Department of Civil, Architectural, & Environmental Engineering
The University of Texas at Austin

Network Partitioning Algorithms for Solving the Traffic Assignment Problem using a Decomposition Approach

Cesar N. Yahia, Venktesh Pandey, & Stephen D. Boyles
CM2

April 26, 2018

Content

Background & Motivation
- Traffic & Congestion
- Traffic Assignment
- A Decomposition Approach for the Static Traffic Assignment Problem

Network Partitioning
- Objectives
- Partitioning Algorithms
- Partitioning Results

Conclusions

Background & Motivation

Traffic & Congestion

- As megaregions grow and become more interconnected, the flow of goods and people across megaregions increases
- This creates traffic management challenges for the different metropolitan planning organizations involved within the megaregion

Figure: commuters and traffic across megaregions (Routley, 2017)

- The traffic assignment problem (TAP) is used to predict driver route choice and link flows (congestion) for a given level of travel demand
- The static version of the problem is commonly employed by planning agencies, and it can be formulated as a convex program
- Modern specialized algorithms can solve this problem efficiently

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{h}} \quad & \sum_{(i,j)\in A} \int_0^{x_{ij}} t_{ij}(x)dx \\
\text{s.t.} \quad & x_{ij} = \sum_{\pi\in\Pi} h^\pi \delta_{ij}^\pi & \forall (i,j)\in A \\
& \sum_{\pi\in\Pi^{rs}} h^\pi = d^{rs} & \forall (r,s)\in Z^2 \\
& h^\pi \geq 0 & \forall \pi \in \Pi
\end{aligned}
$$

Figure: The traffic assignment problem as a convex optimization problem (Beckmann, 56)

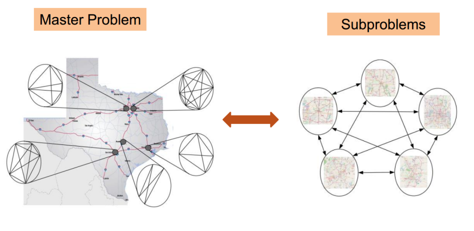
Background & Motivation
Traffic Assignment

- Benefits of solving the traffic assignment problem for an entire megaregion include:
  1. determining congestion levels on transport links connecting megaregion cities
  2. identifying dependencies between congestion states in different megaregion cities
  3. determining the impact of policies on travelers from different origins across the megaregion
- However, such large instances of the traffic assignment problem are computationally challenging to solve using traditional specialized algorithms
- Therefore, there is a need to parallelize the traffic assignment problem across subnetworks within the megaregion

Background & Motivation

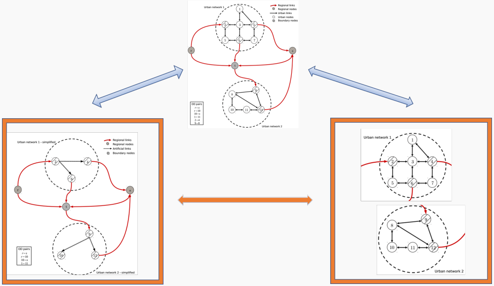
A Decomposition Approach for the Static Traffic Assignment Problem (DSTAP)

- DSTAP is a spatial parallelization scheme for the static traffic assignment problem
- The network is divided into subnetworks that correspond to traffic assignment subproblems solved in parallel
- The algorithm alternates between equilibrating the subproblems and a master problem that represents a simplified version of the full network

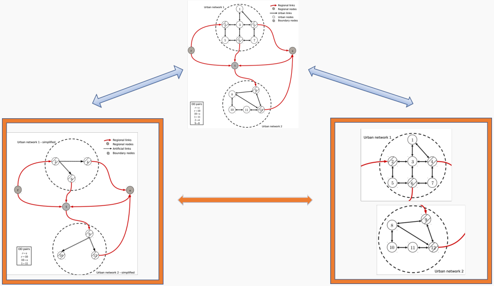


Background & Motivation

A Decomposition Approach for the Static Traffic Assignment Problem (DSTAP)



1. The full network is divided into a simplified master problem (left) and several detailed subproblems (right)
2. The algorithm alternates between equilibrating flows in the master network and in parallel across subproblems
3. The convergence of the algorithm to the global equilibrium across the full network is proved in (Boyles, 2017)
4. This global equilibrium accounts for effects between subnetworks and determines flows on links connecting subnetworks



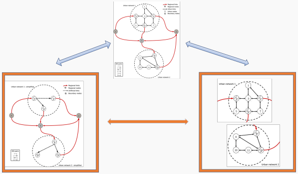
- the algorithm relies on results from equilibrium sensitivity analysis to approximate changes in traffic within a subnetwork using aggregate links (Boyles, 2012)

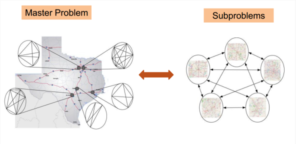
$$
\begin{aligned}
\text{minimize} \quad & \sum_{a \in \tilde{\mathbf{A}}_w} \int_0^{\varrho_a^w} t'_a(\omega)\, d\omega \\
\text{subject to} \quad & \sum_{\pi \in p_w} \beta_\pi^w = 1, \\
& \varrho_a^w = \sum_{\pi \in p_w} \beta_\pi^w \delta_{a\pi}, \quad \forall a \in \tilde{\mathbf{A}}_w
\end{aligned}
$$

$$
t_\theta^{k+1}(x_{\theta,r}^{k+1}) = \mu_\theta^k + \psi_\theta^k(x_{\theta,r}^{k+1} - x_{\theta,r}^k), \qquad \forall \theta \in \Theta_u, u \in U
$$

Background & Motivation

A Decomposition Approach for the Static Traffic Assignment Problem (DSTAP)





Computation time at each iteration depends on:

1. The aggregate links generated using equilibrium sensitivity analysis to represent subnetworks
2. The time needed to solve TAP on the subproblems in parallel which is dominated by the largest subproblem

Computation time needed to converge towards global equilibrium depends on the inter-flow between subnetworks

$\epsilon_{OD} \leq 2B(\epsilon_M + \epsilon_S)$

Network Partitioning

Objectives

- The computational performance of DSTAP critically depends on the *boundary of the subnetworks* since this determines:
  1. The number of aggregate links for each subnetwork
  2. The size of each subnetwork equilibrium subproblem
  3. The inter-flow between subnetworks
- Additionally, the region associated with each metropolitan planning agency should correspond to a distinct subnetwork
  1. This is desired since different planning agencies and DOT's do not have a common platform for sharing network data relevant to traffic assignment
  2. The planning agencies may also be very different in terms of technology used to run traffic assignment for their own transportation network

Network Partitioning

Objectives

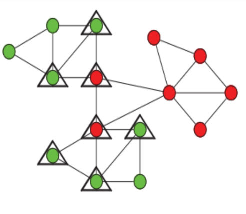
Therefore, the objective of this research is to develop partitioning algorithms that result in subnetworks which

1. optimize the computational performance of DSTAP
2. enhance the application of this algorithm considering the needs of distinct metropolitan planning agencies in a megaregion

Algorithm 1:

Shortest domain decomposition algorithm (SDDA) (Johnson et al., 2016)

- An agglomerative heuristic that aims to minimize boundary nodes as a primary objective and to maintain balanced partitions



Algorithm 2:

Flow-weighted spectral partitioning

- This algorithm partitions the network based on the second smallest eigenvalue of the flow-weighted normalized graph Laplacian
- This minimizes the inter-flow between subnetworks and creates subnetworks that are balanced by total flow

$$L_{symm} = D_G^{-1/2} L_G D_G^{-1/2} \tag{1}$$

$$L_G \varphi_i = \lambda_i \varphi_i \tag{2}$$

$$\lambda_i = \min_{S\ of\ dim\ i} \max_{x \in S} \frac{x^T L_G x}{x^T x} \tag{3}$$

$$\varphi_i = arg \min_{S\ of\ dim\ i} \max_{x \in S} \frac{x^T L_G x}{x^T x} \tag{4}$$

Computation time per DSTAP iteration

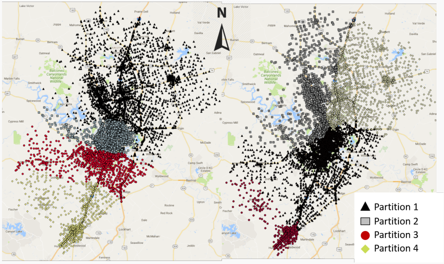
Network	Boundary nodes	Time (s)
Austin (SDDA)	174	632.83
Austin (Spectral)	329	746.81
Austin (4 subnets, SDDA)	296	290.97
Austin (4 subnets, spectral)	440	82.50
Anaheim (SDDA)	46	0.10
Anaheim (Spectral)	48	0.13
Chicago sketch (SDDA)	74	9.79
Chicago sketch (Spectral)	50	7.42

Network Partitioning

Partitioning Results

Computation time per DSTAP iteration

- SDDA generates partitions with a lower number of boundary nodes
- Flow balanced partitions are important for parallelizing subproblems. For Austin (4 partitions), SDDA subproblems require 3.5 times more computation time due to the flow imbalance



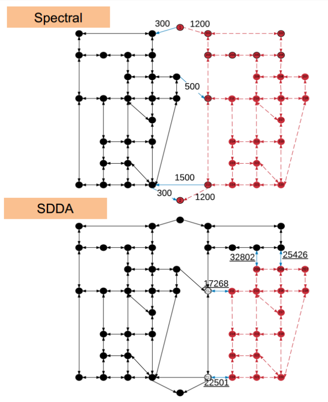
DSTAP convergence rate

Network	Inter-flow
Austin (SDDA)	186161
Austin (Spectral)	137940
Austin (4 subnets, SDDA)	368718
Austin (4 subnets, spectral)	296870
Anaheim (SDDA)	81991
Anaheim (Spectral)	56539
Chicago sketch (SDDA)	154791
Chicago sketch (Spectral)	201603

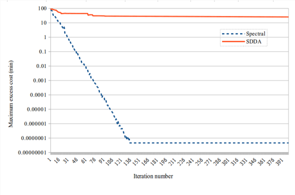
Network Partitioning

Partitioning Results

DSTAP convergence rate



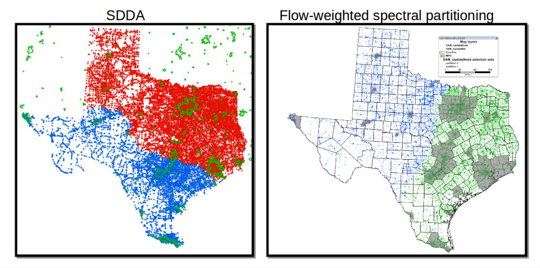
- Spectral partitioning generates subnetworks with lower inter-flow
- For a hypothetical network of two Sioux Falls networks connected by low demand, DSTAP converges faster with spectral partitioning subnetworks



Network Partitioning

Partitioning Results

Separating MPO's into distinct subnetworks



- The full Texas network was divided into two subnetworks using SDDA (left) & flow-weighted spectral partitioning (right)
- The cut generated by SDDA crosses through the Austin area, dividing Austin into two portions across each subnetwork
- The cut generated by the spectral partitioning algorithm keeps the region controlled by a planning agency intact within one subnetwork

Conclusions

- Traffic assignment across a megaregion has significant benefits for planning, but solving this problem can be computationally challenging
- An efficient parallelization scheme that determines a global equilibrium solution was developed (DSTAP)
- Computation time per DSTAP iteration can be reduced by minimizing boundary nodes and balancing subproblems
  - SDDA generates partitions with a lower number of boundary nodes
  - Flow-weighted spectral partitioning generates subproblems with better balance

Conclusions

- DSTAP convergence rate depends on inter-flow between subnetworks
  - Flow-weighted spectral partitioning generates subnetworks with lower inter-flow
  - For a hypothetical network of two Sioux Falls networks, DSTAP converges faster with flow-weighted spectral partitioning subnetworks
- Assigning a planning agency to one distinct subnetwork is critical for a parallelization algorithm
  - Flow-weighted spectral partitioning for Texas did not divide any planning agency across subnetworks
  - SDDA partitions for Texas cut through the Austin area

Thank you! Questions?