Macroscopic Fundamental Diagram Based Discrete Transportation Network Design

Guojing Hu[1], Feng Wang[2] and Robert W. Whalin[3*]

1. Department of Mathematics and Statistical Sciences, Jackson State University;
2. Ingram School of Engineering, Texas State University;
3. Department of Civil and Environmental Engineering, Jackson State University

Introduction

- Different from the existing traditional models for network design problems, the proposed bi-level programming model aims to improve network performance by obtaining optimal MFD-based network capacity.
- To solve the proposed model, we design an algorithm framework, in which VISSIM software with COM interface is used to achieve and simulate traffic flow states under user equilibrium and further to obtain MFDs, while k-means clustering algorithm is employed to quantify the MFD-based network capacity.
- In case study, the results show that MFDs are able to reflect the influence of different network building strategies on the traffic network capacity.

Mathematical Formulation

Upper-level Network Design

$$Q = \max_{k\in K} Q_k \qquad (1)$$

$$\sum_{a\in \bar{A}} b_a y_a \le B, \qquad \forall a \in \bar{A} \qquad (2)$$

$$A = A^0 \cup \{a \mid y_a > 0, a \in \bar{A}\} \qquad (3)$$

$$Q_k = \frac{\sum_{a\in A} x_a^k \cdot l_a}{\sum_{a\in A} l_a} \qquad \forall k \in K \qquad (4)$$

$$y_a = \{0,1\}, \qquad \forall a \in \bar{A} \qquad (5)$$

In objective function (1), the optimal network capacity is expressed as the maximum weighted flow in all feasible building plans under varied traffic demand scenarios. Constraint (2) ensures the total cost of building new roads not to overrun the given budget. Constraint (3) is used to calculate the extended network A based on original network A^0 and potential links to build. Constraint (4) is a calculation function for weighted flow Q_k. Constraint (5) decides whether or not to construct the new road a.

Lower-level Flow Assignment

$$Z^k = \min \sum_{a\in A} \int_0^{x_a^k} t_a^k(x)dx \qquad \forall k \in K \qquad (6)$$

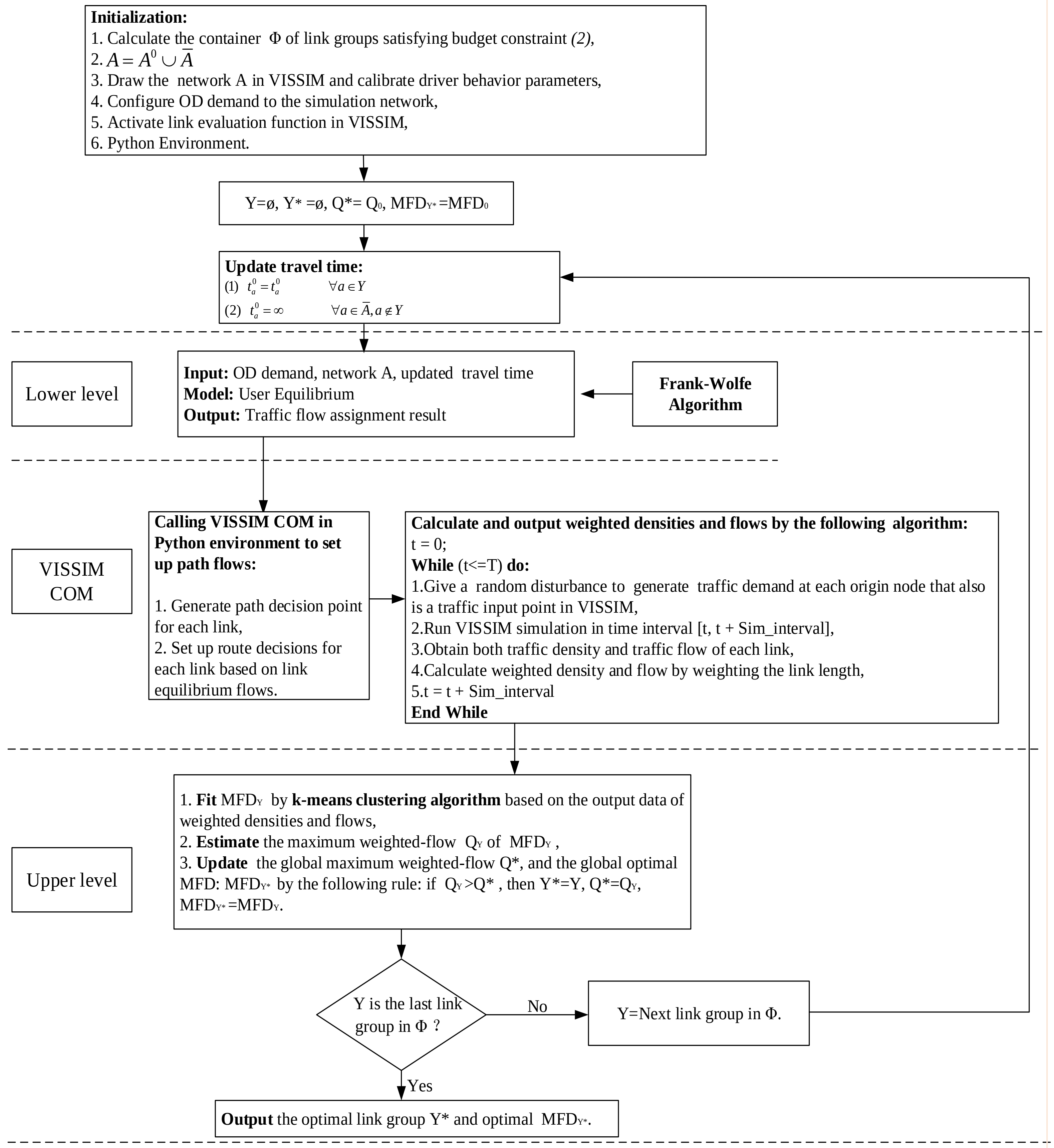
$$\sum_{p\in P_{rs}} f_{p,k}^{rs} = q_k^{rs}, \qquad \forall rs \in D, k \in K \qquad (7)$$

$$x_a^k = \sum_{rs\in D} \sum_{p\in P_{rs}} f_{p,k}^{rs} \delta_{a,p}^{rs,k}, \qquad \forall a \in A, k \in K \qquad (8)$$

$$f_{p,k}^{rs} \ge 0, \qquad \forall p \in P_{rs}, rs \in D, k \in K \qquad (9)$$

Travelers in the lower-level are assumed to follow the user equilibrium (UE) principle whose purpose is to minimize their own travel cost described in formula (6). Constraint (7) depicts the OD demand conservation law under each traffic demand scene, constraint (8) demonstrates the relationship between link flow and route flow under each traffic demand scene, and constraint (9) guarantees the nonnegativity of the route flow.

Algorithm



Initialization:
1. Calculate the container Φ of link groups satisfying budget constraint *(2)*,
2. $A = A^0 \cup \bar{A}$
3. Draw the network A in VISSIM and calibrate driver behavior parameters,
4. Configure OD demand to the simulation network,
5. Activate link evaluation function in VISSIM,
6. Python Environment.

Y=ø, Y* =ø, Q*= Q_0, MFD_{Y*} =MFD_0

Update travel time:
(1) $t_a^0 = t_a^0 \qquad \forall a \in Y$
(2) $t_a^0 = \infty \qquad \forall a \in \bar{A}, a \notin Y$

Lower level

Input: OD demand, network A, updated travel time
Model: User Equilibrium
Output: Traffic flow assignment result

Frank-Wolfe Algorithm

VISSIM COM

Calling VISSIM COM in Python environment to set up path flows:
1. Generate path decision point for each link,
2. Set up route decisions for each link based on link equilibrium flows.

Calculate and output weighted densities and flows by the following algorithm:

```
t = 0;
While (t<=T) do:
    1.Give a random disturbance to generate traffic demand at each origin node that also is a traffic input point in VISSIM,
    2.Run VISSIM simulation in time interval [t, t + Sim_interval],
    3.Obtain both traffic density and traffic flow of each link,
    4.Calculate weighted density and flow by weighting the link length,
    5.t = t + Sim_interval
End While
```

Upper level

1. **Fit** MFD_Y by **k-means clustering algorithm** based on the output data of weighted densities and flows,
2. **Estimate** the maximum weighted-flow Q_Y of MFD_Y ,
3. **Update** the global maximum weighted-flow Q*, and the global optimal MFD: MFD_{Y*} by the following rule: if Q_Y >Q* , then Y*=Y, Q*=Q_Y, MFD_{Y*} =MFD_Y.

Y is the last link group in Φ ?

No

Y=Next link group in Φ.

Yes

Output the optimal link group Y* and optimal MFD_{Y*}.

Fig 1. Algorithmic framework for solution strategies

Case Study

MFD Analysis

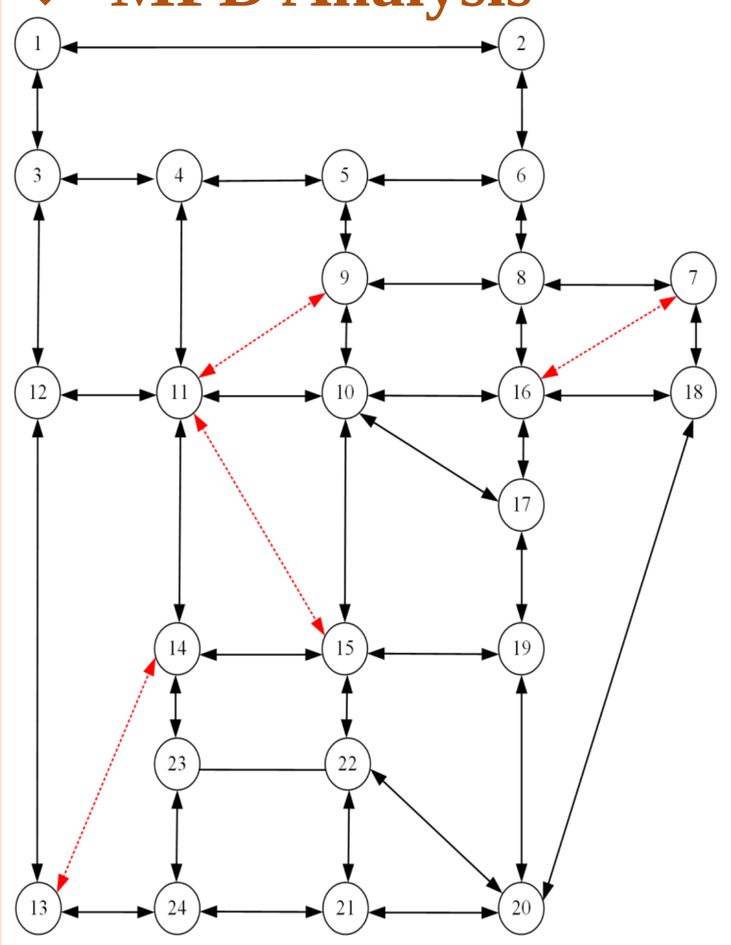


Fig 2. Sioux Falls test network

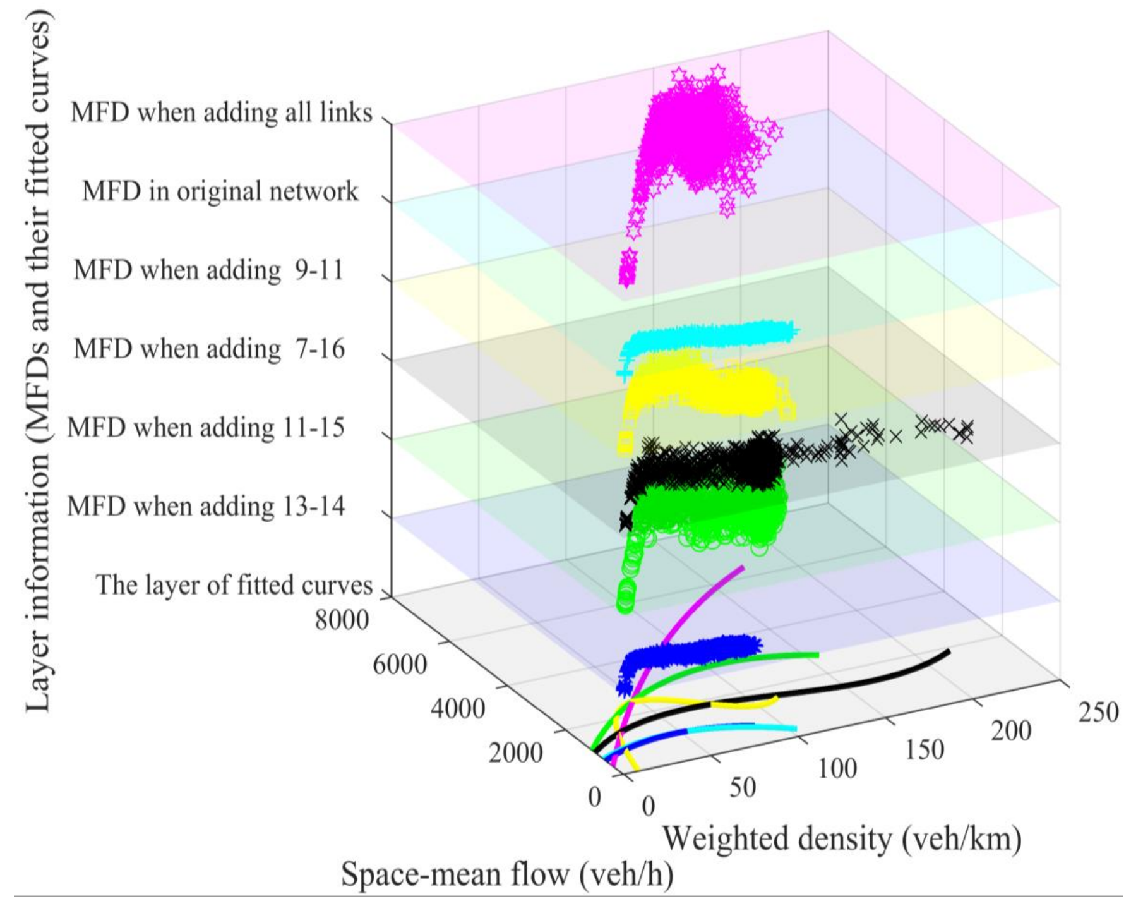


Fig 3. MFDs of different extended Sioux Falls networks

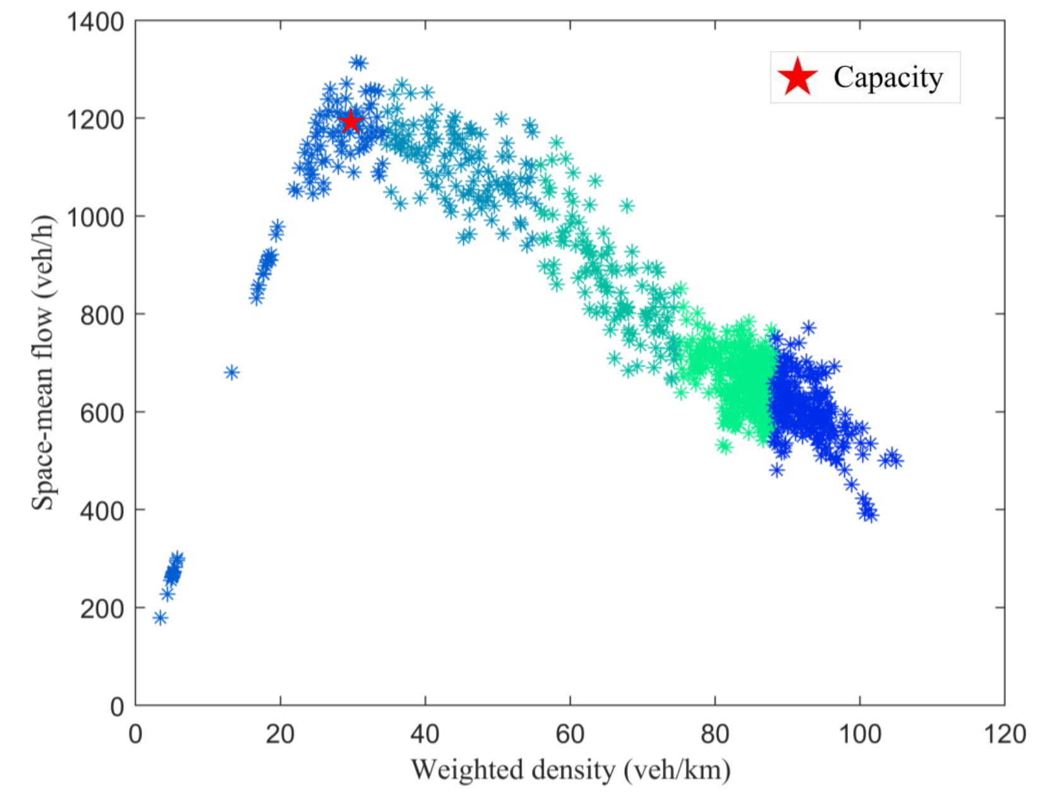


Fig 4. K-means clustering for an MFD

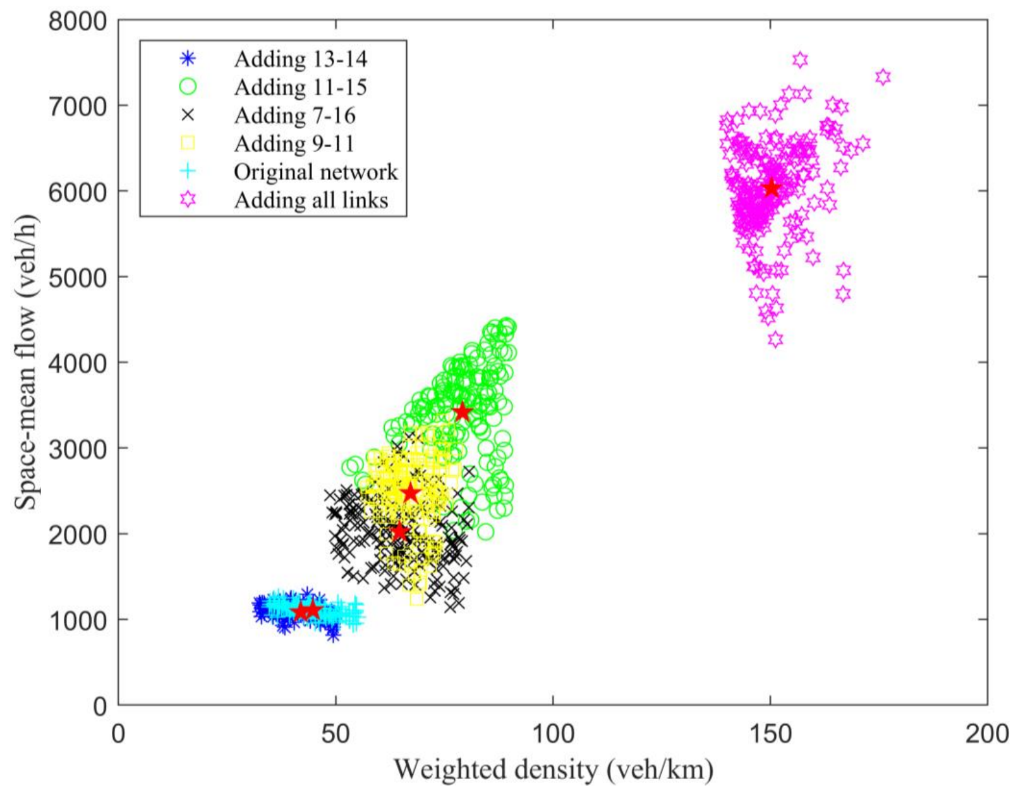


Fig 5. The highest cluster / capacity from the MFD of each building strategy

Capacity Paradox: Building a new link reduces the network capacity on the contrary.

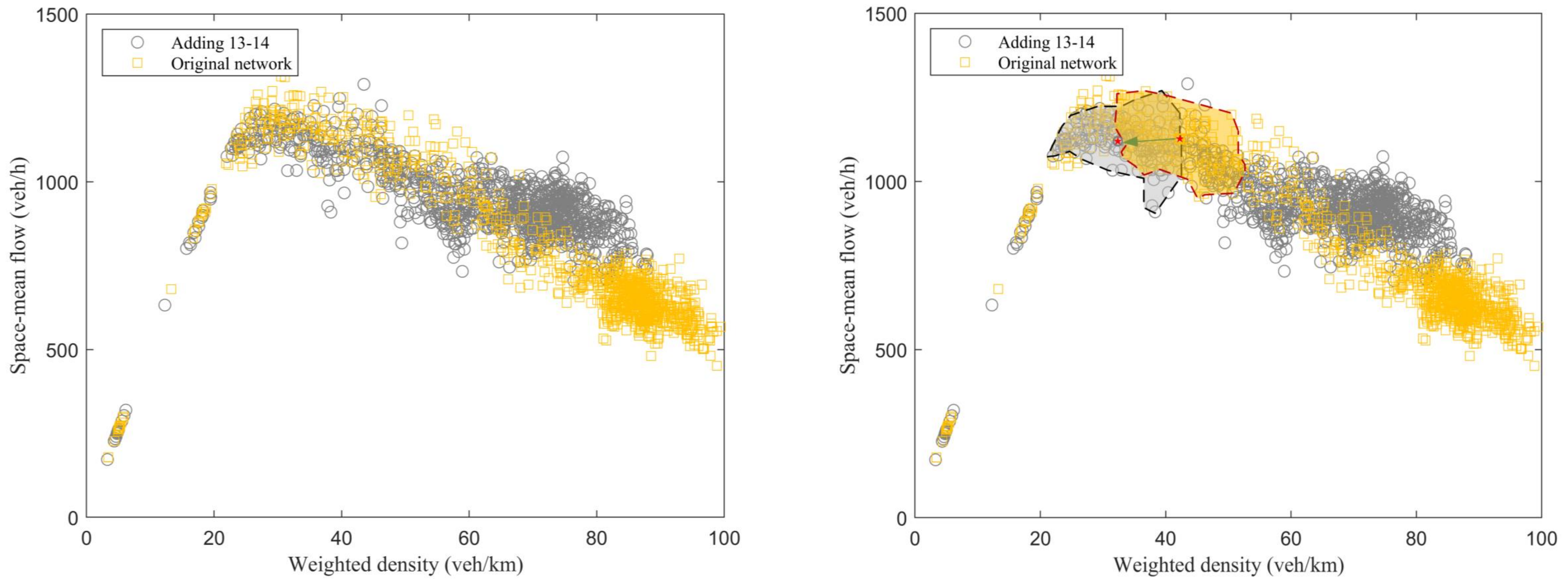


Fig 6. Adding 13-14 reduces the network capacity and optimal accumulation on the contrary

Relationship with Network Betweenness

Fig 7. The relationship between network betweenness and network capacity for each building strategy.

Conclusions

First, the feasibility of the model and algorithm to solve network design problems is verified. Second, the proposed approach has the benefit of avoiding the capacity paradox. Third, the relationship between network capacity and network betweenness is discussed.

Acknowledgement

The study is supported by grants from Southeastern Transportation Research, Innovation, Development and Education Center (STRIDE) and China Scholarship Council.