OPTIMAL NETWORK DESIGN II: RAPID TRANSIT NETWORK DESIGN
Course 4 Modelling and optimization algorithms in networks design and energy planning

Juan A. Mesa
Department of Applied Mathematics II
University of Seville
Outline

• Data
• Variables
• Objectives
• Constraints
• GRASP algorithm
• The robust case
• Computational Complexity
Data

Over an area, where there already exists a transportation mode (bus, car) we have:

- A set $N = \{1, 2, \ldots, n\}$ of nodes representing potential sites for stations is given.
- A set $E = \{ij : i, j \in N; i < j\}$ of $m$ feasible edges linking the elements of $N$ is known.
- Every feasible edge $ij \in E$ has an associated length $d_{ij}$.
- $c_i$ is the cost of building a station at node $i$, $c_e$ is the cost of building link $e$. A bound $C_{\text{max}}$ on the available budget is also given.
- The mobility pattern is given by a matrix $G = (g_{pq}) : pq \in W$; where $W$ is the ordered index pair set: $W = \{pq : p, q \in N\}$, also referred to as the set of demands.
- The generalized cost of satisfying each demand $pq$ by the complementary mode is $u_{pq}^{COM}$. 
Data: OD and Complementary generalized cost matrices

\[
G = \begin{pmatrix}
0 & 9 & 26 & 19 & 13 & 12 & 4 & 6 & 4 \\
11 & 0 & 14 & 26 & 7 & 18 & 3 & 7 & 9 \\
30 & 19 & 0 & 30 & 24 & 8 & 3 & 9 & 11 \\
21 & 9 & 11 & 0 & 22 & 16 & 21 & 18 & 16 \\
14 & 14 & 8 & 9 & 0 & 20 & 12 & 18 & 9 \\
26 & 1 & 22 & 24 & 13 & 0 & 11 & 28 & 21 \\
7 & 5 & 6 & 19 & 15 & 13 & 0 & 16 & 14 \\
5 & 9 & 11 & 16 & 17 & 25 & 17 & 0 & 21 \\
6 & 8 & 10 & 18 & 11 & 20 & 14 & 20 & 0 \\
\end{pmatrix};
\]

\[
U^{COM} = \begin{pmatrix}
0 & 1.6 & 0.8 & 2 & 1.6 & 2.5 & 4 & 3.6 & 4.6 \\
2 & 0 & 0.9 & 1.2 & 1.5 & 2.5 & 3.2 & 3.5 & 4.5 \\
1.5 & 1.4 & 0 & 1.3 & 0.9 & 2 & 3.3 & 2.9 & 3.9 \\
1.9 & 2 & 1.9 & 0 & 1.8 & 2 & 2 & 3.8 & 4.1 \\
3 & 1.5 & 2 & 2 & 0 & 1.5 & 3 & 2 & 3 \\
2.1 & 2.7 & 2.2 & 1 & 1.5 & 0 & 2.5 & 3 & 2.5 \\
3.9 & 3.9 & 3.9 & 2 & 3 & 2.5 & 0 & 2.5 & 2.5 \\
5 & 3.5 & 4 & 4 & 2 & 3 & 2.5 & 0 & 2.5 \\
4.6 & 4.5 & 4 & 3.5 & 3 & 2.5 & 2.5 & 2.5 & 0 \\
\end{pmatrix}.
\]
Data: Example
Objective functions

• Oriented to the society: covered population, trip coverage, pollution and congestion reduction, sustainability, etc.

• Operator oriented: costs of construction, rolling stock purchase, human and operation, benefit, etc.

• User oriented: accessibility, riding times, directness trips, etc.
Objective functions

\[ z_{\text{cover}} = \sum_{w=(p,q) \in W} g_w p_w \]

\[ z_{\text{cost}} = \sum_{n_i,n_j \in A', i < j} c_{ij} x_{ij} + \sum_{l \in L} \sum_{n_i \in N} c_i y_i^l \]

\[ z_{\text{PUB Time}} = \sum_{(i,j) \in A} \sum_{w \in W} T_{ij}^w g_w \]

\[ z_{\text{Total Time}} = \sum_{w \in W} g_w (p_w u_w^{PUB} + (1 - p_w) u_w^{PRIV}) \]

\[ \cdots \cdots \cdots \cdots \cdots \]
Objective: our problem

- Our goal is to choose a set of edges satisfying the budget constraints, so that the resulting network covers the maximum number of travelers, meaning, so that as many travelers as possible find the railway network more attractive (faster?) than the already existing alternative transportation mode.
Objective

• How is allocated the demand to RTN?:

By comparing the time with that of the alternative mode

By comparing generalized costs

By comparing utilities values
Objective

Laurier-Université de Montréal
Bus: 25 min.
Metro: 20 min.  ➔ Metro!
Variables

\[ y_i = \begin{cases} 
1, & \text{if node } i \text{ is chosen to be a station} \\
0, & \text{otherwise} 
\end{cases} \]

\[ x_{ij} = \begin{cases} 
1, & \text{if } ij \text{ is chosen to be in RTN} \\
0, & \text{otherwise} 
\end{cases} \]

\[ f_{ij}^{kl} = \begin{cases} 
1, & \text{if } kl \text{ uses arc } ij \in A(E) \\
0, & \text{otherwise} 
\end{cases} \]

\[ r_{kl} = \begin{cases} 
1, & \text{if OD pair } kl \text{ is assigned to RTN} \\
0, & \text{otherwise} 
\end{cases} \]
Objective and constraints

• Maximize $\sum_{kl \in W} g_{kl} r_{kl}$

• subject to

  budget constraints
  alignment location constraints
  routing demand conservation constraints
  location-allocation constraints
  splitting demand constraints
  binary conditions.
Constraints

• Budget constraints

\[ \sum_{i,j \in A, i < j} c_{ij}x_{ij} + \sum_{i \in N} c_iy_i \leq C_{\text{max}} \]
Constraints

• Alignment location constraints

\[ x_{ij} \leq y_i, ij \in A \]
\[ x_{ij} = x_{ji}, ij, ji \in A \]
Constraints

- Routing demand conservation constraints

\[
\sum_{i,j \in A, j = k} f^{kl}_{ij} = 0, \ kl \in W \\
\sum_{i,j \in A, i = k} f^{kl}_{ij} = r_{kl}, \ kl \in W \\
\sum_{i,j \in A, j = l} f^{kl}_{ij} = r_{kl}, \ kl \in W \\
\sum_{i,j \in A, i = l} f^{kl}_{ij} = 0, \ kl \in W \\
\sum_{i,j \in A} f^{kl}_{ij} - \sum_{j,i \in A} f^{kl}_{ji} = 0, \ \forall j \notin \{k, l\}, \ kl \in W
\]
Constraints

• Splitting demand constraints

\[ \epsilon + u_{kl} - u_{kl}^{COM} - M(1 - r_{kl}) \leq 0, \quad kl \in W \]

where \( u_{kl} = \sum_{ij \in A} d_{ij} f_{ij}^{kl} + u_{kl}^{COM}(1 - r_{kl}) \)
Constraints

• Location-allocation constraints

\[ f_{ij}^{kl} + r_{kl} - 1 \leq x_{ij}, \, kl \in W \]
Constraints

• Binary constraints

\[ x_{ij}, y_i f_{ij}^k, r_{kl} \in \{0, 1\} \]
Computational Complexity

• The problem posed before is (strongly) NP-hard.
• Solving toy instances made of 15 stations and 30 edges takes around 1 week.
• Real instances (more than 100 possible stations and 200 possible edges) would be impossible to solve with the current technology.
• Heuristic algorithms are needed. In this presentation: GRASP algorithms.
Greedy Randomized Adaptive Search Procedure

• Is a metaheuristic algorithm commonly applied to large combinatorial optimization problems.

• Consists of iterations made up from successive constructions of a greedy randomized solution, which are generated by adding elements to the problem’s solution set, randomly from a list of elements ranked by a greedy function according to the quality of the solution they will achieve

• GRASP was first introduced in Feo and Resende (1989).
GRASP: notation

Given a set of edges E, let \( V(E) \) denote the set of vertices included in the edges of E.

Let \( k \in \mathbb{N} \) be the number of possible edges to be included at each iteration.

Let \( R_{N_t} \) be the set of edges that form the solution, not necessarily optimal, obtained after the \( t \)-th iteration.

In a given iteration \( t \), edge \( e_t \in E \) is said to be feasible if \( e_t \not\in R_{N_{t-1}} \), graph \((V(R_{N_{t-1}} \cup \{e_t\}), R_{N_{t-1}} \cup \{e_t\})\) is connected and the construction cost of such a graph is not larger than the maximum cost \( C_{\text{max}} \). \( F_{E_t} \) denotes the set of feasible edges at iteration \( t \).
GRASP: construction phase

• Step 0: Initialize RN0 = empty set, t = 0.
• Step 1: Set t = t + 1 and determine the set FE_t. If \( |FE_t| = 0 \), go to Step 4.
• Step 2: If \( |FE_t| > k \), determine a set \( SE_t^k = \{e_{i1},...,e_{ik}\} \) consisting of \( k \) edges of \( FE_t \) that individually generate the \( k \) largest improvements of the objective function when they are added to RN_{t-1}. Otherwise, that is if \( |FE_t| \leq k \), set \( SE_t^k = FE_t \).
• Step 3: Randomly choose one edge \( e_t \in SE_t^k \). Set \( RN_t = RN_{t-1} U\{e_t\} \) and go to Step 1.
• Step 4: RN_{t-1} is the best solution found. In order to obtain a variety of solutions, we repeat this \( l_{max} \) times, choosing the best solution as a final network in this phase.
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Improvement phase

Let $\text{RN} = \{e_1, \ldots, e_n\}$ be the best solution found in the construction phase, and let $\text{RN} = E \setminus \text{RN}$.

For $i$ from 1 to $n$ do:

Step 1: Let $\text{RN}^i = \text{RN} \setminus \{e_i\}$.

Step 2: Let $e_j \in \text{RN}$ such that it gives the largest improvement in the objective function when added to $\text{RN}^i$ among all edges in $\text{RN}$ that satisfy the cost constraints. Keep doing this while there is budget available.

Step 3: $\text{RN}(i) = \text{RN}^i \setminus \{e_j\}$.

Let $i^*$ be such that $\text{RN}(i^*)$ is not worse than $\text{RN}(j)$; for all $j \neq i^*$. If $\text{RN}(i^*)$ improves $\text{RN}$, then we choose as a final solution $\text{RN}(i^*)$. Otherwise we remain with $\text{RN}$.
Example
Example
Example
A real example
A real example
A real example
A real example
The robust network design problem

Now the link Parc-Acadie is out of service.
Laurier-Université de Montréal:
Bus: 25 min.
Metro: 30 min
The robust network design problem

• Now we take into account possible edge failures. Let $\gamma_e$ be the probability that edge $e$ fails, and let $K(r; e)$ the trip coverage of network $r$ if edge $e$ fails.

  The new goal could be:

  1. maximize (over all possible networks) the minimum trip coverage when one of the edges fails,

  \[
  \max_r \min_e K(r; e); \quad \text{(SRND)}
  \]

  2. maximize the expected trip coverage,

  \[
  \max_r ((1 - \sum_e \gamma_e)K(r) + \sum_e \gamma_e K(r; e)); \quad \text{(PRND)}
  \]
The robust network design problem: references

- A game theoretic framework for the robust railway transit network design problem, Gilbert Laporte, Juan A. Mesa, Federico Perea, Transportation Research Part B, 44, 2010
Robust network design with alternative routes

- Maximize $\sum_{kl \in W} g_{kl} r_{kl}$
- subject to
  - budget constraints
  - alignment location constraints
  - routing demand conservation constraints
  - location-allocation constraints
  - splitting demand constraints
  - demand-arc flow constraints
  - binary conditions
Robust network design with alternative routes

\[ f_{ij}^{kl} \leq \frac{1}{r_{ij}^{kl}}, \quad ij \in A_{DAF} \subset A, \quad W_{DAF} \subset W \]

where \( A_{DAF} \) and \( W_{DAF} \) are the sets of selected arcs and O/D pairs and \( r_{ij}^{kl} \) is the number of different routes for demand \( kl \)
Robust network design with alternative routes: reference

• Designing robust rapid transit network with alternative routes,

Gilbert Laporte, Ángel Marín, Juan A. Mesa, Federico Perea,

Journal of Advanced Transportation, 45, 2011
Computational Complexity

Let $G = (N, E)$ be the underlying and alternative network.

Let $c_i$, $c_{ij}$ and $C_{max}$ costs.

Let $W \subset N \times N$ be the set origin-destination pairs $g_{pq}$ the number of travelers of the pair $pq$ and $u_{pq}^{ALT}$ the utility of pair $pq$ with the alternative mode.
Computational Complexity

For the subnetwork $S \subset G$, let $u_{pq}^S$ the utility of the pair $pq$ for network $S$.

Given $S \subset G$, the proportion of travelers between $p$ and $q$ that will use the new system is given by: $\phi : R \rightarrow [0, 1]$ Examples:

Logit function:

$$\phi(u_{pq}^S) = \frac{1}{1 + \gamma_1 e^{-\gamma_2 (u_{pq}^S - u_{pq}^{ALT})}}$$

Binary function:

$$\phi(u_{pq}^S) = \begin{cases} 1, & \text{if } u_{pq}^S \geq u_{pq}^{ALT} \\ 0, & \text{if } u_{pq}^S < u_{pq}^{ALT} \end{cases}$$
Computational Complexity

Estimated ridership: \[ Z(S) = \sum_{pq \in W} g_{pq} \phi(u_{pq}^S) \]

Total cost: \[ C(S) = \sum_{i \in N(S)} c_i + \sum_{ij \in E(S)} c_{ij} \]
Computational Complexity

Properties:

(1) \( Z(S) \geq 0 \)

(2) \( Z(S) \) is non-decreasing

(3) If \( S \) is a forest such that \( S = \bigcup S_i \) then \( Z(S) = \sum_i Z(S_i) \)

(4) \( Z \) is supermodular
Computational Complexity

(RND1), (RND2) and (RND3) are NP-hard problems

If G is a forest then (RND1) and (RND3) are NP-hard and (RND3) is trivial
Further research

Integrating network design with other phases

- A general rapid network design, line planning, and fleet investment integrated model.
  David Canca, Alicia de-los-Santos, Gilbert Laporte, Juan A. Mesa,
Thank you for your attention!

Juan A. Mesa

Department of Applied Mathematics II, University of Seville