power systems operators worldwide

Unit Commitment Problem: Parameters and Variables

Parameters

- C_g^{F}: Fixed cost of generating unit g.
- C_g^{V}: Variable cost of power produced by unit g.
- C_g^{SU}: Start-up cost of generating unit g.
- C_g^{SD}: Shut-down cost of generating unit g.
- P_g^{min} / P_g^{max}: Minimum/Maximum power output of unit g.
- R_g^{U} / R_g^{D}: Ramp up/down rate of unit g.
- P_t^{D}: Power demand at time t.
- R_t^{D}: Reserve demand at time t. (Reserve demand refers to the additional capacity that a power system must have available to handle sudden increases in power demand or unexpected drops in power supply.)

Variables

- u_{gt}: Binary variable, 1 if unit g is operational at time t.
- p_{gt}: Power output of unit g at time t.
- y_{gt}: Binary variable, 1 if unit g starts up at time t.
- z_{gt}: Binary variable, 1 if unit g shuts down at time t.

Objective Function and Constraints

Objective Function

Minimize the total operational cost of all generating units over the planning horizon:

$$\min \sum_t \sum_g \left(C_g^{\mathrm{F}} u_{gt} + C_g^{\mathrm{V}} p_{gt} + C_g^{\mathrm{SU}} y_{gt} + C_g^{\mathrm{SD}} z_{gt} \right)$$

Constraints

- Start-up and shut-down conditions:
  $y_{gt} - z_{gt} = u_{gt} - u_{g,t-1}$ for all g, t.
- A unit cannot start and shut down simultaneously:
  $y_{gt} + z_{gt} \leq 1$ for all g, t.
- Generation within limits:
  $P_g^{\mathsf{min}} u_{gt} \leq p_{gt} \leq P_g^{\mathsf{max}} u_{gt}$ for all g, t.

- Ramp-up and ramp-down rates:
  - $p_{gt} - p_{g,t-1} \leq R_g^{\mathrm{U}} u_{g,t-1} + R_g^{\mathrm{SU}} y_{gt}$ for all g, t,
  - $p_{g,t-1} - p_{gt} \leq R_g^{\mathrm{D}} u_{gt} + R_g^{\mathrm{SD}} z_{gt}$ for all g, t.
- Demand satisfaction:
  $\sum_g p_{gt} = P_t^{\mathrm{D}}$ for all t.
- Reserve requirement:
  $\sum_g P_g^{\max} u_{gt} \geq P_t^{\mathrm{D}} + R_t^{\mathrm{D}}$ for all t.
- Binary conditions for operational status and start-up/shut-down actions:
  $u_{gt}, y_{gt}, z_{gt} \in \{0, 1\}$ for all g, t.

Superstructure optimization for process synthesis

- Process synthesis in chemical engineering involves the design and creation of chemical processing systems, from the selection of reaction pathways to the configuration of unit operations, aiming to efficiently convert raw materials into desired products. It encompasses the development of optimal process flowsheets that meet specified performance, safety, environmental, and economic criteria.
- Superstructure-based process synthesis, involves three main sequential steps:
    1. the postulation of a superstructure, which encapsulates the set of all feasible alternative process structures;
    2. the translation of the superstructure into a mathematical programming model
    3. the computation of an optimal structure by solving the mathematical optimization model.

Distillation sequence synthesis

- Goal: identify the most cost effective way to separate a chemical mixture into pure components.
- Example: separate the mixture ABC, into pure components A, B, and C. There are different ways of designing the distillation columns.

right

A
B
C heavy

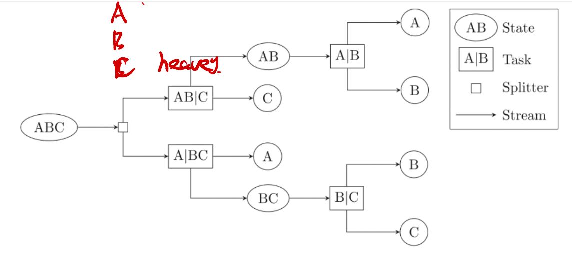
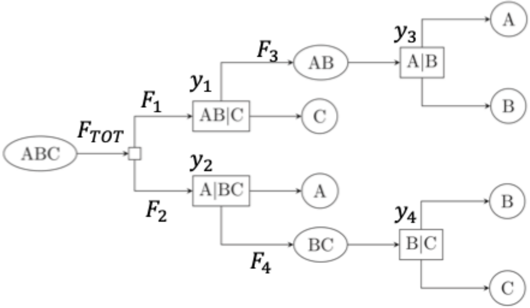


Figure: State-Task Network (STN) superstructure for a 3 component sharp split distillation sequence.

MILP model



- Variables: F_i flow rates. y_i binary variable denoting whether each column is installed.
- Constraints
  1. Mass balance, e.g. $F_{TOT} = F_1 + F_2$. $F_3 = \xi^{AB} * F_1$ where ξ represents the split fraction.
  2. Logic constraint: $F_k \leq U y_k$
  3. Hot and cold utility consumption: $Q_R^k = K_R^k F_k$ (reboiler), $Q_C^k = K_C^k F_k$ (condenser).

MILP model

$$\min C = \sum_{k \in \mathrm{COL}} \left(\alpha_k y_k + \beta_k F_k + C_H Q_R^k + C_C Q_c^k \right)$$

$$\text{s. t.} \quad \sum_{k \in FS_F} F_k = F_{TOT}$$

$$\sum_{j \in FS_m} F_j - \sum_{i \in PS_m} \xi_i^m F_i = 0, \quad m \in IP$$

$$Q_R^k - K_R^k F_k = 0$$

$$Q_C^k - K_C^k F_k = 0$$

$$F_k - U y_k \leq 0$$

$$k \in \mathrm{COL}$$

$$F_k, Q_R^k, Q_C^k \geq 0, y_k = \{0, 1\}, k \in \mathrm{COL}$$

References

- Grossmann, I. E. (2021). Advanced optimization for process systems engineering. Cambridge University Press.
- Conforti, M., Cornuéjols, G., Zambelli, G (2014). Integer programming. Graduate Texts in Mathematics
- Wolsey, L. A. (2020). Integer programming. John Wiley & Sons.
- Conejo, A. J., & Baringo, L. (2018). Power system operations (Vol. 11). New York: Springer.
- Mencarelli, L., Chen, Q., Pagot, A., & Grossmann, I. E. (2020). A review on superstructure optimization approaches in process system engineering. Computers & Chemical Engineering, 136, 106808.