USC Viterbi
School of Engineering

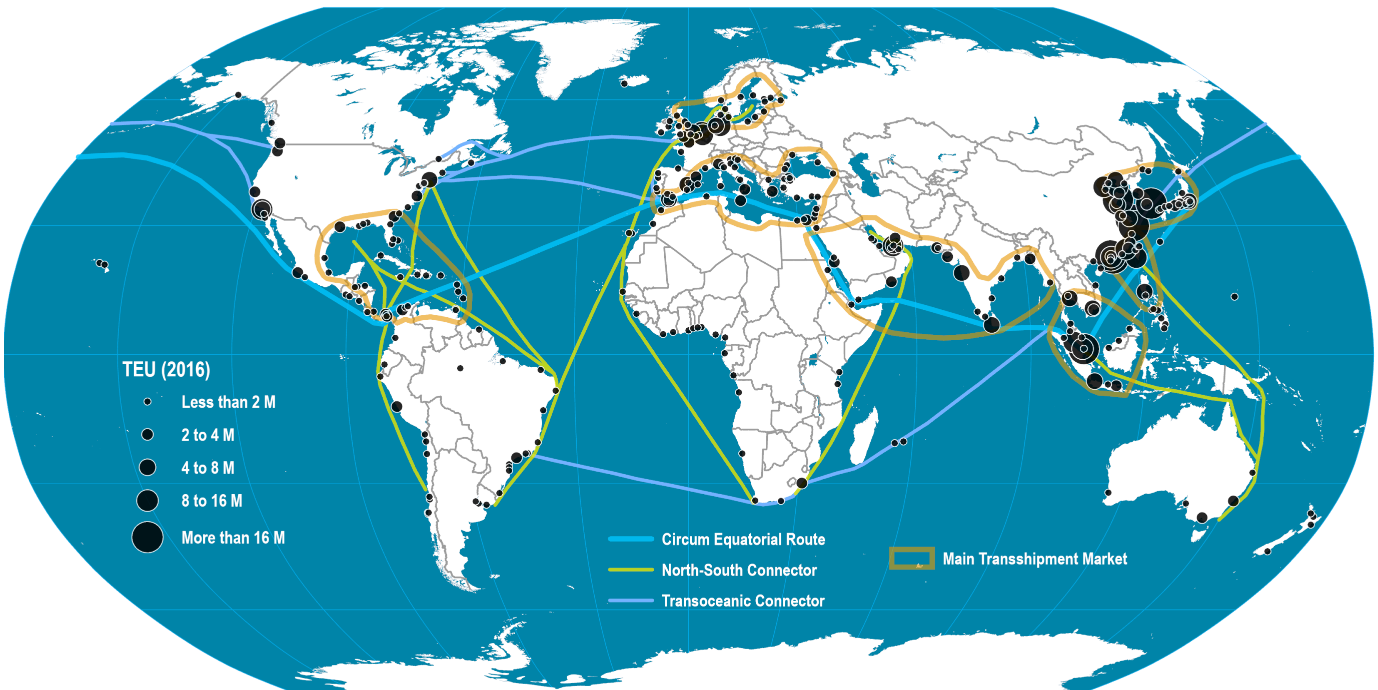
DISTRIBUTED LOAD BALANCING FREIGHT ROUTING

WITH A CO-SIMULATION OPTIMIZATION METHOD

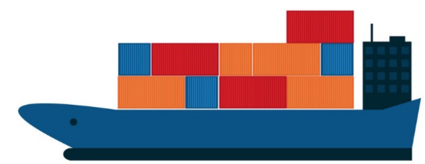
Pengfei Chen
Petros Ioannou, Maged Dessouky
University of Southern California

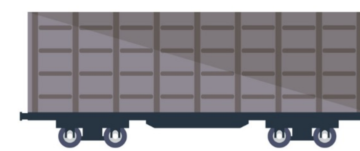
05/25/2022

Global Transhipment

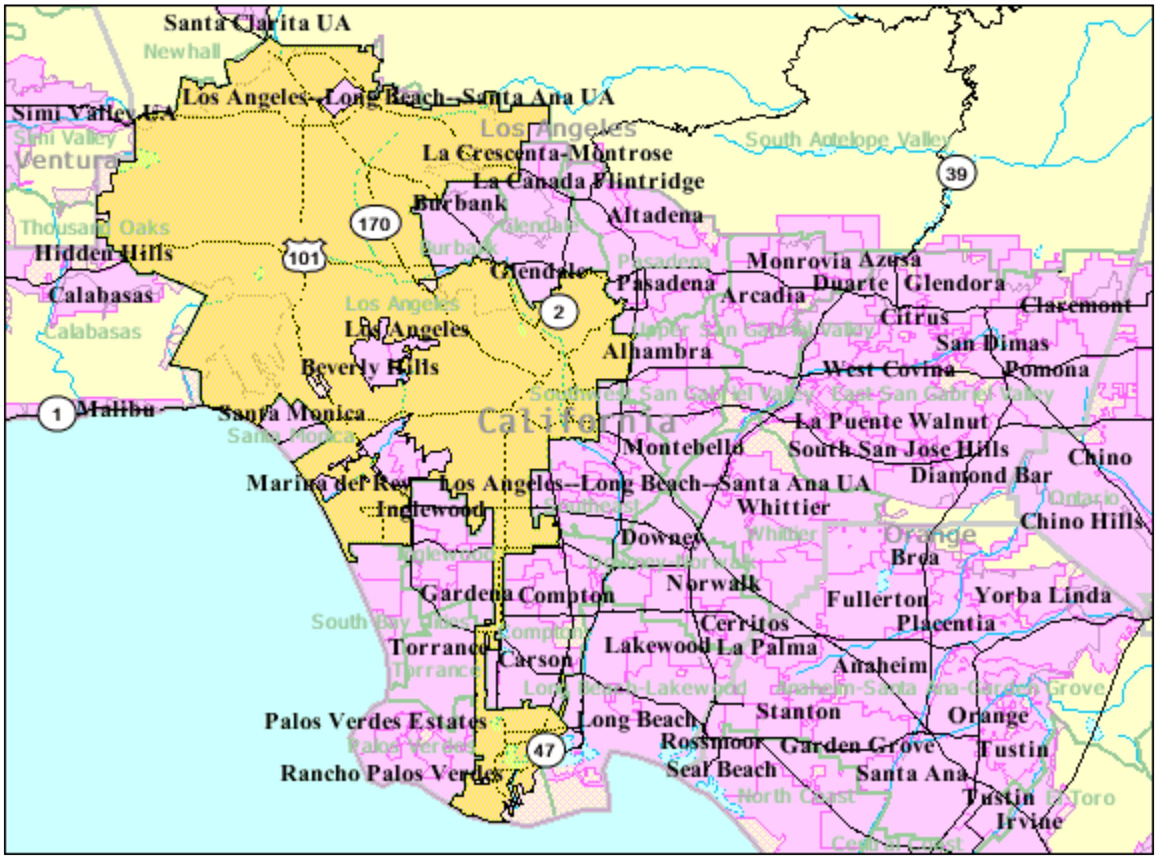


Jean-Paul Rodrigue (2017), Maritime Transportation, New York: Routledge

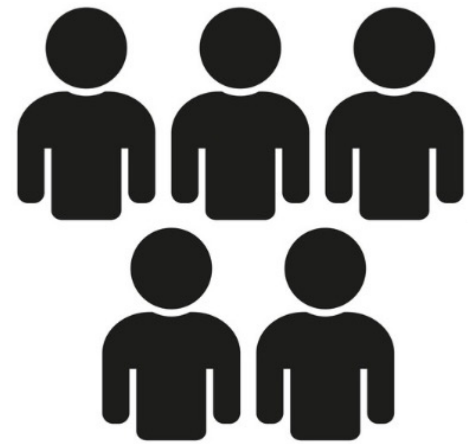




Los Angeles Metropolitan Area



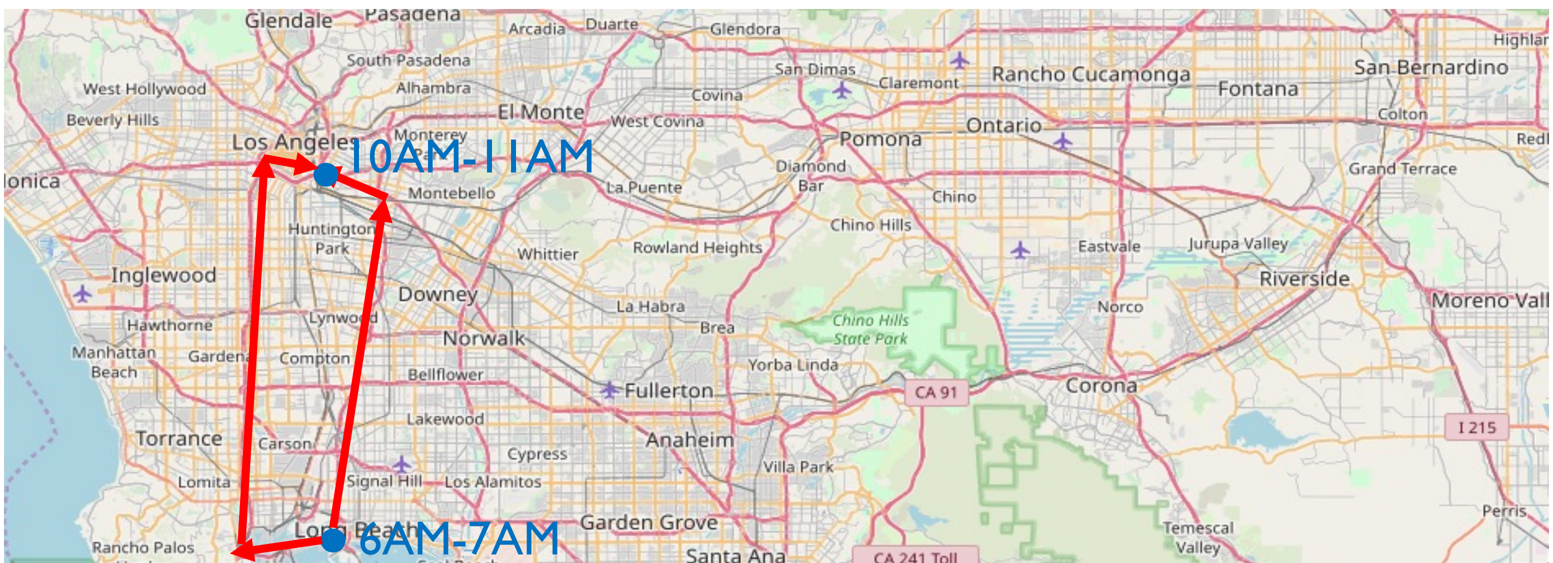
US Federal Gov, Map of Los Angeles, California



Research Problem

To assign a number of demands from origins to destinations with considerations of several factors: *time window*, *road network traffic condition*, *emissions*, etc.

- *Time window*: pick up time window, delivery time window
- *Road network traffic condition*: Vehicle Travel Time
- *Emissions*: $CO_2, PM_{2.5}, NO_x$



openstreetmap.org

Formulation

$$\min TC(X) = \sum_{k\in K}\sum_{i\in I}\sum_{j\in J}\sum_{r\in R_{i,j}} S_{i,j}^{r}(k)X_{i,j}^{r}(k)$$

$$= \sum_{k\in K}\sum_{i\in I}\sum_{j\in J}\sum_{r\in R_{i,j}} (C_{i,j}^{r}(k) + \theta^{r}T_{i,j}^{r}(k))X_{i,j}^{r}(k)$$

subject to:

$\sum_{k\in K}\sum_{r\in R_{i,j}} X_{i,j}^{r}(k) = d_{i,j}, \forall i \in I, \forall j \in J$ (1)

$\sum_{i\in I}\sum_{j\in J}\sum_{r\in R_{i,j}}\sum_{\tau\le k} X_{i,j}^{r}(\tau)\delta_{l,\tau,k}^{r} = x_l(k), \forall l \in L, \forall k \in K$ (2)

$0 \le x_l(k) \le u_l v_l(k), \forall l \in L^R, \forall k \in K$ (3)

$X_{i,j}^{r}(k) \ge 0, \forall i \in I, \forall j \in J, \forall k \in K$ (4)

$d_{i,j}$: The total demand from an origin i to a destination j;

$X_{i,j}^{r}(k)$: The freight demand in units of containers from an origin i to a destination j using a route r with a departure time k;

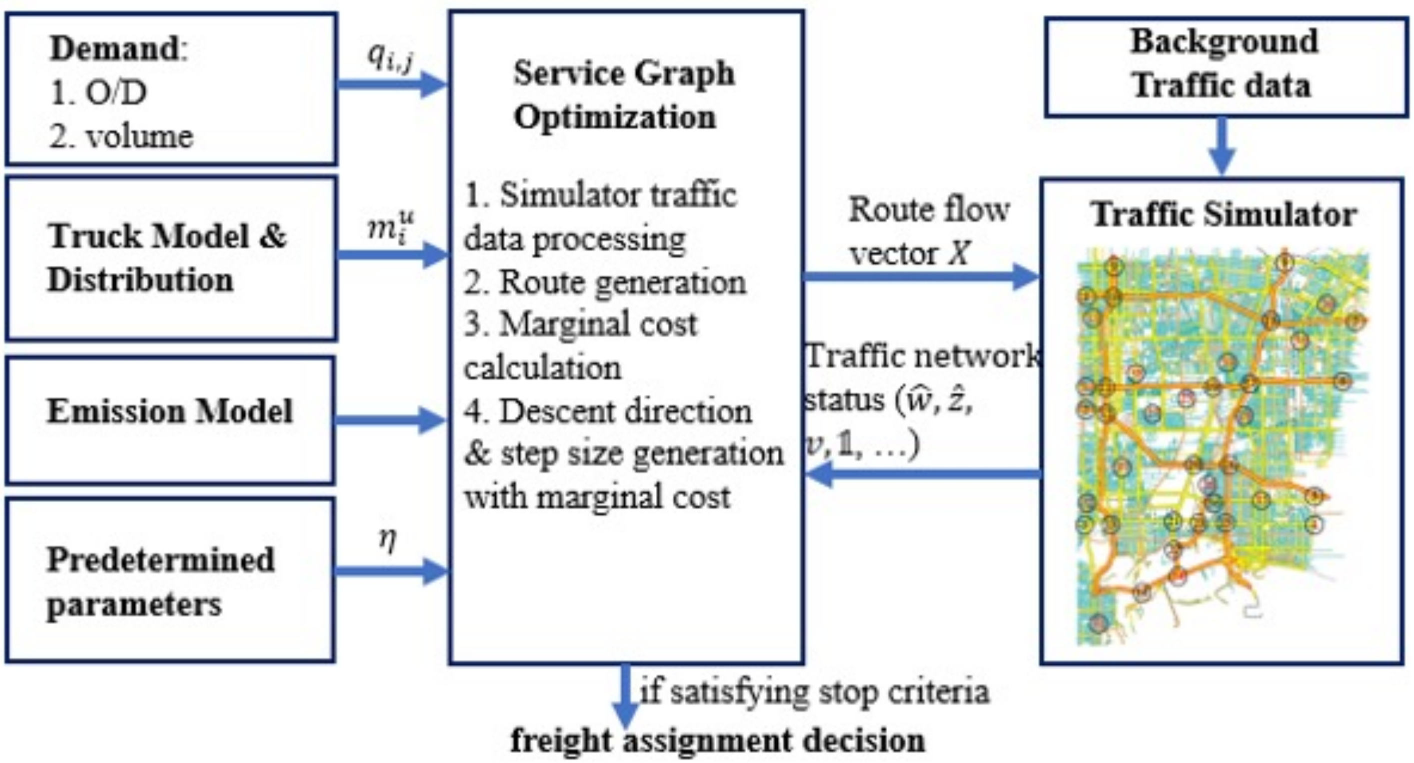
$x_l(k)$: The number of containers using edge l at time k;

u_l : The edge capacity in units of vehicles for edge l;

$v_l(k)$: The vehicle capacity in units of containers per freight vehicle for edge l;

$S_{i,j}^{r}(k)$: combination of the non-travel time vehicle cost $C_{i,j}^{r}(k)$ and the cost of the route travel time $T_{i,j}^{r}(k)$;

Methodology: co-simulation optimization



Methodology:

- Step 1: Initialize iteration number, cost coefficients, route collections.
- Step 2: Check if the stop criteria is satisfied. if not, go to Step 3; otherwise, terminate the algorithm and output the result.
- Step 3: Input the route flow vector into the traffic simulator and obtain the marginal cost of each segment.
- Step 4: Update the marginal cost of each segment as well as routes for each O/D pair and check whether there is a new minimal marginal cost route. If there is, then add it into the route collection.
- Step 5: Construct an augmented route flow vector by assigning the trucks onto the minimum cost route in the current traffic status in an all-or-nothing manner.
- Step 6: Set the route flow vector for the next iteration as a linear combination of the previous route flow vector and the augmented route flow vector. Go to Step 2.

USC Viterbi School of Engineering

Methodology: marginal cost

- Marginal cost : the change in total cost if we add one truck at time instance k on service link l

$$MCP_{l'}^{p'}(k') \approx c_{l'}^{p'} + \sigma_{l'}^{p'} t_{l'}^{p'}(k) + \sum_{n_{p'}=1}^{N_{p'}} \left(\sigma_{l'}^{p'} y_{l'}^{p'}(e_{a_{p',n_{p'}}}(k')) \frac{1}{v_{l'}\left(e_{a_{p',n_{p'}}}(k')\right)\Delta t} \frac{\partial w_{a_{p',n_{p'}}}}{\partial z_{a_{p',n_{p'}}}}(e_{a_{p',n_{p'}}}(k')) \right)$$

$e_{a_{p',n_{p'}}}(k')$: entering time at arc $a_{p',n_{p'}}$ for a freight vehicle using path p' with a departure time of k' from the origin.

$w_{a_{p',n_{p'}}}(k')$: the travel time of arc $a_{p',n_{p'}}$ at time k'

$z_{a_{p',n_{p'}}}(k')$: the traffic volume on road network arc $a_{p',n_{p'}}$ at time k'

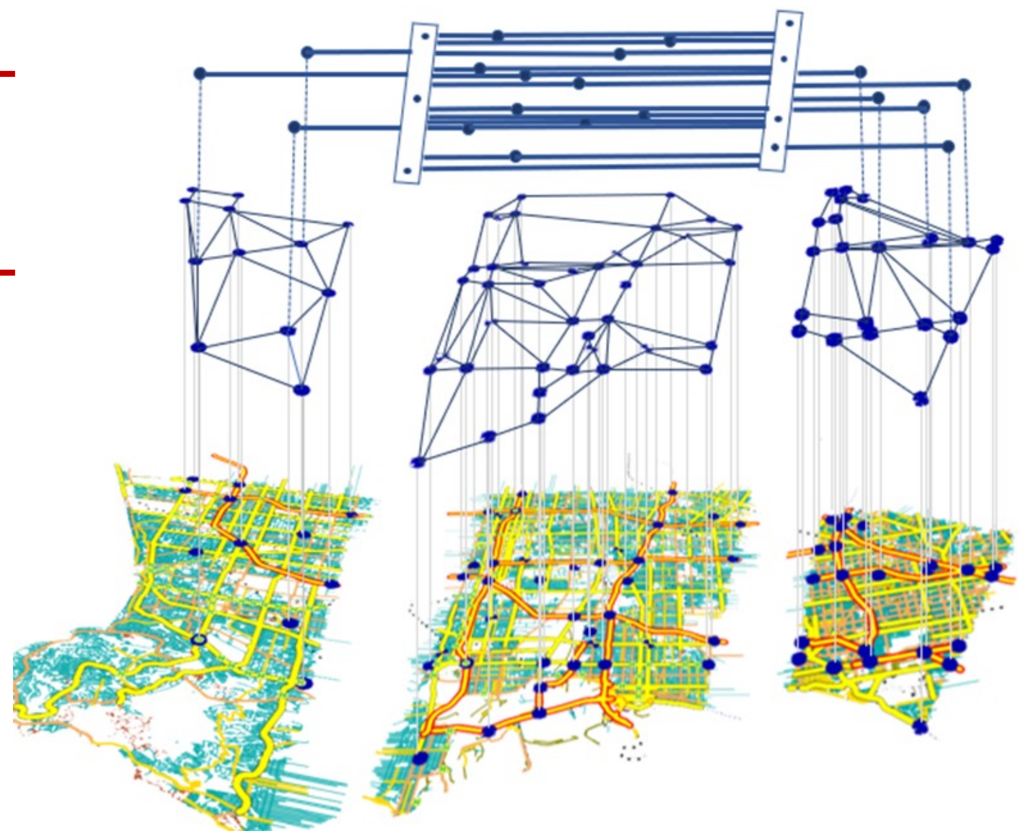
Distributed co-simulation optimization method

- Limitation of centralized co-simulation optimization method: computational complexity when applied to a large-scale road network.

Abstracted service network

service subnetworks

road network in simulator

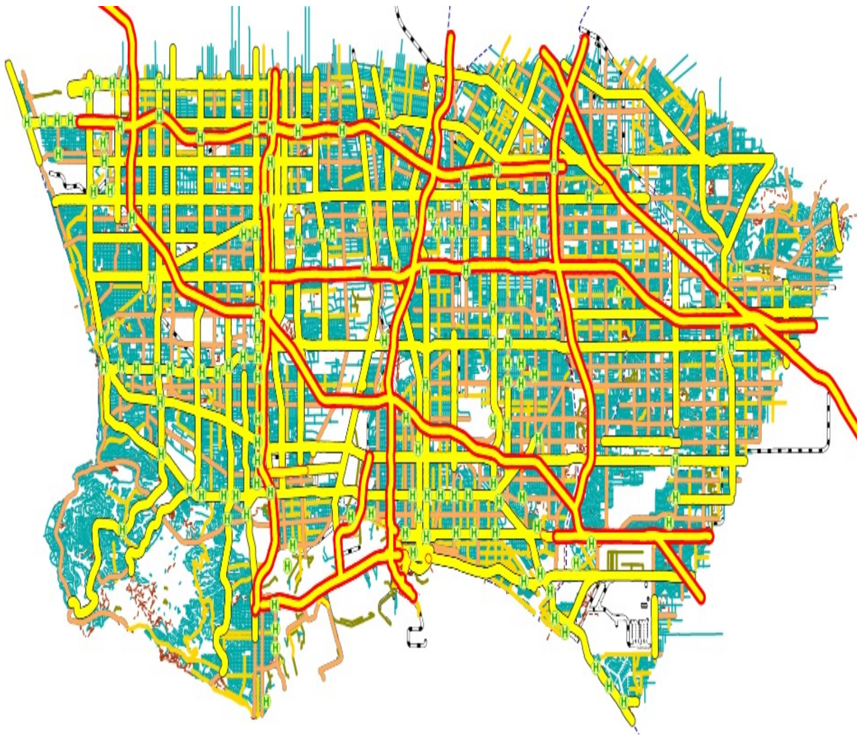


Distributed co-simulation optimization method

- Step 1: Initialize iteration number, cost coefficients, route collections, etc.
- Step 2: Check if the stop criteria is satisfied. if not, go to Step 3; otherwise, terminate the algorithm and output the result.
- Step 3: Input the abstract service network route flow vector into each service subnetwork.
- Step 4: For each service subnetwork and road subnetwork pair, perform a load balancing scheme as in the centralized co-simulation method.
- Step 5: Update the marginal cost for the abstracted service network and check whether there is a new minimal marginal cost route. If there is, add it to the associated route collection and created an augmented route flow vector.
- Step 6: Set the route flow vector for the next outer loop iteration as a linear combination of the previous route flow vector and the augmented route flow vector. Go to Step 2.

Numerical Results

- Long Beach Network

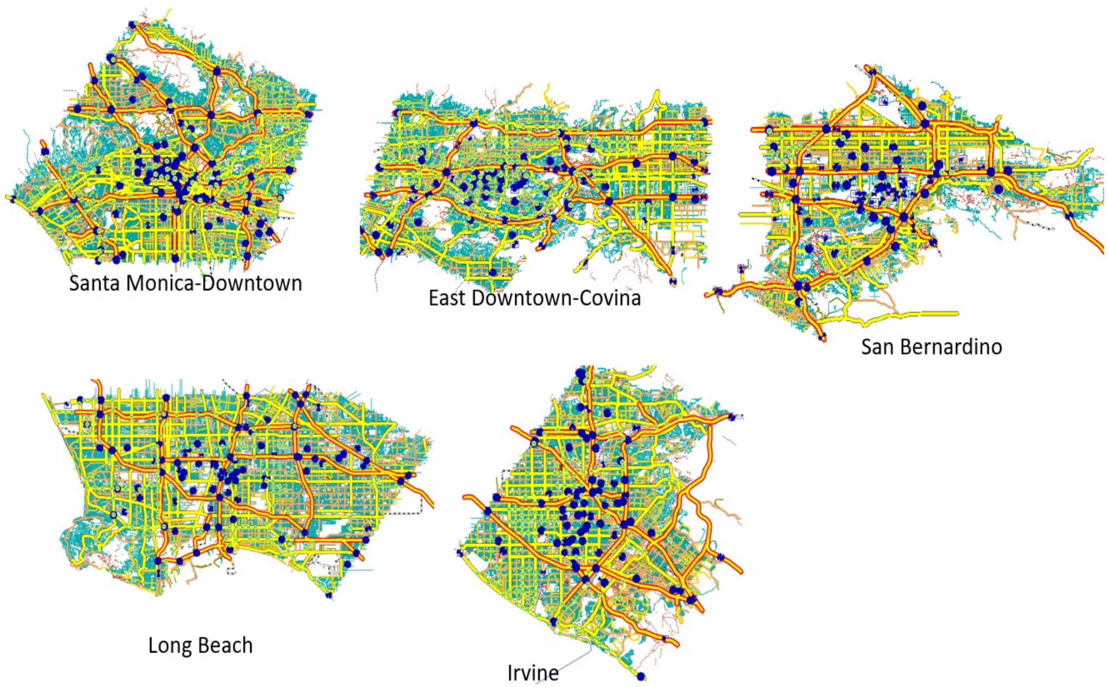


- Total number of demand: 3514
- Configure the scenarios by decomposing the network into 2, 3, 4 subnetworks. Apply the distributed co-simulation optimization method for each scenario and gain the total cost value and computation time of each scenario.
- The centralized load balancing method is also applied on the network as a base for comparison.

USC Viterbi School of Engineering

Numerical Results

- Metropolitan Network



- Total number of demand: 13600
- Configure the scenarios by decomposing the network into 3, 4, 5 subnetworks. Apply the distributed co-simulation optimization method for each scenario and gain the total cost value and computation time of each scenario.
- The centralized load balancing method can not be applied on the network since the capability of the simulator and exponentially growing variables.

Numerical Results

- Long Beach Network

- Performance pattern under different number of subnetworks

# of Subnetworks	0	2	3	4
Total Cost	C_1	$1.053C_1$	$1.091C_1$	$1.147C_1$
Computation Time (second)	4792	3558	3102	3907

$c_1 = \$342189.00$

- Observation:

With the increase of the number of subnetworks, the total cost gained from the distributed load balancing method is moving away from the optimal one gained from centralized load balancing method. However, the computational time is gained through dividing the network.

Numerical Results

- Long Beach Network
- Pattern under different number of demands

Demands	**# of Subnetworks**	0	2	3	4
# of demands = 3514	**Total Cost**	C_1	$1.053C_1$	$1.091C_1$	$1.147C_1$
	Computation Time (second)	4792	3558	3102	3907
# of demands = 7028	**Total Cost**	C_2	$1.059C_2$	$1.096C_2$	$1.151C_2$
	Computation Time (second)	T_2	$0.784T_2$	$0.664T_2$	$0.827T_2$
# of demands = 14056	**Total Cost**	C_3	$1.073C_3$	$1.117C_3$	$1.194C_3$
	Computation Time (second)	T_3	$0.806T_3$	$0.675T_3$	$0.862T_3$

$c_1 = \$342189.00$
$c_2 = \$728162.00$
$c_3 = \$1708326.00$
$T_2 = 5238$ s
$T_3 = 7965$ s

Numerical Results

- Long Beach Network
- Performance pattern under different number of boundary nodes

# of demands = 3514	**# of Boundary Nodes**	5	6	7
	Total Cost	$1.024C$	$1.012C$	C
# of subnetworks = 2	**Computation Time (second)**	T	$1.037T$	$1.083T$

$C = \$355880.00$
T = 3558 s

- Observation:

With the decrease of the number of boundary nodes, we gain benefits on computation time, while lose some optimality.

Numerical Results

- Metropolitan Network
- Performance pattern under different number of subnetworks

Demands	# of Subnetworks	3	4	5
# of demands = 13600	Total Cost	C_4	$1.096C_4$	$1.155C_4$
	Computation Time (second)	$1.648T_4$	$1.217T_4$	T_4
# of demands = 27200	Total Cost	C_5	$1.114C_5$	$1.189C_5$
	Computation Time (second)	$1.918T_5$	$1.39T_5$	T_5

- Observation:

With the increase of the number of subnetworks, the total cost gained from the distributed load balancing method is moving away from the optimal one gained from centralized load balancing method. However, the computational time is gained through dividing the network.

Conclusions

- The distributed co-simulation optimization method is tested under two networks: Long Beach network and metropolitan network and tested can address the scalability issue encountered by centralized co-simulation method.
- For Long Beach network, the distributed co-simulation optimization method is tested based on different number of subnetworks, boundary nodes and demand. By decreasing the number of boundary nodes, we can achieve less computational time with some loss on optimality. By increasing the number of subnetworks, we can achieve saving a large amount of computational time with a relatively small loss on the optimality. However, a proper decomposition is needed since if the network is decomposed too much, the interactions between subnetworks will compromise the computational time gained from decomposition.
- Under various number of demand, the pattern of computational time VS optimality sustains.
- For metropolitan network, similar relation between performance and number of subnetworks is revealed.

Thank you!