	UTC Project Information – Cooperative Mobility for Competitive Megaregions (CM^2)
Project Title	Beyond Political Boundaries: Constructing Network Models for Megaregion
University	University of Texas at Austin
Principal Investigator	Stephen Boyles
PI Contact Information	sboyles@mail.utexas.edu
Funding Source(s) and Amounts Provided (by each agency or organization)	U.S. Department of Transportation: $41,481 UT Austin (reduced overhead and donated salary): $19,505 TxDOT: $1,235.50
Total Project Cost	$62,221.50
Agency ID or Contract Number	UTDOT Grant number: 69A3551747135
Start and End Dates	11/1/2017 - 11/31/2018
Brief Description of Research Project	The objective of this project is to explore how to build networks which are suitable for long-range planning in megaregions, and how to solve them efficiently given their large size and high degree of interconnection. The research team will leverage DSTAP, a recently-developed framework for "decentralizing" network assignment problems, which has promise for efficiently solving traffic assignment on networks with a good deal of structure or hierarchy, as with cities in a megaregion.
Describe Implementation of Research Outcomes (or why not implemented)	Dr. Boyles presented with Graduate Research Assistant Cesar Yahia on this research at CTR in 2018. The presentation was titled "Network Partitioning Algorithms for Solving the Traffic Assignment Problem using a Decomposition Approach." He produced two journal articles titled "Network Partitioning Algorithms for Solving the Traffic Assignment Problem using a Decomposition Approach" and "A decomposition approach to the static traffic assignment problem."
Impacts/Benefits of Implementation (actual, not anticipated)	This project assisted with a quantitative comparison of methods for developing large, multi-city networks, and partitioning schemes for efficient and accurate solution.
Web Links (to reports, project website, etc.)	https://journals.sagepub.com/doi/full/10.1177/0361198118799039 https://ideas.repec.org/a/eee/transb/v105y2017icp270-296.html

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

CENTER FOR TRANSPORTATION RESEARCH

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

The University of Texas at Austin

TEXAS
The University of Texas at Austin

Background

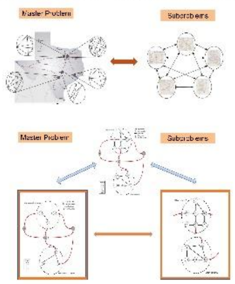
- Recent methods in the literature to **parallelize the static traffic assignment problem** consider partitioning a network into subnetworks to reduce the computation time for large-scale networks
- We seek a partitioning method that generates subnetworks minimizing the computation time of a decomposition approach for solving the traffic assignment problem (**DSTAP**)
- **Research goal**: we aim to minimize the number of boundary nodes between subnetworks, inter-flow, and computation time when the traffic assignment problem (TAP) is solved independently and in parallel for each subnetwork. This will minimize the time needed to reach global equilibrium using DSTAP
- **Method**:
  - We test two different partitioning algorithms.
    - The first algorithm is an agglomerative clustering heuristic developed by Johnson et al. The primary objective of this heuristic is to minimize the number of boundary nodes
    - The second algorithm is a flow-weighted spectral partitioning algorithm. The primary objective of this algorithm is to minimize the inter-flow between subnetworks

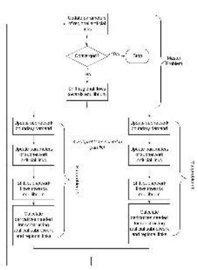
Decomposition Approach to the Static Traffic Assignment Problem (DSTAP)

DSTAP is an iterative aggregation-disaggregation algorithm for the solving the static traffic assignment problem in parallel. The algorithms consists of two levels: a master problem and a set of lower level subproblems corresponding to the respective subnetworks

- The master problem is an aggregated representation of the full network. Subnetworks are aggregated using first order approximations based on equilibrium sensitivity analysis. The master problem models interactions between subproblems
- The subproblems correspond to solving the static traffic assignment problem for a specific subnetwork. Subproblems interact with the full network through boundary flows

The algorithm converges to global equilibrium as follows: 1) the subproblems are solved in parallel independently 2) subnetworks are aggregated using first order approximation to form artificial links 3) flows are shifted towards equilibrium in the simplified master problem 4) subnetwork boundary flow is obtained from the master level iteration 5) subnetwork flow is disaggregated and the subproblems are solved in parallel again. The flowchart below shows this process.





Partitioning Objectives & Algorithms

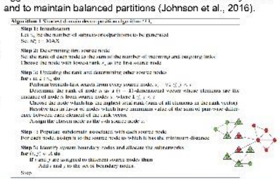
DSTAP computational performance

1. The *computation time at each iteration* depends on the number of artificial links. These artificial links need to be updated at each iteration using equilibrium sensitivity analysis. The number of subnetwork boundary nodes needs to be minimized to reduce regional artificial links
2. The *computation time at each iteration* is also influenced by the time needed to solve TAP for each subproblem in parallel. This is dominated by the subproblem that requires the greatest computational cost. Balanced subproblems would minimize this computational cost.
3. Based on the bound for maximum excess cost ϵ_{OD} of the full network shown below. For a fixed number of boundary nodes, faster *convergence rate* towards global equilibrium could be reached by minimizing inter-flow between subnetworks. This follows from minimization of the master level excess cost at a higher rate when flow is contained in subnetworks (relatively invariant artificial links across iterations).

$$\epsilon_{OD} < 2\dot{B}(\epsilon'_{OD} + \epsilon^{t}_{OD})$$

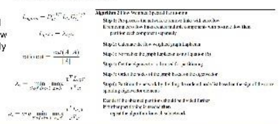
Algorithm 1: SDDA

Agglomerative heuristic that aims to minimize boundary nodes and to maintain balanced partitions (Johnson et al., 2016).



Algorithm 2: Flow-weighted Spectral Partitioning

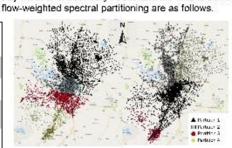
Partitioning the network based on the second smallest eigenvalue of the flow-weighted normalized graph Laplacian. This minimizes inter-flow and creates flow balanced subnetworks.



Results: Computation time per DSTAP iteration

Computation time per DSTAP iteration is dominated by number of boundary nodes and the time to solve the subproblems in parallel. Results for SDDA and flow-weighted spectral partitioning are as follows.

Network	Boundary Nodes	Subnetwork Computation time (s)
Austin (SDDA)	174	632.83
Austin (Spectral)	329	748.61
Austin 4 subnets (SDDA)	266	290.87 (65% of total flow)
Austin 4 subnets (Spectral)	440	82.60
Anaheim (SDDA)	48	0.10
Anaheim (Spectral)	48	0.10
Chicago (SDDA)	74	8.78 (90% of total flow)
Chicago (Spectral)	50	7.12

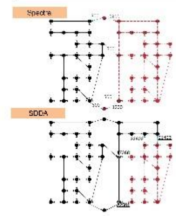


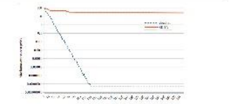
- Flow balanced partitioning is superior for solving TAP in subproblems as shown for Austin network with 4 partitions. SDDA computational cost for solving subproblems is 3.5 times the corresponding cost for spectral partitioning.
- SDDA generated partitions with a lower number of boundary nodes.

DSTAP Convergence Rate

DSTAP convergence rate is dominated by inter-flow between subnetworks. Results for SDDA and flow-weighted spectral partitioning are as follows. Convergence rate was computed for a network composed of two Sioux Falls networks connected by artificial demand. Artificial demand was kept at 1.5% of total demand within each network.

Network	Austin (SDDA)	Austin (Spectral)	Austin 4 subnets (SDDA)	Austin 4 subnets (Spectral)	Anaheim (SDDA)	Anaheim (Spectral)	Chicago (SDDA)	Chicago (Spectral)
Inter-flow	166161	157940	266716	206870	81991	58538	154791	201903





- Spectral partitioning generated partitions with lower inter-flow. Chicago partitions for SDDA have lower inter-flow but the partitions are heavily imbalanced (90% of flow is in one subnetwork)
- Spectral partitioning was able to find the proper cut for a hypothetical network composed of two connected Sioux Falls networks. SDDA failed since it only considers boundary nodes. This has undesirable implications for statewide planning models with flows concentrated in cities.
- In terms of convergence rate for the hypothetical Sioux Falls network, DSTAP converged much faster with partitions generated from the spectral partitioning algorithm.

Conclusions and Future work

Conclusions:

1. Computation time per iteration of DSTAP can be reduced by minimizing number of boundary nodes and balancing subproblems
   1. The subproblems can be balanced by creating flow balanced subnetworks using the flow-weighted spectral partitioning algorithm
   2. For the Austin network partitioned into four regions, the SDDA computational cost for solving subproblems is 3.5 times the corresponding cost for spectral partitioning
   3. SDDA performed better at minimizing the number of boundary nodes
2. DSTAP convergence rate depends on inter-flow between subnetworks
   1. Spectral partitioning generates partitions with lower inter-flow as compared to SDDA partitions
   2. For a hypothetical network of two Sioux Falls networks connected by low levels of demand, DSTAP converged faster using subnetworks from flow-weighted spectral partitioning
3. SDDA relies on number of boundary nodes only. In statewide planning models, SDDA may not identify cities concentrated with flow as separate subnetworks. This was shown for the hypothetical Sioux Falls network.

Future work:

1. Assess trade-offs between minimizing per iteration computation time and maximizing convergence rate
2. Seek alternative approximations that minimize the number of boundary nodes in DSTAP

Acknowledgement: This research was supported by the National Science Foundation under Grant No. 1254921. Partial support was provided by the Data-Supported Transportation Operations & Planning University Transportation Center (D-STOP). Partial support was provided by the Cooperative Mobility for Competitive Megaregions center (CM2). The authors are grateful for this support