A Carpool Matching Problem

Liutong Zhou

November 15, 2016

1 A Carpool Matching Problem

1.1 A Carpool Matching Problem Modeled in Integer Programming

Given a dataset comprising n trip records with each record including origin, destination and departure time, we are interested in a strategy to pair as many trips as possible.

1.1.1 Notation

Each trip $R_i = \{\vec{O}_i, \vec{D}_i, t_i, p_i, C_i\}$ $(i = 1, ..., n)$ comprises

- Origin: $\vec{O}_i = (latitude_i, longitude_i)$
- Destination: $\vec{D}_i = (latitude_i, longitude_i)$
- Departure Time: t_i
- Number of passengers: p_i
- Allowable number of passengers in this car: C_i

Introduce a decision variable $x_{i,j}$ such that $\forall i, j = 1, ..., n$

$$x_{i,j} = \begin{cases} 1 & \text{if trip i and trip j are paired} \\ 0 & \text{otherwise} \end{cases}$$

then $x_{i,j}$ automatically satisfies $x_{i,j} = x_{j,i}$ meaning $X \in S^n$. Besides, we set $x_{i,i} = 1$ merely for the convenience of calculation.

Consider that each passenger has a tolerance for waiting time, and therefore each paired trip cannot have different origins located too far from each other

- Tolerance for waiting time: $Tolt_i$
- Tolerance for origins' distance: $Tolo_i$
- Tolerance for destinations' distance: $Told_i$

For each trip i, maintain a set of indices to $\{R\}$ called Candidate such that

$$\begin{array}{ll} \forall i = 1, ..., n & Candidate_i = \{indices\} \\ \text{where } \forall k \in Candidate_i & distance(\vec{O}_i, \vec{O}_k) \le Tolo_i \\ & distance(\vec{D}_i, \vec{D}_k) \le Told_i \\ \text{and} & |t_i - t_k| \le Tolt_i \end{array}$$

1.1.2 Model Setup

objective: Maximize total number of paired trips

$$max \quad Z = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} x_{i,j} = nnz(triu(X, 1))^1 \qquad (1)$$

subjective to:

$$\begin{array}{lll} \forall i, j = 1, ..., n & x_{i,j} = x_{j,i} & (2) \\ & x_{i,i} = 1 & (3) \\ & x_{i,j} \in \{0, 1\} & (4) \\ \forall i = 1, ..., n & \sum_{j=1}^{n} x_{i,j}\, p_j = X_i^T \vec{p} \le C_i & (5) \\ & x_{i,j} = 0 \text{ if } j \notin Candidate_i & (6) \end{array}$$

Constraint (5) sets an upper bound for the total number of passengers of each paired trip. Constraint (6) eliminates those candidates that go beyond passengers' tolerance.

1.2 A Carpool Matching Problem Solved by Brute Force

2 A Carpool Routing Problem

[1] The last equality is just a Matlab syntax for calculating the summation