High Performance Dynamic Traffic Assignment Based on Variational Inequality

GU Yangsong, LIANG Geyu

Mentor: Dr. Cheng LIU, Dr. Kwai WONG

06/28/2018
Outline

1. Introduction
   - Dynamic Traffic Assignment

2. Progress
   - Dynamic Network Loading Based on ODE
   - Dynamic Network Loading Based on LWR
   - Variational Inequality

3. Implementation
   - CVODE in SUNDIAL
   - Interpolators in boost library
   - Ceres Library

Which runs on GPU
Heat Map Based on Vehicle Density

Chart of Traffic Flow
Introduction

Dynamic traffic assignment is the positive modeling of time-varying flows of automobiles on road network consistent with established traffic flow theory and travel demand theory.

A Simple Network

- Nodes:
- Links
- Origin-Destination Pair
- Time cost = Delay = Travel Time
Introduction

Sioux Fall Network

Departure rate function and cost function
Introduction

Continuous Time Dynamic User Equilibrium (DUE)

- Users choose the path with the **minimum** travel time, and the effective travel delay is **identical** for all the path and departure time of the same travel purpose.

Desired solution:

- Desired solution:
  - The **departure rate function** for each path
  - The corresponding **cost function**

![Fig Departure rates and corresponding travel cost in the DUE solution](image-url)
**Progress**

**Flow chart**

- **Get path delay** → **Made up of** → **Get link delay** → **Dependent on** → **Get link volume** → **Solved By** → **ODE Dynamic Network loading**

  - **ODE Group**

(Each OD pair)

- **DUE principle** → **DVI Problem** → **Solved by** → **Fixed-Point Problem** → **Get result** → **The best departure time and path choice**

- **Update**
Progress

Part 1: Dynamic Network Loading

Link state equation
\[
\frac{dx_{a_i}^p(t)}{dt} = h_{p}^{\tau,k}(t) - g_{a_i}^p(t) \quad \forall p \in P
\]
\[
\frac{dx_{a_i}^m(t)}{dt} = g_{a_{i-1}}^p(t) - g_{a_i}^p(t) \quad \forall p \in P, i \in [2, m(p)]
\]

Medium equation
\[
\frac{dg_{a_i}^p(t)}{dt} = r_{a_i}^p(t) \quad \forall p \in P, i \in [1, m(p)]
\]
\[
\frac{dr_{a_i}^p(t)}{dt} = R_{a_i}^p(x, g, r, h^{\tau,k}) \quad \forall p \in P
\]
\[
\frac{dr_{a_i}^m(t)}{dt} = R_{a_i}^m(x, g, r) \quad \forall p \in P, i \in [2, m(p)]
\]

Initial conditions
\[
x_{a_i}^p((\tau - 1) \cdot \Delta) = x_{a_i}^p_0 \quad \forall p \in P, i \in [1, m(p)]
\]
\[
g_{a_i}^p((\tau - 1) \cdot \Delta) = 0 \quad \forall p \in P, i \in [1, m(p)]
\]
\[
r_{a_i}^p((\tau - 1) \cdot \Delta) = 0 \quad \forall p \in P, i \in [1, m(p)]
\]
Progress

Part 2: Convert DUE to DVI

\[ \text{find } h^* \in A_0 \text{ such that } \left\{ \sum_{p \in P_{ij}} \int_{t_0}^{t_f} \Psi_p(t, h^*)(h_p - h^*) \, dt \geq 0 \quad \forall h \in A \right\} \]

Where spatial condition is

\[ \frac{dy_{ij}}{dt} = \sum_{p \in P_{ij}} h_p(t) \quad \forall (i,j) \in \mathcal{W} \]

\[ y_{ij}(t_0) = 0 \quad \forall (i,j) \in \mathcal{W} \]

\[ y_{ij}(t_f) = Q_{ij} \quad \forall (i,j) \in \mathcal{W} \]

satisfy Flow propagation constraints
Progress

Part 3: Solve DVI by Fixed-Point Iteration

Equal solution

For each iteration step

Update new departure rate
Loop of fixed point algorithm

Start

Initialization of h(t)

Calculate the delay operator Phi by solving the ODE system

Calculate the subproblem dual variable v for each OD pair

Calculate hk+1 based on v and Phi(hk)

Check if hk and hk+1 satisfies the error condition

Output the result

NO

YES
Example
Implementation

Current Work:
- modify MATLAB
- convert MATLAB code to C code
  (enlarge graph scale)

ODE solver:
- CVODE in SUNDIAL
- odeint in boost library

Interpolator:
- Interpolators in boost library

Root solver:
- CERES Library
Feature Work: Dynamic Network Loading Based on LWR
Dynamic Network Loading Based on LWR

LWR model

Microcosmic Based on PDE

\[
\begin{align*}
\partial_t \rho^e(t, x) + \partial_x f^e(\rho^e(t, x)) &= 0, \quad (t, x) \in [t_0, t_f] \times [a^e, b^e] \\
\rho^e(0, x) &= 0, \quad x \in [a^e, b^e]
\end{align*}
\]

Advantages capture

1. Queues and delay
2. Density-speed relationship
3. First-in-First-out principle
4. Route information
Dynamic Network Loading Based on LWR

Link Flow Propagation
The solution of PDE in the form of Lax-Hopf formula.

NetWork Loading

Question

Link delay
Reference


End

Thank You