

High Performance Traffic Assignment Based on Variational Inequality

Mentee: XIAO Yujie (CityU), SHI Zhenmei (HKUST)
Mentor: LIU Cheng (ORNL), WONG Kwai (UTK)
zshiad@connect.ust.hk, evening810808@gmail.com
cheng.n.liu@gmail.com, kwong@utk.edu

August 4, 2017

Abstract

Variational Inequality (VI) is a mathematical problem that is widely applied to equilibrium problems in different fields. This project focuses on modeling the transportation assignment problem using variational inequality and solving it in a highly efficient way. The solver implements on GPU using parallel computing. The problem can be classified into two categories Static Traffic Assignment (STA) and Dynamic Traffic Assignment. STA, in this project, refers to solving for the user equilibrium which is a Nash equilibrium about the travel cost of each user without considering about demand changing over time. DTA refers to solving for the dynamic user equilibrium where the cost of travel is minimized for every user, in a continuous period of time

1 Introduction

1.1 Traffic Assignment

Traffic assignment is a kernel component in transportation planning and real-time applications in optimal routing, signal control, and traffic prediction in traffic networks. The problem is represented by graphs (or networks), which include set of nodes (or vertices or points) and a set of links (or arcs or edges) connecting these nodes. Each network link is typically associated with some impedance that affects the flow using it. Traffic assignment models the flow pattern in a network given a set of travel demands between the origin destination (OD) pairs. The most widely used route choice model is the user-equilibrium (UE) principle. This UE assignment finds the flow pattern by allocating the OD demands to the network in such a way that no drivers can unilaterally change routes to achieve better travel times. (Hong,1999)

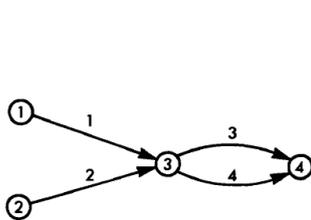


Figure 1: Network

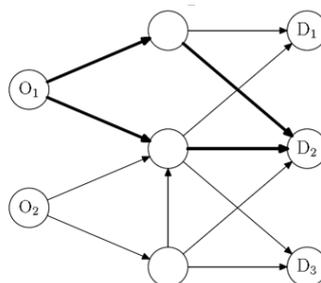


Figure 2: OD pair

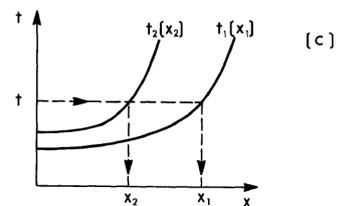


Figure 3: Equilibrium

Traffic Assignment Definition

Given:

1. A graph representation of the urban transportation network
2. The associated link performance functions
3. An origin-destination matrix

Find the flow (and travel time) on each of the network links, such that the network satisfies UE principle.

Traffic Assignment Time Cost Function

$$time = freeflowtime * (1 + B * (flow/capacity)^{Power})$$

1.2 Variational Inequality (VI)

In mathematics, a VI is an inequality involving a functional, which has to be solved for all possible values of a given variable, belonging usually to a convex set. The applicability of the theory has been expanded to include problems from economics, finance, optimization and game theory.

VI Definition

In general form, a VI is formally defined below.

Given a subset K of the Euclidean n -dimensional space Rn and a mapping $F : K \rightarrow Rn$, the variational inequality, denoted $VI(K, F)$, is to find a vector $x \in K$ such that $(y - x)^T F(x) \geq 0, \forall y \in K$. The set of solutions to this problem is denoted $SOL(K, F)$.

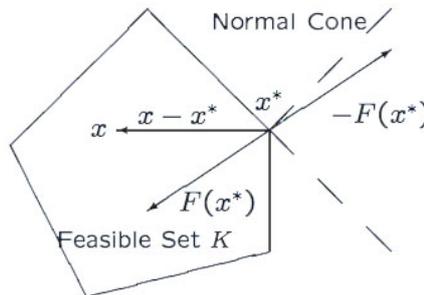


Figure 4: Variational Inequality

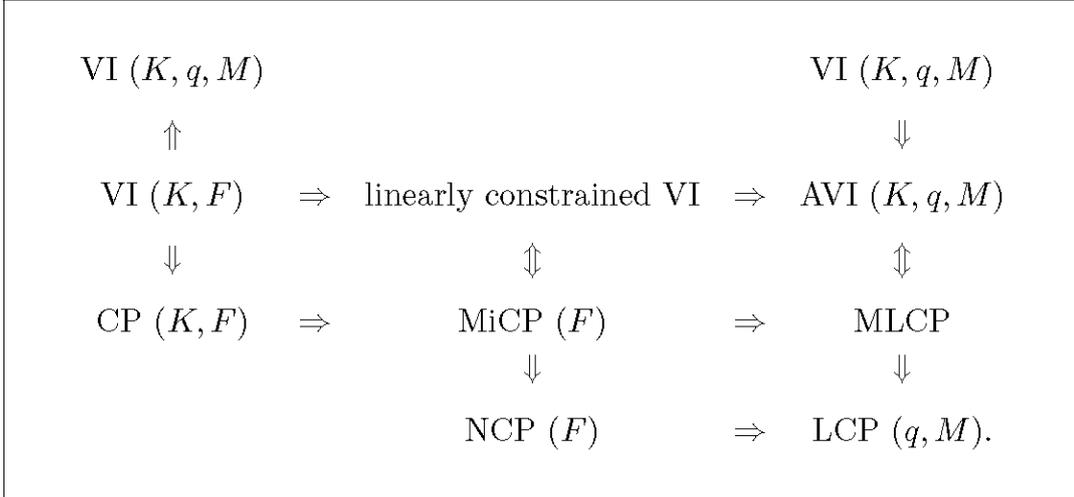
There is a special case of VI called Nonlinear Complementarity Problem (NCP), which will be used latter.

NCP definition

In general form, a variational inequality is formally defined below.

Given a mapping $F : Rn_+ \rightarrow Rn$, the $NCP(F)$ is to find a vector $x \in Rn$ satisfying $0 \leq x \perp F(x) \geq 0$.

Variational Inequality Category



1.3 Static Traffic Assignment

Static Traffic Assignment(STA) refers to solving for the user equilibrium which is a Nash equilibrium about the travel cost of each user without considering about demand changing over time. The purpose of a static traffic equilibrium model is to predict steady-state traffic flows in a congested traffic network. The node set of the network is denoted N and arc set is denoted A . It is assumed that users of the network compete non-cooperatively for the resources of the network in an attempt to minimize their travel costs, where the cost of travel along an arc $a \in A$ is a nonlinear function $c_a(f)$ of the total flow vector f with components f_b for all $b \in A$. Let $c(f)$ be the vector with components $c_a(f)$, $a \in A$. There are two distinguished subsets of N that represent the set of origin nodes O and destination nodes D , respectively. The set of origin-destination (OD) pairs is a given subset W of $O \times D$. The traditional algorithm to solve STA problem is Frank - Wolfe algorithm which is mainly use the shortest path algorithm. We will convert STA problem to be NCP formulation to solve it and compare the performance with the Frank - Wolfe algorithm. (Yosef,1984)

1.4 Dynamic Traffic Assignment

Dynamic traffic assignment(DTA) is the positive modeling of time-varying flows of automobiles on road network consistent with established traffic flow theory and travel demand theory. And in this project we concentrate on dynamic user equilibrium(DUE), one type of DTA wherein the effective unit travel delay, including early and late arrival penalties, of travel for the same purpose is identical for all utilized path and departure time pairs.

1.4.1 Dynamic network loading

To solve for the equilibrium, first a dynamic network loading model should be established which will return the link activity when travel demand and departure rates are provided. Effective path delays are constructed from arc delays that, directly or indirectly, depend on arc activity; moreover, activity on a given arc is influenced by the delays on paths that utilize that arc. Thus, dynamic network loading is intertwined with the determination of path delays.

After obtaining the most crucial ingredient of a user equilibrium model, the path delay operator $D_p(h)$, we can derive the cost function corresponding to the input $h(t)$ by adding the scheduled delay $F[t + D_p(h) - T_A]$, which is the final result of the DNL procedure and will be used in the next step.

1.4.2 DUE and its DVI formulation

In the general case, different users will, because of the congestion externalities, depart at different times and travel over different routes, while all users whose travel has the same purpose will expe-

rience the same disutility. It is this generalization of user disutilities together with the associated time-varying network flows and costs which we define dynamic user equilibrium.

Using measure theoretic arguments, Friesz et al. (1993) established that a dynamic user equilibrium is equivalent to a differential variational inequality (DVI) under suitable regularity conditions. And it can be solved numerically by restating it into a fixed point iteration.

2 Problem Definition

Basic Network Notation

N ,	node (index) set
A ,	arc (index) set
R ,	set of origin nodes; $R \subseteq N$
S ,	set of destination nodes; $S \subseteq N$
K_{rs} ,	set of paths connecting O-D pair r - s ; $r \in R, s \in S$
W ,	OD pair set
P ,	Path set
h_p ,	flow on path p ; $h = (\dots, h_p, \dots)$
x_a ,	flow on arc a ; $x = (\dots, x_a, \dots)$
t_a ,	travel time on arc a ; $t = (\dots, t_a, \dots)$
f_k^{rs} ,	flow on path k connecting O-D pair r - s ; $f^{rs} = (\dots, f_k^{rs}, \dots)$; $f = (\dots, f^{rs}, \dots)$
c_k^{rs} ,	travel time on path k connecting O-D pair r - s ; $c^{rs} = (\dots, c_k^{rs}, \dots)$; $c = (\dots, c^{rs}, \dots)$
q_{rs} ,	trip rate between origin r and destination s ; $(q)_{rs} = q_{rs}$
$\delta_{a,k}^{rs}$,	indicator variable: $\delta_{a,k}^{rs} = 1$ if link a is on path k between O-D pair r - s ; otherwise $\delta_{a,k}^{rs} = 0$ (Δ^{rs}) $_{a,k} = (\dots, \delta_{a,k}^{rs}, \dots)$; $\Delta = (\dots, \Delta^{rs}, \dots)$
$x_{a_i}^p(t)$,	arc volume function of the i -th link of the path p
$g_{a_i}^p(t)$,	exit flow function of the i -th link of the path p
$r_{a_i}^p(t)$,	the first derivative of exit flow function of the i -th link of the path p
$h_p(t)$,	departure rate function

The incidence matrix for the network example depicted in Figure 1 can be written as follows:

$$\begin{array}{c}
 \text{link} \backslash \text{path} \\
 \begin{array}{c}
 1 \\
 2 \\
 3 \\
 4
 \end{array}
 \begin{array}{cc}
 \text{O-D} & \text{O-D} \\
 \underbrace{1-4} & \underbrace{2-4} \\
 \begin{array}{cc}
 1 & 2 \\
 1 & 2
 \end{array}
 \end{array}
 \begin{pmatrix}
 1 & 1 & 0 & 0 \\
 0 & 0 & 1 & 1 \\
 1 & 0 & 1 & 0 \\
 0 & 1 & 0 & 1
 \end{pmatrix}
 \end{array}$$

3 Methodology

3.1 Static Traffic Assignment (STA)

There are several descriptions of the static traffic equilibrium problem; one description is in terms of the flows on paths in the network, and another is in terms of multicommodity flows on arcs, where each commodity represents one OD pair.

3.1.1 The path formulation

A common assumption on the path cost function $C(h) \equiv (C_p(h))$ is that it is additive; that is, for each $p \in P$, $C_p(h)$ is the sum of the arc costs $c_a(f)$ on all the arcs a traversed by the path $p \in P$. In vector notation, this assumption says $C(h) = \Delta^T c(f) = \Delta^T c(\Delta h)$. On the demand side, for each $w \in W$ a function $d_w(u)$ is given that represents the travel demand between the OD pair w , where

$u \equiv (u_w)$ is the (unknown) vector of minimum travel costs between all OD pairs. This general case corresponds to an elastic demand model; it is in contrast to the fixed demand model where $d_w(u)$ is a constant for all $w \in W$. The user equilibrium principle is a behavioral axiom that postulates the route choice of the network users. Specifically, the principle stipulates that users of the traffic network will choose the minimum cost path between each OD pair, and through this process the paths that are used (i.e., have positive flows) will have equal costs; moreover, paths with costs higher than the minimum will have no flow. Mathematically, this principle can be phrased succinctly as follows:

$$0 \leq C_p(h) - u_w \perp h_p \geq 0, \forall w \in W \text{ and } p \in P_w;$$

moreover, the travel demand must be satisfied:

$$\sum_{p \in P_w} h_p = d_w(u), \forall w \in W,$$

and the minimum travel costs must be nonnegative:

$$u_w \geq 0, w \in W.$$

The static traffic user equilibrium problem is to find a pair (h, u) of path flows and minimum travel costs, called a traffic user equilibrium, so that the above conditions are satisfied. At first glance, these conditions are not quite in the form of a complementarity problem or variational inequality. Under a reasonable assumption on the travel cost and demand functions, the conditions are indeed equivalent to an $NCP(F)$ where the defining function F is given by:

$$F(h, u) \equiv \begin{bmatrix} C(h) - \Omega^T u \\ \Omega h - d(u) \end{bmatrix}$$

where Ω is the (OD pair, path)-incidence matrix whose entries are $w_{wp} \equiv 1$ if $p \in P_w$, 0 otherwise. Note that the matrix size is the number of path add the number of OD pair. (Jong-Shi,2002)

3.1.2 Path finding

In NCP formulation, path is kernel data information. In general, input data does not have any information about that. Therefore, finding path is necessary. However, in a big graph, it is meaningless to find all path for each OD pair. Finding candidate path for each OD pair is our method. Shortest k paths algorithm is a nice method to find candidate path, but it has flaws. Usually, paths found by shortest k paths algorithm are similar. Most their arcs are overlapping. It is noneffective when we assign flow on these similar path. To find non-similar path, we use shortest path algorithm other than shortest k path algorithm. Each time we find a shortest path for an OD pair, we double the time cost of all arcs on this path. Then, run shortest path algorithm again to find a new path. Due to all the arcs' time cost of last shortest path have increased, the new path will avoid these arcs probably. We do this iteration 7 times to find at most 7 candidate path for each OD pair. The reason of 7 will be discussed later. Note that this method can reduce matrix size significantly, although the result of this method is approximate optimization.

3.2 Dynamic Traffic Assignment (DTA)

3.2.1 DNL

ODE Approximation Dynamic network loading

(Friesz,2011)

$$\frac{dx_{a_1}^p(t)}{dt} = h_p^k(t) - g_{a_1}^p(t) \quad (1)$$

$$\frac{dx_{a_i}^p(t)}{dt} = g_{a_{i-1}}^p(t) - g_{a_i}^p(t) \quad (2)$$

$$\frac{dg_{a_i}^p(t)}{dt} = r_{a_i}^p(t) \quad (3)$$

$$\frac{dr_{a_1}^p(t)}{dt} = R_{a_1}^p(x, g, r, h^{\tau, k}) \quad (4)$$

$$\frac{dr_{a_i}^p(t)}{dt} = R_{a_i}^p(x, g, r) \quad (5)$$

$$x_{a_i}^p((\tau - 1) * \Delta) = x_{a_i}^{p,0} \quad (6)$$

$$g_{a_i}^p((\tau - 1) * \Delta) = 0 \quad (7)$$

$$r_{a_i}^p((\tau - 1) * \Delta) = 0 \quad (8)$$

3.2.2 dVI

For each origin-destination pair, the actual flow unit costs from time of departure to time of arrival on utilized paths, including any early or late arrival penalties, are identical and equal to the minimum unit path cost which can be realized from among all route choice and departure time decisions, the corresponding flow pattern is said to be a simultaneous route -departure equilibrium.

(Friesz,1993)

An equilibrium for simultaneous route choice and departure time decisions is defined as following:

The pair (h, μ) is designated as a simultaneous route-departure equilibrium if and only if the following two conditions are satisfied:

$$h_p(t) > 0 \Rightarrow C_p(t, h) = \mu_{kl} C_p(t, h) \geq \mu_{kl}$$

for all p and for all $k, l \in N$.

The proof is presented in (Friesz,1993).

(Friesz,2013)

$DVI(\Psi, \Lambda, [t_0, t_f])$: find $h^* \in \Lambda_0$ such that

$$\left\{ \begin{array}{l} \sum_{p \in P} \int_{t_0}^{t_f} \Psi_p(t, h^*) (h_p - h_p^*) dt \geq 0 \forall h \in \Lambda \\ \text{where } \Lambda = h \geq 0 : \frac{dy_{ij}}{dt} = \sum_{p \in P_{ij}} h_p(t), y_{ij}(t_0) = 0, y_{ij}(t_f) = Q_{ij} \end{array} \right. \quad (9)$$

$$\left\{ \begin{array}{l} \sum_{p \in P} \int_{t_0}^{t_f} \Psi_p(t, h^*) (h_p - h_p^*) dt \geq 0 \forall h \in \Lambda \\ \text{where } \Lambda = h \geq 0 : \frac{dy_{ij}}{dt} = \sum_{p \in P_{ij}} h_p(t), y_{ij}(t_0) = 0, y_{ij}(t_f) = Q_{ij} \end{array} \right. \quad (10)$$

4 Implementation

4.1 Static traffic assignment

The whole algorithm is in 3 steps.

4.1.1 Step1: Path finding

Use One to All shortest path algorithm to find 7 paths for each OD pair. Here the solver uses nvGRAPH package in CUDA library which runs on GPU.

```
nvgraphStatus_t nvgraphSssp (nvgraphHandle_t, const nvgraphGraphDescr_t ,
const size_t, const int *, const size_t);
```

4.1.2 Step2: Convert to NVP formulation

Convert all data in to NCP formulation in Siconos, which is a non-smooth numerical simulation package. There are two matrix. One is F mapping matrix and the other one is Jacobian matrix, which represents the relationship between path flow and cost on OD pair. Both matrix is sparse, because each path is only associated with a few arcs compared with the whole graph.

4.1.3 Step3: Solve NCP problem

Use NCP FBLSA Algorithm to solve the problem with given error bound. Here the solver uses Siconos and MUMPS library, which is a parallel sparse direct solver using MPI.

```
info = ncp_driver(problem, z, F, &options);
```

4.2 Dynamic traffic assignment

4.2.1 DNL

Given $h(t)$, then the ODE system is known as equations (1) (8). And arc volume $x(t)$ is the solution to the ODE system.

From $x(t)$ to derive traversal time $D_p(t)$:

$$D_p = \sum_{i=1}^{m(p)} [\tau_{a_i}^p(t) - \tau_{a_{i-1}}^p(t)] = \tau_{a_{m(p)}}^p(t) - t \quad (11)$$

$$\tau_{a_1}^p(t) = t + D_{a_1}[x_{a_1}(t)] \quad (12)$$

$$\tau_{a_i}^p(t) = \tau_{a_{i-1}}^p(t) + D_{a_i}[x_{a_i}(\tau_{a_{i-1}}^p(t))] \quad (13)$$

$$D(x) = \alpha * x + \beta \quad (14)$$

$$(15)$$

From $D_p(t)$ to derive cost function $\Phi(t)$:

$$\Phi_p(t) = D_p(t) + F[D_p(t) + t - T_A] \quad (16)$$

$$F(D_p(t) + t - T_A) = 0.5 * (D_p(t) + t - T_A)^2 \quad (17)$$

where F is the penalty function

4.3 DVI

The solution of DVI problem as in (19) (20) can be obtained by solving a fixed-point iteration:

$$h^* = P_\Lambda[h^* - \alpha\Phi(t, h^*)]$$

The proof is provided in Friesz,et al(2013).

For each iteration step, get a v value once $\Phi_p(t)$ for all p are obtained:

$$\sum_{p \in P_{ij}} \int_{t_0}^{t_f} [h_p^k(t) - \alpha\Phi(t, h_p^k) + v_{ij}]_+ = Q_{ij}$$

From $h^k(t)$ to h^{k+1} :

$$h_p^{k+1} = [h_p^k(t) - \alpha\Phi(t, h_p^k) + v_{ij}]_+$$

5 Numerical results

5.1 STA

5.1.1 Result

Result with 4 OD Pair and Result with 24 OD pair

The graphs on the left is the network. The graphs on the right is the assignment. The darker the line means there are more flow on that link.

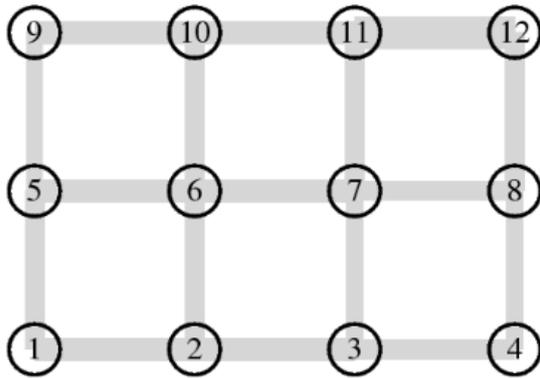


Figure 5: Network1

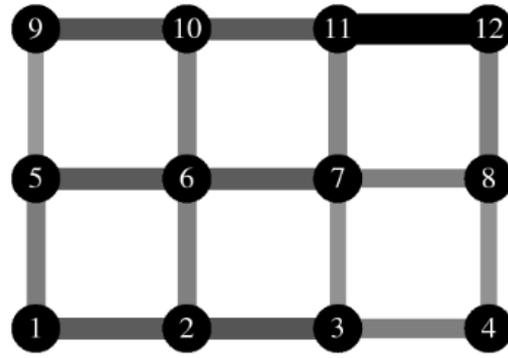


Figure 6: Assignment1

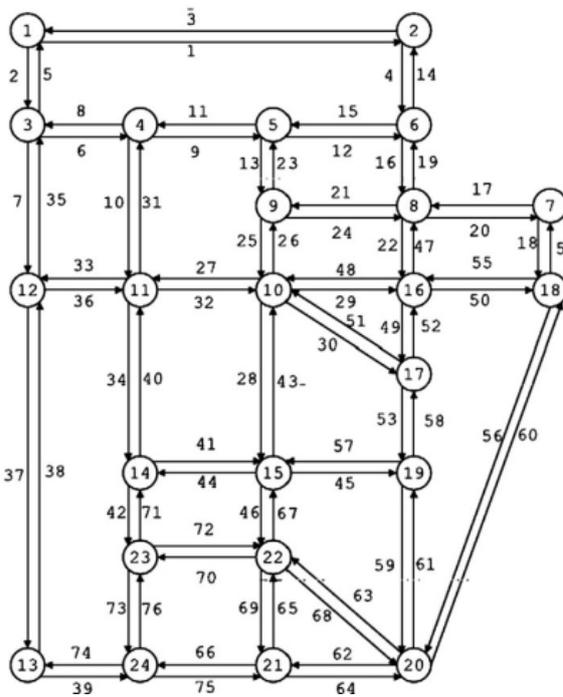


Figure 7: Network2

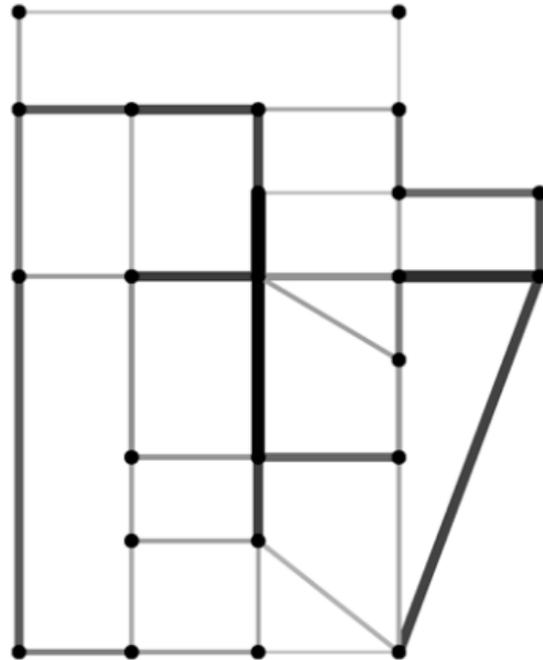
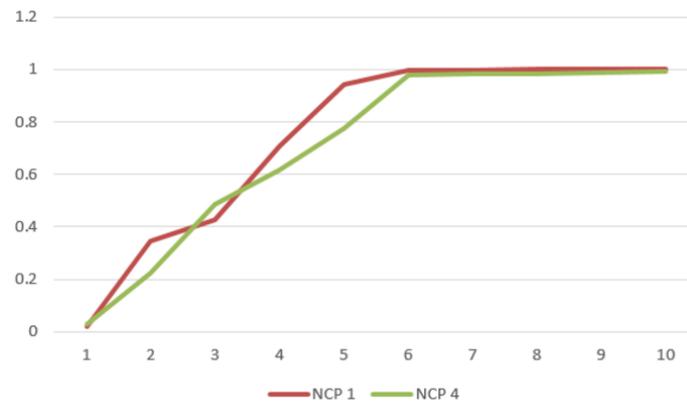


Figure 8: Assignment2

5.1.2 Analysis



The graph is about accuracy varies with path number for each OD pair. From the graph we can

see that 7 paths for each OD pair is optimal choice.
 Following we compare NCP algorithm and Frank Wolfe Algorithm.

NCP:

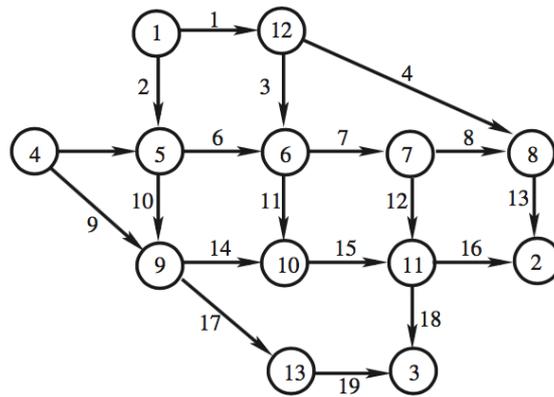
1. Dominant cost is Matrix solver
 2. Approximate optimize result
 3. A little faster when graph is big and with a few OD pair (Matrix size is OD pair number + path number)
- Running time is $O(n^3)$, where n is OD pair number

FW:

1. Dominant cost is shortest path algorithm
 2. Real Optimize result
 3. Faster when OD pair is more
- Running time is $O(n * m \log m)$, where n is OD pair number and m is node number

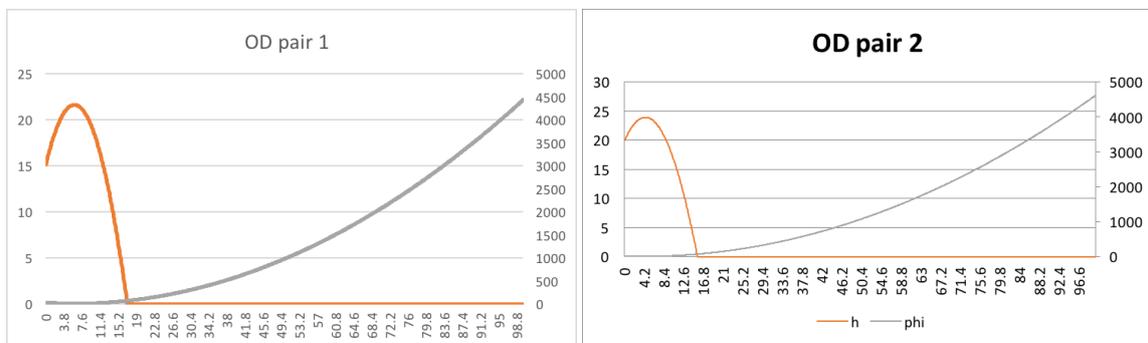
5.2 DTA

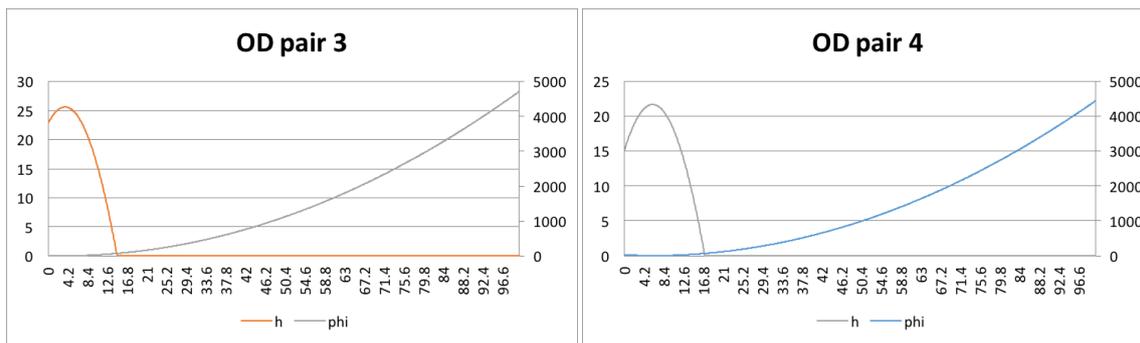
5.2.1 Result



We consider the 19-arc, 13-node network shown above, and the following four OD pairs are chosen: $W = (1,2), (1,3), (4,2), (4,3)$ each having a fixed travel demand of $Q_{1,2} = Q_{1,3} = Q_{4,2} = Q_{4,3} = 275$. The expected arrival time for the four OD pairs are $T_{(1,2)} = T_{(1,3)} = T_{(4,2)} = T_{(4,3)} = 12$. We utilize one path per OD pair. Specifically, $P = \{p_1, p_2, p_3, p_4\}$

$$p_1 = \{1, 4, 13\}, p_2 = \{2, 10, 17, 19\}, p_3 = \{9, 14, 15, 16\}, p_4 = \{9, 17, 19\}$$





5.2.2 Analysis

The result of the medium size network indicates that there is a symmetric pattern in the departure rate function when there is only one expected time, it is reasonable since the cost function involves a quadratic function.

What remains to solve is that the peak of the departure rate function appears earlier than the expected time. And that should be corrected when the input parameter is in accordance with the real life scenario.

6 Conclusion and Remarks

For STA problem, Frank Wolfe Algorithm is still better than NCP Algorithm in general. In special cases, when graph is big and number of OD - Pair is little NCP Algorithm is faster than Frank Wolfe Algorithm. When select 7 paths for each OD - pair in NCP algorithm, the result accuracy can reach 95%.

The performance can be significantly improved if cuSPARSE is plugged in, which is direct sparse matrix solver on GPU.

For DTA problem, the algorithm presented above can work fine for network of relatively small scale while it remains to see whether it performs well for larger network. Besides that, the road parameter configuration needs to be set more appropriately to simulate the practical case.

Also, the performance can be significantly improved if some parallel ODE solver is plugged in.

References

- Yosef, S., URBAN TRANSPORTATION NETWORKS: Equilibrium Analysis with Mathematical Programming Methods
- Jong-Shi, P., Finite-Dimensional Variational Inequalities and Complementarity
- Hong K. Lo, Traffic equilibrium problem with route-specific costs: formulation and algorithms
- Han, K., Friesz, T.L., Yao, T., in press. A partial differential equation formulation of Vickrey's bottleneck model, part I: Methodology and theoretical analysis. Transportation Research Part B.
- Han, K., Friesz, T.L., Yao, T., in press. A partial differential equation formulation of Vickrey's bottleneck model part II: Computation and application. Transportation Research Part B.
- Han, K., Friesz, T.L., Yao, T., 2012a. Existence of simultaneous route and departure choice dynamic user equilibrium. arXiv:1211.0898.
- Han, K., Piccoli, B., Friesz, T.L., Yao, T., 2012b. A continuous-time link-based kinematic wave model for dynamic traffic networks. arXiv:1208.5141v1.
- Han, K., Yao, T., Friesz, T.L., 2012c. Lagrangian-based hydrodynamic model: freeway traffic estimation. arXiv:1211.4619.
- Friesz, T.L., Han, L., Neto, P.A., Meimand, A., Yao, T., Dynamic user equilibrium based on a hydrodynamic model