Invited Tutorial on Network Economics: Dynamic Congestion Games

MITACS Workshop on Internet and Network Economics
Vancouver, British Columbia

30 May - 1 June 2011

Some Comments

1. Throughout this talk, the perspective is deterministic.
2. The principal physical reality we will mathematically describe is that of automobiles on an urban road network.
3. Our main goal is to acquire enough background to be able to state and discuss a theory of dynamic congestion games known as dynamic user equilibrium (DUE) on a network.

Some More Comments

1. An international community of scholars views the dynamic user equilibrium (DUE) problem as one of the foundation problems of operations research.
2. It is hoped that a theory of dynamic congestion games useful in both OR and computer science will evolve.
3. Once we have established a DUE model we will turn to our second goal: state and discuss a dynamic theory of mechanism design, known as dynamic congestion pricing.

Dynamic Traffic Assignment (DTA): A Class of Problems Including DUE

1. Descriptive modeling of time-varying flows on a road network consistent with travel demand and traffic flow theory.
2. Dynamic user equilibrium (DUE) is the primary form of dynamic traffic assignment presently being studied.
3. One form of DUE is <u>simultaneous route and departure choice</u> (SRD-choice) equilibrium.

Simultaneous Route and Departure Choice (SRD-choice) Dynamic User Equilibrium (DUE)

- SRD-choice DUE requires equal effective delay for utilized (path, departure time) pairs between a given origin-destination pair

SRD DUE Assumptions

1. Differential Nash game among users
2. Closed loop
3. Atomic
4. Deterministic
5. Perfect information
6. Today the above define the problem class principally studied by "DTA/DUE scholars."
7. Main topic of debate is what link-based traffic model to employ and how should that traffic model be integrated with the DUE formalism.
8. Shift to a more general problem class – relaxing some of items 1 thru 5 above – is imminent.

SRD DUE Model Components (Peeta & Ziliaskopoulos, 2001):

1. Model of path delay
2. Arc dynamics
3. Flow propagation constraints
4. Flow conservation and nonnegativity
5. Route and departure time choice model

Some Terminology: Differential Algebraic System

- An illustrative differential algebraic equation (DAE) is

$$F(x, \frac{dx}{dt}, u, t) = 0$$

- An illustrative partial differential algebraic equation (PDAE) is

$$F(x, \frac{\partial\rho(x,t)}{dx}, \frac{\partial\rho(x,t)}{\partial t}, u, t) = 0$$

- A key issue in DAE theory is index reduction, where "index" is the number of differentiations required to obtain an implicit ODE system from a DAE system.

Comment

- I am going to use a particular variety of DUE model, wherein the network loading phase is based on the so-called point-queue model (PQM), to illustrate how DUE models are anayzed and their solutions computed.

DTA History: Merchant and Nemhauser, 1974

Merchant & Nemhauser: *Transportation Science*, 12(3), 1974a and 1974b. Two papers.

1. Assignment: system optimal assignment
2. Dynamics: flow balance with exit-flow functions

$$\frac{dx_a}{dt} = u_a(t) - g_a[x_a(t)] \qquad \forall a \in \mathcal{A}$$

3. Delay: separable arc congestion functions form path delay
4. Computable: dynamic programming
5. Exit flow functions admit certain anomalies
6. Marks the birth of DTA literature
7. Influenced all later DTA and DUE work

Hendrickson & Kocur: *Transportation Science*, 15(1), 1981.

1. Assignment: SRD, time-dependent utility, single bottleneck
2. Dynamics: implicit
3. Delay: deterministic queueing
4. Computable: NA
5. A major influence in defining the SRD equilibrium

DTA History, Paper 3: Ran et al, 1993

Ran et al: *Operations Research*, 41(1), 1993.

1. Assignment: route-based; no departure time strategies; continuous-time optimal control
2. Dynamics: flow balance where entrance flow and exit flow are both controls

$$\frac{dx_a^{ij}}{dt} = u_a^{ij}(t) - v_a^{ij}(t) \qquad \forall a \in \mathcal{A}, (i,j) \in \mathcal{W}$$

3. Delay: highly specific arc delay functions
4. Computable: yes, NLP/discrete time approximation
5. Flow-propagation constraints must be added
6. Proposed equivalent optimal control problem has not withstood the test of time

DTA History, Paper 4: Friesz et al, 1993

Friesz et al: *Operations Research*, 41(1), 1993.

1. Assignment: Defines the SRD DUE in measure-theoretic terms using exit-time functions
2. Dynamics: integral equation involving exit-time functions
3. Delay: endogenous path delay, deterministic queueing: point queue model (PQM)
4. Computable: This formulation is "notoriously difficult to solve ..." (Nie and Zhang, 2010)

DTA History, Paper 5 & 6: Chen and Florian, 1998; Wu et al, 1999

Chen and Florian: *Transportation Research Part B*, 32(3), 1998; Wu et al: *Transportation Science*, 33(4) 1999.

1. Assignment: network loading only based on Friesz et al, 1993
2. Arc dynamics: ditto (Ran et al, 1993)
3. Delay: ditto (PQM)
4. Computable: yes, (i) finite element method for finding exit-time functions, (ii) heuristic
5. Identifies the network loading phase as a subproblem representing the greatest computational challenge

DTA History, Paper 7: Zhu and Marcotte, 2000

Zhu and Marcotte: *Transportation Science*, 34(4), 2000.

1. Assignment: SRD (Friesz et al, 1993)
2. Dynamics: ditto (Ran et al, 1993)
3. Delay: ditto (PQM)
4. Computable: NA
5. Proves existence of SRD-PQM DUE solutions
6. Assumed bounds on variables
7. No proof yet available without a priori bounds

DTA History, Paper 8: Friesz et al, 2001

Friesz et al: *Networks & Spatial Economics*, 1(3), 2001.

1. Assignment: SRD DUE differential variational inequality
2. Dynamics: system of algebraic differential equations (DAEs)
3. Delays: found from DAE system endogenously
4. Computable: yes, but later
5. Network loading phase not treated separately
6. Analyzes necessary conditions with time lags

DTA History, Paper 9: Lo and Szeto, 2010

Lo and Szeto: *Transportation Research Part B*, 36(5), 2002.

1. Assignment: SRD dynamic user equilibrium, discrete time
2. Dynamics: cell-transmission model (CTM), a spatially and temporally discrete approximation of the Lighthill-Whitham-Richards (LWR) PDE of hydrodynamic traffic flow theory
3. Delay: from CTM (discrete LWR PDE)
4. Computable: yes, alternating direction method for coercive VIs

DTA History, Paper 10: Ban et al, 2008

Ban, Ferris, Ran: *Transportation Research Part B*, 42(9), 2008.

1. Assignment: SRD MiCP formulation
2. Dynamics: controlled entering and exiting flow (Ran et al, 1993)
3. Delays: from flow propagation constraints (Astarita, 1996)
4. Computable: NCP relaxation of the MiCP, successive linearization/Lemke's LCP algo

DTA History, Paper 11: Perakis and Roels, 2006

Perakis and Roels: *Operations Research*, 54(6), 2006.

1. Assignment: SRD
2. Dynamics: intrinsic to the delays
3. Delays: Extracts arc delay operators from LWR PDE (hydrodynamic traffic flow theory)
4. Computable: yes, Frank-Wolfe extension

DTA History, Paper 12: Kachani and Perakis, 2010

Kachani and Perakis: *Networks & Spatial Economics*, 10(2), 2010

1. Presents a differential algebraic system based on LWR PDE (hydrodynamic traffic flow theory) for finding path delay
2. Continuous-time equivalent of the cell transmission model
3. Elegant; a potentially very influential work

DTA History, Paper 13: Nie and Zhang 2010

Nie and Zhang: *Networks & Spatial Economics*, 10(2), 2010.

1. Assignment: SRD
2. Dynamics: (i) PQM, (ii) ad hoc, (iii) CTM
3. Delays: from DAEs based on dynamics
4. Computable: yes; comparative numerical tests using various delay models
5. Algorithms considered: gap-functions, MSA, projected gradient, alternating direction

DTA History, Paper 14: Friesz and Mookherjee, 2006

Friesz and Mookherjee: *Transportation Research Part B*, 40(3), 2006.

1. Assignment: Friesz et al, 2001, SRD differential variational inequality
2. Dynamics: Ran et al, 1993
3. Delays: endogenous, from the point queue model
4. Computable: yes; computation in continuous time; fixed point with projection

Overview of the Remainder of This Talk

1. Part I: Static Nash Games and Static User Equilibrium. We review some well-known results:
   1. basic facts about noncooperative game-theoretic equlibria
   2. Nash-like traffic equilibria in a static setting
   3. finite dimensional variational inequalities, complementarity problems, fixed points, and equivalent nonlinear programs.

2. Part II: Dynamic User Equilibrium

    1. We define the notion of a moving Nash equilibium.
    2. We state one version the dynamic network user equilibrium (DUE) problem.
    3. We discuss the challenges and pitfalls of the DUE problem.
    4. We present an algorithm for the DUE problem involving computation in continuous time.
    5. We present the dynamic equilibrium network design problem.
    6. We present one version of the dynamic mechanism design problem.
    7. Time permitting, we may introduce a naive model of the Internet.

Overview (continued)

3. Part III: Equilibrium Network Design

   1. We introduce the fundamental problem of traffic network equilibrium design and note it is a specific instance of mechanism design.
   2. We introduce the Braess paradox and comment briefly on the price of anarchy.
   3. We discuss some fundamental solution perspectives for different versions of network design.
   4. We generalize and extend the aforementioned results to a dynamic setting and note the distinction between dynamic equilibrium design and dynamic disquilibrium design.

Overview (continued)

4. Part IV: Dynamic Mechanism Design and Dynamic Congestion Pricing
   1. Dynamic mechanisms are potentiall difficult to design because of stability issues.
   2. We consider 2nd best dynamic congestion pricing.
   3. Dynamic Stackelberg games are the central focus.
   4. Algorithms are challenging and various forms of computational intelligence become necessary.

Part I. Static Nash Games and Static User Equilibrium

Definition (Nash Equilibrium)

Nash equilibrium. Suppose N agents, each playing the feasible strategy vector x^i from the strategy set Λ_i which is independent of the other players' strategies. Furthermore, every agent $i \in [1, N] \subset \mathcal{I}_{++}$ has a cost (disutility) function $\Theta_i(x) : \Lambda \longrightarrow \Re^1$ that depends on all agents' strategies where

$$\Lambda = \prod_{i=1}^{N} \Lambda_i$$

and $x = \left(x^i : i = 1, \ldots, N\right)$. Every agent $i \in [1, N]$ seeks to solve the problem

$$\min \Theta_i(x^i, x^{-i}) \quad \text{s.t.} \quad x^i \in \Lambda_i \subseteq \Re^{n_i} \tag{1}$$

for each fixed yet arbitrary non-own tuple $x^{-i} = \left(x^j : j \neq i\right)$. A Nash equilibrium $NE(\Theta, \Lambda)$ is a tuple of strategies x such that each x^i solves mathematical program (1).

Definition (Generalized Nash Equilibrium)

Generalized Nash equilibrium. Suppose N agents, each playing the feasible strategy vector x^i from the strategy set $\Lambda_i(x)$ that depends on all agents' strategies where $x = \left(x^i : i = 1, \ldots, N\right)$. Furthermore, every agent $i \in [1, N] \subseteq \mathcal{I}_{++}$ has a cost (disutility) function $\Theta_i(x) : \Lambda(x) \longrightarrow \Re^1$ that depends on all agents' strategies where

$$\Lambda(x) = \prod_{i=1}^{N} \Lambda_i(x)$$

Every agent $i \in [1, N]$ seeks to solve the problem

$$\min \Theta_i(x^i, x^{-i}) \quad \text{s.t.} \quad x^i \in \Lambda_i(x) \tag{2}$$

for each fixed yet arbitrary non-own tuple $x^{-i} = \left(x^j : j \neq i\right)$. A generalized Nash equilibrium $GNE(\Theta, \Lambda)$ is a tuple of strategies x such that each x^i solves mathematical program (2).

Note the Essential Difference:

$$\text{Nash} \quad : \quad x^i \in \Lambda_i$$
$$\text{Generalized Nash} \quad : \quad x^i \in \Lambda_i(x)$$

Traffic Equilibrium: Two Behavioral Principles

- Wardrop's First Principle (WFP) for Route Choice. Users of the transport network noncooperatively select routes to minimize their individual generalized costs of transportation. (Nash-like network game, also referred to as a "congestion game")
  - Note that flows obeying WFP are said to be *user optimal* or to constitute a *user equilibrium flow* pattern.
- Wardrop's Second Principle (WSP) for Route Choice. Users of the transport network cooperatively select routes to minimize system-wide, total transport costs. (minimum cost flow problem with path variables)
  - Note that flows obeying WSP are said to be *system optimal* or to constitute a *system optimized flow* pattern.

Notation, Indices and Sets

p	denotes a path
a	denotes an arc
i, j, k, l	denote nodes
$\mathcal{W}$	set of network origin-destination (OD) pairs (i, j)
$\mathcal{A}$	set of directed network arcs
$\mathcal{N}$	set of network nodes
$\mathcal{N}_O$	set of origin nodes
$\mathcal{N}_D$	set of destination nodes
$\mathcal{P}_{ij}$	set of paths connecting origin-destination pair $(i, j) \in W$
$\mathcal{P}$	set of all paths

Notation, Variables

$$
\begin{aligned}
f_a &= \text{the flow on arc } a \\
f &= (\ldots, f_a, \ldots)^T \in \Re^{|\mathcal{A}|} \\
h_p &= \text{the flow on path } p \\
h &= (\ldots, h_p, \ldots)^T \in \Re^{|\mathcal{P}|} \\
u_{ij} &= \text{minimum cost of transportation b'twn OD pair } (i, j) \\
u &= (\ldots, u_{ij}, \ldots)^T \in \Re^{|\mathcal{W}|} \\
Q_{ij} &= \text{demand for travel between origin } i \text{ and destination } j \\
Q &= (\ldots, Q_{ij}, \ldots)^T \in \Re^{|\mathcal{W}|}
\end{aligned}
$$

Notation, Functions

$$
\begin{aligned}
c_a(f) : \Re^{|\mathcal{A}|} \longrightarrow \Re^1 &= \text{unit cost (latency) of flow on arc } a \\
c(f) : \Re^{|\mathcal{A}|} \longrightarrow \Re^{|\mathcal{A}|} &= \text{unit arc cost (latency) vector}
\end{aligned}
$$

$$
\begin{aligned}
C_p(h) : \Re^{|\mathcal{P}|} \longrightarrow \Re^1 &= \text{unit cost (latency) of flow on path } p \in \mathcal{P} \\
C(h) : \Re^{|\mathcal{P}|} \longrightarrow \Re^{|\mathcal{P}|} &= \text{unit path cost (latency) vector}
\end{aligned}
$$

$$
\begin{aligned}
Q_{ij}(u) : \Re^{|\mathcal{W}|} \longrightarrow \Re^1 &= \text{travel demand function for } (i,j) \in \mathcal{W} \\
Q(u) : \Re^{|\mathcal{W}|} \longrightarrow \Re^{|\mathcal{W}|} &= \text{vector travel demand function}
\end{aligned}
$$

$$
\begin{aligned}
\theta_{ij}(Q) : \Re^{|\mathcal{W}|} \longrightarrow \Re^1 &= \text{inverse travel demand function for } (i,j) \in \mathcal{W} \\
\theta(Q) : \Re^{|\mathcal{W}|} \longrightarrow \Re^{|\mathcal{W}|} &= \text{inverse vector travel demand function}
\end{aligned}
$$

Why Nonseparable Functions? Slide 1

1. Why use $c_a(f)$ instead of $c_a(f_a)$?
2. The short answer is: to model inter-arc and intra-arc flow interactions.
3. An example of inter-arc flow interactions is the intersection of two streets (at a node).
4. An example of intra-arc interations is bus traffic and automobile traffic using the same street (arc).
5. For very similar reasons, we use nonseparable demand and nonsepable inverse demand functions to model multiple modes, multiple user classes, and interdependent trip purposes.
6. More generally, nonseparable cost and demand functions arise when we use a *multi-copy formulation* of the network of interest.

Multicopy Notation

1. copies of arcs
2. copies of nodes:
3. links carry mode, class, interaction, real physical identity

Notation, Matrices

$$
\begin{aligned}
\delta_{ap} &= \text{element of arc-path incidence matrix} \\
\Delta &= (\delta_{ap}), \text{ the arc-path incidence matrix} \\
\gamma_{ij}^{p} &= \text{element of the OD-path incidence matrix} \\
\Gamma &= \left(\gamma_{ij}^{p}\right), \text{ the OD-path incidence matrix}
\end{aligned}
$$

Fundamental Identities

1. Path costs are additive in arc costs:

$$C = \Delta c \quad \text{or} \quad C_p = \sum_{a \in \mathcal{A}} \delta_{ap} c_a \qquad \forall p \in \mathcal{P}$$

2. Arc flows are additive in path flows:

$$f = \Delta h \quad \text{or} \quad f_a = \sum_{a \in \mathcal{A}} \delta_{ap} h_p \qquad \forall a \in \mathcal{A}$$

3. travel demand is additive in path flows (market clearing or flow conservation constraints):

$$\Gamma h = Q \quad \text{or} \quad \sum_{p \in \mathcal{P}_{ij}} h_p = Q_{ij} \qquad \forall (i, j) \in \mathcal{W}$$

Feasible Solutions

$$
\begin{aligned}
\Omega &= \left\{ \begin{pmatrix} h \\ u \end{pmatrix} : \Gamma h - Q(u) = 0, h \geq 0, u \geq 0 \right\} \text{ elastic demand} \\
\Omega_0 &= \left\{ h : \Gamma h = Q, h \geq 0 \right\} \text{ fixed demand} \\
\Omega_1 &= \left\{ f : f = \Delta h, \Gamma h = Q, h \geq 0 \right\} \text{ fixed demand} \\
\Omega_2 &= \left\{ \begin{pmatrix} f \\ Q \end{pmatrix} : f = \Delta h, \Gamma h - Q = 0, h \geq 0 \right\} \text{ elastic, invertible demand} \\
\Omega_3 &\equiv \left\{ \begin{pmatrix} f \\ u \end{pmatrix} : f = \Delta h, \Gamma h - Q(u) = 0, h \geq 0, u \geq 0 \right\}
\end{aligned}
$$

The model of interest is

$$h_p > 0, p \in P_{ij} \Longrightarrow c_p = u_{ij} \quad \forall (i,j) \in \mathcal{W}, p \in \mathcal{P}_{ij} \tag{3}$$

$$\sum_{p \in P_{ij}} h_p = Q_{ij}(u) \quad \forall (i,j) \in \mathcal{W} \tag{4}$$

$$h_p \geq 0 \quad \forall (i,j) \in \mathcal{W}, p \in \mathcal{P}_{ij} \tag{5}$$

Note: (3), (4) and (5) above assure that

$$c_p > u_{ij},\ p \in \mathcal{P}_{ij} \Longrightarrow h_p = 0$$

User Equilibrium Restated

The problem of interest $UE(C,\Omega)$:

$$
\begin{aligned}
\left[c_p(h) - u_{ij}\right] h_p &= 0 \quad (i,j) \in \mathcal{W}, p \in \mathcal{P}_{ij} \\
c_p(h) - u_{ij} &\geq 0 \quad (i,j) \in \mathcal{W}, p \in \mathcal{P}_{ij} \\
Q_{ij}(u) - \sum_{p \in P_{ij}} h_p &= 0 \ \forall (i,j) \in \mathcal{W} \\
h_p &\geq 0 \ \forall (i,j) \in \mathcal{W}, p \in \mathcal{P}_{ij}
\end{aligned}
$$

User Equilibrium as A Nonlinear Complementarity Problem, Slide 1

Under the assumption of cost positivity, that is

$$c_a(f) > 0 \quad \forall a \in \mathcal{A} \quad \text{and} \quad f \in \Omega_j \quad j = 1, 2, 3$$

the user equilibrium problem $UE(C, \Omega)$ is equivalent to a nonlinear complementarity problem:

$$\begin{aligned} \left[F(x)\right]^T x &= 0 \\ F(x) &\geq 0 \\ x &\geq 0 \end{aligned}$$

which is also sometimes written as

$$0 \leq F(x) \perp x \geq 0$$

User Equilibrium as A Nonlinear Complementarity Problem, Slide 2

Under cost positivity, $\Gamma h = Q$ may be relaxed so that

$$
\begin{aligned}
\left[c_p\left(h\right) - u_{ij}\right] h_p &= 0 \quad \forall \left(i,j\right) \in \mathcal{W}, p \in \mathcal{P}_{ij} \\
c_p\left(h\right) - u_{ij} &\geq 0 \quad \forall \left(i,j\right) \in \mathcal{W}, p \in \mathcal{P}_{ij} \\
h_p &\geq 0 \quad \forall \left(i,j\right) \in \mathcal{W}, p \in \mathcal{P}_{ij} \\
\left[\sum_{p \in P_{ij}} h_p - Q_{ij}(u)\right] u_{ij} &= 0 \quad \forall \left(i,j\right) \in \mathcal{W} \\
\sum_{p \in P_{ij}} h_p - Q_{ij}(u) &\geq 0 \\
u_{ij} &\geq 0 \quad \forall \left(i,j\right) \in \mathcal{W}
\end{aligned}
$$

is equivalent to $UE(C, \Omega)$.

User Equilibrium as A Nonlinear Complementarity Problem, Slide 3

The key observation establishing equivalency is:

$$\text{assume } \sum_{p \in P_{ij}} h_p - Q_{ij}(u) > 0 \Longrightarrow \exists \ h_q > 0 \Longrightarrow u_{ij} = 0$$

$$\text{however } h_q > 0 \Longrightarrow C_p(h) - u_{ij} = 0 \Longrightarrow C_p(h) = 0$$

$$\text{arc cost positivity} \Longrightarrow C_p(h) > 0$$

$$\text{contradiction!!} \Longrightarrow \sum_{p \in P_{ij}} h_p - Q_{ij}(u) = 0$$

Alternative Formulations of the UE Problem

1. nonlinear complementarity problem
2. fixed point problem
3. variational inequality problem
4. Kuhn-Tucker conditions for VIs
5. mathematical program based on gradient mapping
6. mathematical program based on gap function
7. constraint programming based on nonlinear constraints

Let Us Now Introduce the Fixed Demand Assumption

Now take $Q_{ij} \in \Re_{++}$ to be an exogenously determined constant

Therefore Q is a fixed $|\mathcal{N}_O| \times |\mathcal{N}_D|$ matrix

Let us, for the time being, take Q to be a fixed matrix of travel demands, generally called the trip matix. Consider

$$h = P_{\Omega_0}\left[h - \alpha C\left(h\right)\right] \tag{6}$$

where

$$\begin{aligned} \Omega_0 &= \left\{h : \Gamma h - Q = 0, h \geq 0\right\} && (7) \\ P_{\Omega_0}\left(v\right) &= \arg\min_{y} \|v - y\| \text{ s.t. } y \in \Omega_0 && (8) \\ \alpha &\in \Re^1_{++} && (9) \end{aligned}$$

and $P_{\Omega_0}\left(v\right)$ is the minimum norm projection of the fixed vector v onto Ω_0.

Fixed Point Formulation of the UE Problem, Slide 2

Note that

$$\min_{y} \|v - y\| \ \text{ s.t. } \ y \in \Omega_0$$

is equivalent to

$$\min \frac{1}{2}(v-y)^T(v-y) = \frac{1}{2}(v-y)^2 \ \text{ s.t. } \ y \in \Omega_0$$

a convex program whose Kuhn-Tucker identity is

$$(-1)(v-y) + u^T\Gamma - \rho = 0$$

where complementary slackness holds:

$$\begin{aligned} \rho^T y &= 0 \\ \rho &\geq 0 \end{aligned}$$

Thus, upon recalling that $y = \arg\min_{y \in \Omega_0} \|v - y\|$, the Kuhn-Tucker conditions yield

$$v - u^T\Gamma = \rho \geq 0$$

Taking

$$v = \alpha C$$

we have

$$\alpha c_p(h) - u_{ij} = \rho_p \quad \forall (i,j) \in \mathcal{W}, p \in \mathcal{P}_{ij}$$

Taking $\alpha = 1$, without loss of generality, we have by complementary slackness that

$$[c_p(h) - u_{ij}] h_p = \rho_p h_p = 0 \quad \forall (i,j) \in \mathcal{W}, p \in \mathcal{P}_{ij}$$

which is recognized as user equilibrium. Thus, since the KT conditions are both necessary and suffiecient, (6) and $UE(C, \Omega_0)$ are equivalent.

UE Existence

Recall

Theorem

Brouwer's Fixed-Point Theorem. If S is a convex, compact set and $F(x)$ is continuous on S, then

$$x = F(x)$$

has a solution.

The fixed-point formulation allows us to apply Brouwer's theorem directly with the following result:

Theorem

Existence of a user equilibrium. If $C(h)$ is continuous on Ω_0 and the travel demand $Q < +\infty$, then a user equilibrium $UE(C, \Omega_0)$ exists.

Abstract Variational Inequality Problem Defined

Definition

Variational inequality. Given a nonempty set, $\Lambda \subseteq \Re^n$, and a function, $F : \Lambda \longrightarrow \Re^n$, the variational inequality problem $VI(F, \Lambda)$ is to find a vector y such that

$$\left.\begin{array}{c} y \in \Lambda \\ \left[F(y)\right]^T (x - y) \geq 0 \quad \forall\, x \in \Lambda \end{array}\right\} \quad VI(F, \Lambda) \tag{10}$$

Abstract Quasivariational Inequality Problem Defined

It is possible to generalize $VI(F, \Lambda)$ in a variety of ways. One generalization of importance to us is the following:

Definition

Quasivariational inequality. Given a nonempty set $\Lambda \subseteq \Re^n$ and a function $F : \Re^n \longrightarrow \Re^n$, let Λ be a point-to-set mapping from $\Re^n$ to subsets of $\Re^n$. The *quasivariational inequality problem* $QVI\,(F, \Lambda)$ is to find a vector $y \in \Lambda\,(y)$ such that

$$\left.\begin{array}{cc} y \in \Lambda(y) & \\ \left[F\,(y)\right]^T (x - y) \geq 0, & \forall x \in \Lambda\,(y) \end{array}\right\} \quad QVI\,(F, \Lambda)$$

Note the Essential Difference:

$$
\begin{aligned}
VI &: \quad x \in \Lambda \\
QVI &: \quad x \in \Lambda(x)
\end{aligned}
$$

Variational Inequality Formulation of the UE Problem, Slide 1

Theorem

User equilibrium as a variational inequality. The flow pattern $h^* = \left(h_p^* : p \in \mathcal{P}\right)$ *is a user equilibrium if and only if*

$$\left.\begin{array}{c} h^* \in \Omega_0 \\ \sum_{p \in \mathcal{P}} C_p\left(h^*\right)\left(h_p - h_p^*\right) \geq 0 \qquad \forall h \in \Omega_0 \end{array}\right\} \quad VI\left(c, \Omega_0\right) \tag{11}$$

where we recall that

$$\Omega_0 = \left\{ h \geq 0 : \sum_{p \in \mathcal{P}_{ij}} h_p = Q_{ij} \quad \forall \left(i, j\right) \in \mathcal{W} \right\} \tag{12}$$

where we have taken the trip table Q to be a fixed matrix.

Variational Inequality Formulation of the UE Problem, Slide 2

Proof: The proof that *UE* and *VI* are equivalent consists of two parts and is largely algebraic:

(Part 1) [$UE \Longrightarrow VI$] Note that

$$C_p\left(h^*\right) \geq u_{ij}$$

for any $p \in \mathcal{P}_{ij}$. so that

$$C_p\left(h^*\right)\left(h_p - h_p^*\right) \geq u_{ij}\left(h_p - h_p^*\right) \tag{13}$$

including the case of $\left(h_p - h_p^*\right) < 0$, for then

$$h_p^* > h_p \geq 0 \Longrightarrow C_p(h^*) = u_{ij}$$

Therefore, from (13), upon summing over paths, we have at once the variational inequality (11).

Variational Inequality Formulation of the UE Problem, Slide 3

(Part 2) $[VI \implies UE]$ Note that VI (11) is equivalent to

$$\sum_{(i,j)\in\mathcal{W}} \sum_{p\in\mathcal{P}_{ij}} C_p\left(h^*\right) h_p \geq \sum_{(i,j)\in\mathcal{W}} \sum_{p\in\mathcal{P}_{ij}} C_p\left(h^*\right) h_p^* \quad \forall h \in \Omega_0$$

Now consider Kuhn-Tucker conditions (KTC) for the equivalent <u>linear</u> program:

$$\min \sum_{(i,j)\in\mathcal{W}} \sum_{p\in\mathcal{P}_{ij}} C_p\left(h^*\right) h_p \quad \text{s.t.} \quad h \in \Omega_0$$

The KTC are equivalent to a user equilibrium:

$$\begin{aligned} C_p\left(h^*\right) - u_{ij} - \rho_p &= 0 \qquad \forall\left(i,j\right) \in \mathcal{W}, p \in \mathcal{P}_{ij} \\ \rho_p h_p &= 0 \qquad \forall\left(i,j\right) \in \mathcal{W}, p \in \mathcal{P}_{ij} \\ \rho_p &\geq 0 \qquad \forall\left(i,j\right) \in \mathcal{W}, p \in \mathcal{P}_{ij} \end{aligned}$$

$\blacksquare$

The Infamous Equivalent Mathematical Program, Slide 1

$$F(x^*) = \min_{x \in \Lambda} F(x)$$

$$\Updownarrow \tag{14}$$

$$\left[\nabla F(x^*)\right]^T (x - x^*) \geq 0 \quad x, x^* \in \Lambda$$

Further note that

$$F(x) = \nabla G(x) \quad \text{and} \quad \nabla G(x)$$

Note that any solution $x^* \in \Lambda$ of

$$G(x^*) = \min \oint_{0 \in \Re^n}^{x \in \Re^n} [F(y)]^T dy \quad \text{s.t.} \quad x \in \Lambda \tag{15}$$

is a solution of

$$[F(x^*)]^T (x - x^*) \geq 0 \tag{16}$$

where $\oint$ denotes a line integral. In the event that $F(x)$ is monotone on Λ and $G(x)$ is well defined, then $G(x)$ is convex on Λ, and (16) is both a necessary and sufficient condition.

For fixed travel demand UE, the equivalent mathematical program is

$$\min \sum_{a \in \mathcal{A}} \oint_0^f c_a(x)\, dx_a$$

subject to

$$f \in \Omega_1 \equiv \{f : f = \Delta h, \Gamma h = Q, h \geq 0\}$$

The Infamous Equivalent Mathematical Program, Slide 4

1. There is a very problematic feature of the equivalent mathematical program
2. In particular, by Green's theorem in a plane, for the line integral

$$\sum_{a \in \mathcal{A}} \oint_0^f c_a(x)\, dx_a$$

to be single-valued (independent of the path of integration), the Jacobian of $c : \Re^{|\mathcal{A}|} \longrightarrow \Re^{|\mathcal{A}|}$ must be symmetric.

3. That is

$$\frac{\partial c_a(f)}{\partial f_b} = \frac{\partial c_b(f)}{\partial f_a} \quad \forall a, b \in \mathcal{A}$$

4. Meaning?
5. Implications?

The Fixed vs. Elastic Demand Assumptions

The preceding discussion of the variational inequality formulation of the user equilibrium problem and the equivalent mathematical program was based on the assumption of fixed travel demand. That assumption may be relaxed, and we now cite the relevant results without proof. (The proofs are easy.)

The VI For UE With Elastic, Invertible Demand

1. Define

$$\Omega_2 = \left\{ \begin{pmatrix} f \\ Q \end{pmatrix} : f = \Delta h, \Gamma h - Q = 0, h \geq 0, Q \geq 0 \right\}$$

2. We denote the equivalent variational inequality if demand functions are invertible by $VI\left(c, \theta, \Omega_2\right)$; its statement is:

$$\text{find } \begin{pmatrix} f^* \\ Q^* \end{pmatrix} \in \Omega_2$$

such that

$$\sum_{a\in\mathcal{A}} c_a\left(f^*\right)\left(f_a - f_a^*\right) - \sum_{(i,j)\in\mathcal{W}} \theta_{ij}\left(Q^*\right)\left(Q_{ij} - Q_{ij}^*\right) \geq 0$$

$$\forall \begin{pmatrix} f \\ Q \end{pmatrix} \in \Omega_2$$

Existence with Elastic Demand, Slide 1

Recall

$$\sum_{a \in \mathcal{A}} c_a(f^*)(f_a - f_a^*) - \sum_{(i,j) \in \mathcal{W}} \theta_{ij}(Q^*)\left(Q_{ij} - Q_{ij}^*\right) \geq 0$$

$$\begin{pmatrix} f \\ Q \end{pmatrix}, \begin{pmatrix} f^* \\ Q^* \end{pmatrix} \in \Omega_2$$

The fixed-point equivalent of this problem is

$$\begin{pmatrix} h \\ Q \end{pmatrix} = P_{\Omega_2}\left[\begin{pmatrix} h \\ Q \end{pmatrix} - \alpha \begin{pmatrix} C(h) \\ -\theta(Q) \end{pmatrix}\right]$$

Existence with Elastic Demand, Slide 2

Consider

$$\begin{pmatrix} h \\ Q \end{pmatrix} = P_{\Omega_2}\left[\begin{pmatrix} h \\ Q \end{pmatrix} - \alpha \begin{pmatrix} C(h) \\ -\theta(Q) \end{pmatrix}\right]$$

Let us

1. observe Ω_2 is closed;
2. assume demands are bounded from above, so that Ω_2 is compact; and
3. assume costs and inverse demands are continuous.

Then the famous Brouwer fixed point theorem assures existence.

The Equivalent Mathematical Program With Invertible Demand, Statement

$$\min \sum_{a \in \mathcal{A}} \oint_0^f c_a(x)\, dx_a - \sum_{(i,j) \in \mathcal{W}} \oint_0^Q \theta_{ij}(y)\, dy_{ij}$$

subject to

$$\begin{pmatrix} f \\ Q \end{pmatrix} \in \Omega_3$$

where

$$\Omega_3 \equiv \left\{ \begin{pmatrix} f \\ Q \end{pmatrix} : f = \Delta h, \Gamma h - Q = 0, h \geq 0, Q \geq 0 \right\}$$

The Equivalent Mathematical Program with Invertible Demand, Issues

1. Now we have two line integrals for a general multicopy network. Thus the Jacobians $J(c)$ and $J(\theta)$ must be symmetric for the objective to be well defined. Symmetric $J(\theta)$ means the cross elasticies of demand for modes of travel are identical, which is not plausible.
2. If the vector cost function is monotonically increasing and the vector inverse demand function is monotonically decreasing, we will have a convex program.
3. If monotonically increasing (decreasing) is strengthened to strictly monotonically increasing (decreasing) in item 3 above, we assure any solution is unique.
4. What if demand is not invertible?

The Equivalent Mathematical Program When Demand is not Invertible

We may use integration by parts, that is

$$\int w dv = w \cdot v - \int v dw,$$

to re-state the equivalent program as

$$\min \sum_{a \in \mathcal{A}} \oint_0^f c_a(x)\, dx_a - \sum_{(i,j) \in \mathcal{W}} u_{ij} \cdot Q_{ij}(u) + \sum_{(i,j) \in \mathcal{W}} \oint_0^u Q_{ij}(z)\, dz_{ij}$$

subject to

$$\begin{pmatrix} f \\ u \end{pmatrix} \in \Omega_3$$

where

$$\Omega_3 \equiv \left\{ \begin{pmatrix} f \\ u \end{pmatrix} : f = \Delta h, \Gamma h - Q(u) = 0, h \geq 0, u \geq 0 \right\}$$

Gap Function for VIs Defined

Formally, we define a gap function for $VI(F, \Lambda)$ as follows:

Definition

Gap function. A function $\zeta : \Lambda \subseteq \Re^n \longrightarrow \Re_+$ is called a gap function for $VI(F, \Lambda)$ when the following statements hold:

1. $\zeta(y) \geq 0$ for all $y \in \Lambda$
2. $\zeta(y) = 0$ if and only if y is a solution of $VI(F, \Lambda)$

Clearly, a gap function with the properties of Definition 8 allows us to re-formulate $VI(F, \Lambda)$ as an optimization problem, namely as

$$\min_{y \in \Lambda} \zeta(y) \tag{17}$$

An optimal solution of (17) solves $VI(F, \Lambda)$ provided $\zeta(y)$ may be driven to zero.

It is reasonable to ask wether there is a gap function that leads to an equivalent unconstrained mathematical program. In fact, the so-called *D-gap function* proposed by Peng (1997) and generalized by Yamashita et al (1997) is such a function. A D-gap function is the difference between two gap functions. The D-gap function we will consider is

$$\begin{aligned} \psi_{\alpha\beta}(y) &= \max_{x\in\Lambda}\{\langle F(y), y-x\rangle - \alpha\phi(y,x)\} \\ &\quad - \max_{x\in\Lambda}\{\langle F(y), y-x\rangle - \beta\phi(y,x)\} \end{aligned} \tag{18}$$

where $0 < \alpha < \beta$ and certain other technical conditions hold.

Mathematical Program Based on the D-Gap Function

The corresponding unconstrained mathematical program equivalent to $VI(F, \Lambda)$ is

$$\min_{y} \psi_{\alpha\beta}(y) \tag{19}$$

Moreover, the gradient of $\psi_{\alpha\beta}(y)$ can be shown to be

$$\begin{aligned} \nabla\psi_{\alpha\beta}(y) &= \nabla F(y)\left(x_\beta(y) - x_\alpha(y)\right) + \beta\nabla_y\phi\left(y, x_\beta(y)\right) \\ &\quad - \alpha\nabla_y\phi\left(y, x_\alpha(y)\right) \end{aligned}$$

For detailed proofs of the assertions we have made concerning $\psi_{\alpha\beta}(y)$ see Yamashita et al (1997).

Step 0. Initialization. Initialization. Determine an initial feasible solution $y^0 \in \Re^n$ set $k = 0$.

Step 1. Finding the steepest descent direction. Find the gradient of the the D-gap function:

$$\begin{aligned}\nabla \psi_{\alpha\beta}\left(y^k\right) &= \nabla F\left(y^k\right)\left(x_\beta\left(y^k\right) - x_\alpha\left(y^k\right)\right) + \beta \nabla_y \phi\left(y^k, x_\beta\left(y^k\right)\right) \\ &\quad - \alpha \nabla_y \phi\left(y^k, x_\alpha\left(y^k\right)\right)\end{aligned}$$

where

$$x_\alpha\left(y^k\right) = \arg\max_{x\in\Lambda}\left\{\left\langle F\left(y^k\right), y^k - x\right\rangle - \alpha\phi\left(y^k, x\right)\right\}$$

$$x_\beta\left(y^k\right) = \arg\max_{x\in\Lambda}\left\{\left\langle F\left(y^k\right), y^k - x\right\rangle - \beta\phi\left(y^k, x\right)\right\}$$

Then find

$$d^k = \arg\min\left\{\left[-\nabla\psi_{\alpha\beta}\left(y^k\right)\right]^T y \quad \text{s.t.} \quad \|y\| \le 1\right\}$$

Step 2. Step size determination. Find

$$\theta_k = \arg\min \left\{ \psi_{\alpha\beta} \left(y^k + \theta d^k \right) \quad \text{s.t.} \quad 0 \leq \theta \leq 1 \right\} \tag{20}$$

or employ a suitably small constant step size.

Step 2. Stopping test and updating. For $\varepsilon \in \Re^1_{++}$, a preset tolerance, if

$$\left\| \nabla \psi_{\alpha\beta} \left(y^k \right) \right\| < \varepsilon,$$

stop; otherwise set

$$y^{k+1} = y^k - \theta_k d^k \left(y^k \right)$$

and go to Step 1 with k replaced by $k+1$.

Example Based on the D-Gap Function, Slide 1

For a numerical example, consider $VI\left(F, \Lambda\right)$ where

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

$$F\left(x\right) = \begin{pmatrix} F_1\left(x_1, x_2\right) \\ F_2\left(x_1, x_2\right) \end{pmatrix} = \begin{pmatrix} x_1 - 5 \\ 0.1x_1x_2 + x_2 - 5 \end{pmatrix}$$

$$\Lambda = \{\left(x_1, x_2\right) : x_1 \geq 0, x_2 \geq 0, x_1 + x_2 - 1 \leq 0\}$$

In this example, we employ a D-gap function of the form

$$\psi_{\alpha\beta}\left(y\right) = \zeta_\alpha\left(y\right) - \zeta_\beta\left(y\right)$$

where

$$\zeta_\alpha\left(y\right) = \max_{x \in \Lambda}\left\{\left\langle F\left(y\right), y - x\right\rangle - \frac{\alpha}{2}\left\|y - x\right\|^2\right\}$$

$$\zeta_\beta\left(y\right) = \max_{x \in \Lambda}\left\{\left\langle F\left(y\right), y - x\right\rangle - \frac{\beta}{2}\left\|y - x\right\|^2\right\}$$

and $0 < \alpha < \beta$.

Example Based on the D-Gap Function, Slide 2

After some effort, the gradient is found to be

$$\nabla\psi_{\alpha\beta}(y) = \nabla F(y)\left(x_\beta(y) - x_\alpha(y)\right) + \beta\nabla_y\phi\left(y, x_\beta(y)\right) - \alpha\nabla_y\phi\left(y, x_\alpha(y)\right)$$

$$= \begin{pmatrix} 1 & 0 \\ 0.1y_2 & 0.1y_1 + 1 \end{pmatrix} \begin{pmatrix} x_{\beta 1}(y) - x_{\alpha 1}(y) \\ x_{\beta 2}(y) - x_{\alpha 2}(y) \end{pmatrix} + \beta \begin{pmatrix} y_1 - x_{\beta 1}(y) \\ y_2 - x_{\beta 2}(y) \end{pmatrix} - \alpha \begin{pmatrix} y_1 - x_{\alpha 1}(y) \\ y_2 - x_{\alpha 2}(y) \end{pmatrix}$$

where

$$x_\alpha(y) = \arg\max_{x\in\Lambda}\left\{(y_1 - 5)(y_1 - x_1) + (0.1y_1y_2 + y_2 - 5)(y_2 - x_2) - A\right\}$$

$$x_\beta(y) = \arg\max_{x\in\Lambda}\left\{(y_1 - 5)(y_1 - x_1) + (0.1y_1y_2 + y_2 - 5)(y_2 - x_2) - B\right\}$$

$$A = \frac{1}{2}\alpha\left((x_1 - y_1)^2 + (x_2 - y_2)^2\right) \quad \ldots\ldots\ldots\ldots$$

Example Based on the D-Gap Function, Slide 3

If we employ the constant step size $\theta_k = 0.5$, the following table of results is generated:

k	gap $\psi_{\alpha\beta}(y^k)$	y^k	$x_\alpha(y^k)$	$x_\beta(y^k)$
0	0.375	$(0, 0)$	$(0.500, 0.500)$	$(0.50, 0.500)$
1	2.3512×10^{-2}	$(0.375, 0.375)$	$(0.514, 0.486)$	$(0.503, 0.496)$
2	1.0139×10^{-4}	$(0.475, 0.463)$	$(0.517, 0.483)$	$(0.508, 0.492)$
3	7.005×10^{-6}	$(0.508, 0.483)$	$(0.515, 0.485)$	$(0.511, 0.489)$
4	4.9345×10^{-7}	$(0.509, 0.487)$	$(0.513, 0.487)$	$(0.512, 0.488)$
5	3.5878×10^{-8}	$(0.512, 0.487)$	$(0.512, 0.487)$	$(0.512, 0.488)$
6	2.7672×10^{-9}	$(0.512, 0.487)$	$(0.513, 0.487)$	$(0.512, 0.488)$
7	1.1008×10^{-10}	$(0.512, 0.487)$	$(0.512, 0.488)$	$(0.512, 0.488)$

Evidently, the algorithm terminates with gap $< 10^{-9}$ and approximate solution $y = (0.512, 0.488)$.

Part II. Dynamic User Equilibrium (DUE)

DUE with Exogenous Delay: Basic Notation

- The relevant interval of continuous time:

$$[t_0, t_f] \subset \Re^1_+$$

- The key ingredient and source of most complications: unit delay operator for path p expressed as

$$D_p(t, h) \quad \forall p \in \mathcal{P}$$

where

$\mathcal{P}$ is the set of all paths employed by travelers

t denotes departure time

h is a vector of departure rates ("path flows")

DUE with Exogenous Delay: Effective Delay Operators

- Let

$$T_A = \text{the desired arrival time} < t_f$$

- A schedule delay assessed when

$$\text{actual arrival time} = t + D_p(t, h) \neq T_A$$

- The schedule delay is denoted by

$$F\left[t + D_p(t, h) - T_A\right] = \text{ so-called schedule delay}$$

- The effective unit path delay operator for path is

$$\Psi_p(t, h) = D_p(t, h) + F\left[t + D_p(t, h) - T_A\right] \qquad \forall p \in P$$

DUE with Exogenous Delay: Flow Conservation

- Fixed trip matrix $Q = (Q_{ij} : (i,j) \in \mathcal{W})$ where

$$\begin{aligned} Q_{ij} &\in \Re^1_{++} \text{ fixed travel demand for origin-destination pair } (i,j) \in \mathcal{W} \\ \mathcal{W} &= \text{the set of all origin-destination pairs} \end{aligned}$$

- Let

$$\mathcal{P}_{ij} = \text{subset of paths that connect origin-destination pair } (i,j) \in \mathcal{W}.$$

- Flow conservation constraints:

$$\sum_{p \in \mathcal{P}_{ij}} \int_{t_0}^{t_f} h_p(t)\, dt = Q_{ij} \quad \forall (i,j) \in \mathcal{W}$$

DUE with Exogenous Delay: Pure Assignment, No Network Loading

- The set of feasible flows:

$$\Lambda_0 = \left\{ h \geq 0 : \sum_{p \in \mathcal{P}_{ij}} \int_{t_0}^{t_f} h_p(t)\, dt = Q_{ij} \ \forall (i,j) \in \mathcal{W} \right\}$$

- User equilibrium:

Definition

Dynamic user equilibrium. A vector of departure rates (path flows) $h^* \in \Lambda_0$ is a dynamic user equilibrium if

$$h_p^*(t) > 0, p \in P_{ij} \Longrightarrow \Psi_p\left[t, h^*(t)\right] = v_{ij}$$

DUE with Exogenous Delay: VI Formulation

- DUE is known to be equivalent to the following variational inequality (VI) under mild regularity conditions:

$$\left.\begin{array}{c} \text{find } h^* \in \Lambda_0 \text{ such that} \\ \sum_{p\in\mathcal{P}} \int_{t_0}^{t_f} \Psi_p(t, h^*)(h_p - h_p^*)dt \geq 0 \\ \forall h \in \Lambda_0 \end{array}\right\}$$

- However, as we shall next see, the above is equivalent to a differential variational inequality (DVI).
- The DVI form is especially easy to analyze and opens the door to a broad collection of computational methods.

- We note that

$$\sum_{p \in \mathcal{P}_{ij}} \int_{t_0}^{t_f} h_p(t)\, dt = Q_{ij} \ \forall (i,j) \in \mathcal{W}$$

$$\Updownarrow$$

$$\begin{cases} \dfrac{dy_{ij}}{dt} = \displaystyle\sum_{p \in \mathcal{P}_{ij}} h_p(t) & \forall (i,j) \in \mathcal{W} \\ y_{ij}(t_0) = 0 & \forall (i,j) \in \mathcal{W} \\ y_{ij}(t_f) = Q_{ij} & \forall (i,j) \in \mathcal{W} \end{cases}$$

which is recognized as a two-point boundary-value problem.

DUE with Exogenous Delay: DVI Formulation

- Thus, the DUE problem is expressible as:

$$\left.\begin{array}{c} \text{find } h^* \in \Lambda \text{ such that} \\ \sum_{p\in\mathcal{P}} \int_{t_0}^{t_f} \Psi_p(t,h^*)(h_p - h_p^*)dt \geq 0 \\ \forall h \in \Lambda \end{array}\right\} DVI(\Psi, \Lambda, t_0, t_f)$$

where

$$\begin{aligned} \Psi &= \text{the effective delay operator} \\ \Lambda &= \left\{ \begin{array}{c} h \geq 0 : \frac{dy_{ij}}{dt} = \sum_{p\in P_{ij}} h_p\left(t\right),\ \ y_{ij}(t_0) = 0,\ \ y_{ij}\left(t_f\right) = Q_{ij} \\ \forall\left(i,j\right) \in \mathcal{W} \end{array} \right\} \\ Q &= \left(Q_{ij}\right) = \text{the trip matrix} \\ t_0 &= \text{ start time} \\ t_f &= \text{ end time} \end{aligned}$$

A Class of Abstract DVIs

- An abstract $DVI(F, U, t_0, t_f)$:

$$\text{find } u^* \in U \text{ such that}$$
$$\langle F(x(u^*), u^*, t), u - u^* \rangle \geq 0 \quad \forall u \in U$$

where

$$u \in U \subseteq \left(L^2[t_0, t]\right)^m$$
$$x = \arg\left\{ \begin{array}{c} \frac{dy}{dt} = f(y, u, t), \; y(t_0) = x^0 \\ \Phi[y(t_f), t_f] = 0 \end{array} \right\} \in \left(\mathcal{H}^1[t_0, t_f]\right)^m$$

Comments on Abstract DVIs

1. The DVIs of the previous slide are related to optimal control problems in essentially the same way mathematical programs are related to finite-dimensional VIs.
2. Cournot-Nash differential (i.e. dynamic) games are a specific realization of the DVI problem.
3. Pang and Stewart: *Mathematical Programming Series B*, 2008
4. Friesz: *Dynamic Optimization and Differential Games*, Springer, 2010

DVI Regularity

- $DVI(F, f, U, t_0, t_f)$ is regular if certain smoothness, differentiability and shape conditions hold:
  - $x(u)$ exists; is unique, continuous and G-differentiable with respect to u
  - $F(x, u, t)$ is continuous with respect to x and u
  - $f(x, u, t)$ is continuous with respect to x and u
  - $\Phi(x, t)$ is continuously differentiable with respect to x
  - U is convex and compact
- Regularity assures existence and allows derivation of necessary conditions that generalize the Pontryagin minimum principle.
- Only boundedness is a problem for DUE.

Gap Function

Definition

Gap function defined. A function $G : U \longrightarrow \Re_+$ is called a gap function for $DVI(F, f, U, t_0, t_f)$ when the following statements hold:

1. $G(u) \geq 0$ for all $u \in U$
2. $G(u) = 0$ if and only if u is the solution of $DVI(F, f, U, t_0, t_f)$.

DUE DVI Equivalence with Exogenous Delay

Theorem

Dynamic user equilibrium equivalent to a differential variational inequality. *Assume* $\Psi_p(\cdot, h) : [t_0, t_f] \longrightarrow \Re^1_+$ *is measurable and strictly positive for all* $p \in \mathcal{P}$ *and all* $h \in \Lambda$. *A vector of departure rates (path flows)* $h^* \in \Lambda$ *is a dynamic user equilibrium if and only if* h^* *solves* $DVI(\Psi, \Lambda, t_0, t_f)$.

Proof.

The proof of equivalence is greatly facilitated by the DVI reformulation.
Two parts:
(i) $[DUE(\Psi, \Lambda, t_0, t_f) \Longrightarrow DVI(\Psi, \Lambda, t_0, t_f)]$ Trivial.
(ii) $[DVI(\Psi, \Lambda, t_0, t_f) \Longrightarrow DUE(\Psi, \Lambda, t_0, t_f)]$ Informative. Let's have a look .... □

DVI: A Fictitious Optimal Control Problem

Note that $\forall h \in \Lambda$:

$$DVI \iff \sum_{(i,j)\in\mathcal{W}} \sum_{p\in\mathcal{P}_{ij}} \int_{t_0}^{t_f} \Psi_p(t,h^*)h_p dt \geq \sum_{(i,j)\in\mathcal{W}} \sum_{p\in\mathcal{P}_{ij}} \int_{t_0}^{t_f} \Psi_p(t,h^*)h_p^* dt$$

$$\Uparrow$$

$$\begin{aligned}
\min J_0 &= \sum_{(i,j)\in\mathcal{W}} v_{ij}\left[Q_{ij} - y_{ij}(t_f)\right] + \sum_{(i,j)\in\mathcal{W}} \sum_{p\in\mathcal{P}_{ij}} \int_{t_0}^{t_f} \Psi_p(t,h^*)h_p dt \\
\frac{dy_{ij}}{dt} &= \sum_{p\in\mathcal{P}_{ij}} h_p(t) \quad \forall (i,j) \in \mathcal{W} \\
y_{ij}(t_0) &= 0 \quad \forall (i,j) \in \mathcal{W} \\
h &\geq 0
\end{aligned}$$

DVI Equivalence: The Optimality Conditions

- Hamiltonian:

$$H = \sum_{(i,j)\in\mathcal{W}} \sum_{p\in\mathcal{P}_{ij}} \Psi_p(t, h^*) h_p + \sum_{(i,j)\in\mathcal{W}} \lambda_{ij} \sum_{p\in\mathcal{P}_{ij}} h_p$$

$$= \sum_{(i,j)\in\mathcal{W}} \left\{ \sum_{p\in\mathcal{P}_{ij}} \left[\Psi_p(t, h^*) + \lambda_{ij}\right] h_p \right\}$$

- Adjoint equations:

$$\frac{d\lambda_{ij}}{dt} = -\frac{\partial H}{\partial y_{ij}} = 0$$

- Transversality:

$$\lambda_{ij}(t_f) = \frac{\partial \sum_{(i,j)\in\mathcal{W}} v_{ij}\left[Q_{ij} - y_{ij}(t_f)\right]}{\partial y_{ij}(t_f)} = -v_{ij} = \text{constant}$$

DVI Equivalence: The Minimum Principle

- Minimum principle:

$$\min H \quad \text{s.t.} \quad -h \leq 0$$

- Since $\lambda_{ij}(t_f) = v_{ij}$, the Kuhn-Tucker conditions are

$$\begin{aligned} \Psi_p(t, h^*) - v_{ij} &= \rho_p \geq 0 \\ \rho_p h_p &= 0 \end{aligned}$$

- Thus

$$\begin{array}{llcll} (1) & h_p^* > 0, p \in \mathcal{P}_{ij} & \Longrightarrow & \Psi_p(t, h^*) = v_{ij} & \textit{DUE} \text{ definition} \\ (2) & \Psi_p(t, h^*) > v_{ij}, p \in \mathcal{P}_{ij} & \Longrightarrow & h_p^* = 0 & \text{trivial flow condition} \end{array}$$

The Notion of a Singular Control, Slide 1

- Consider an optimal control problem with state x, adjoint λ, and control u obeying the constraint

$$L \leq u \leq U \qquad L, U \in \Re^1_{++} \text{ and } L < U$$

- Assume the Hamiltonian is linear in its control u:

$$H = \Phi(x, \lambda, t) + S(x, \lambda, t)u$$

- S is the switching function
- The minimum principle requires u^* to be "bang-bang" with the prospect of a singular control u_s:

$$u^* = \begin{cases} L & \text{if} \quad S > 0 \\ U & \text{if} \quad S < 0 \\ u_s & \text{if} \quad S = 0 \end{cases}$$

The Notion of a Singular Control, Slide 2

- If the switching function S vanishes on some $[t_1, t_2] \subseteq [t_0, t_f]$ for $t_1 < t_2$, then there is a singular control u_s.
- Generally speaking the solution of such a problem requires:
  1. A "bang" control to get on the singular trajectory
  2. Navigation along the singular trajectory
  3. Another "bang" control may be used to get off the singular trajectory
  4. Synthesis of bang-bang and singular controls

Analysis of DUE Solutions, Slide 1

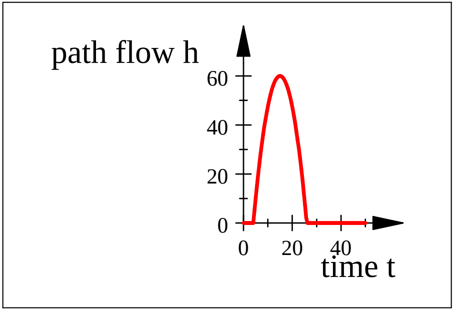
- The Hamiltonian is linear in the controls h_p:

$$
\begin{aligned}
H &= \sum_{(i,j)\in\mathcal{W}} \sum_{p\in\mathcal{P}_{ij}} \left[\Psi_p(t,h^*) + \lambda_{ij}\right] h_p \\
&= \sum_{(i,j)\in\mathcal{W}} \sum_{p\in\mathcal{P}_{ij}} \left[\Psi_p(t,h^*) - v_{ij}\right] h_p
\end{aligned}
$$

- All bounded DUE controls are either zero or singular.
- That is

$$
h_p > 0 \Longrightarrow \begin{cases} \Psi_p(t,h^*) - v_{ij} = 0 \\ \text{and} \\ \dfrac{d^n \Psi_p(t,h^*)}{dt^n} = 0 \qquad n = 1, 2, \ldots \end{cases}
$$

- The departure rates (path flows) are either singular or zero.
- The singular portions of the solution are smooth.
- The departure rates (path flows) are piecewise smooth.
- The departure rates and arc exit flows typically have an inverted parabolic shape:



- Implications?

Network Loading

- I will first introduce universal features of network loading.
- Following that, I use a particular variety of network loading based on the so-called point-queue model (PQM), to illustrate the network loading phase.
- Subsequently, I will present an overview of Lighthill-Whitham-Richards (LWR) based hydrodynamic network loading.

Generic Network Loading: More Notation

- Path:

$$p = \left\{ a_1, a_2, ..., a_i, ..., a_{m(p)} \right\}$$

- Arc exit flows:

$$g^p_{a_i}$$

- Traffic volume on an arc of a path:

$$x^p_{a_i}$$

- Definition:

$$\text{Finding arc exit flows } (g) \text{ and arc volumes } (x) \;=\; \text{network loading}$$

- The path delays $\Psi_p(t, h)$ are a by-product of network loading.

Generic network Loading: The Fundamental Recursion (Generic Loading)

- Denote arc delay by

$$D_{a_i}(t)$$

- The *fundamental recursion* always holds:

$$\xi^{p}_{a_i}(t) = \xi^{p}_{a_{i-1}}(t) + D_{a_i}\left[\xi^{p}_{a_{i-1}}(t)\right] \tag{21}$$

where

$$\xi^{p}_{a_i}(t) \quad = \quad \text{the time of exiting } a_i \text{ of path } p \text{ given departure from the origin at time } t \in [t_0, t_f]$$

- The fundamental recursion (21) is relevant to both the PQM and the LWR PDAE system.

Generic Network Loading: Determining Path Delay

- The fundamental recursion allows the exit times for every arc to be computed for any given departure time t when the arc delay functions $D_{a_i}(t)$ are known.
- Arc exit times allow path delay to be computed.
- Total path delay is the sum of arc delays expressed via the ξ-functions:

$$D_p(t) = \sum_{i=1}^{m(p)} \left[\xi_{a_i}^p(t) - \xi_{a_{i-1}}^p(t) \right] = \xi_{a_{m(p)}}^p(t) - t$$

PQM Network Loading: Arc Dynamics

- Arc delay functions:

$$D_{a_i}\left[x_{a_i}(t)\right] \quad \forall p \in P, i \in [1, m(p)]$$

- State Dynamics:

$$\begin{aligned}
\frac{dx_{a_1}(t)}{dt} &= h_p(t) - g^p_{a_1}(t) \quad \forall p \in P \\
\frac{dx_{a_i}(t)}{dt} &= g^p_{a_{i-1}}(t) - g^p_{a_i}(t) \quad \forall p \in P, i \in [2, m(p)] \\
x^p_{a_i}(t_0) &= x^p_{a_i,0} \quad \forall p \in P, i \in [1, m(p)]
\end{aligned}$$

- Approximation, precise only in the limit of infinitesimal arc length.
- Total Arc Volume:

$$x_a(t) = \sum_{p \in P} \delta_{ap} x^p_a(t) \qquad \forall a \in A$$

PQM Network Loading: Flow Propagation Constraints

- Flow propagation constraints:

$$g_{a_i}^{p}\left(t + D_{a_i}\left[x_{a_i}\left(t\right)\right]\right)\left(1 + \dot{D}_{a_i}\left[x_{a_i}\left(t\right)\right] \cdot \dot{x}_{a_i}\left(t\right)\right) = g_{a_{i-1}}^{p}\left(t\right)$$

- FIFO condition

$$1 + \dot{D}_{a_i}\left[x_{a_i}\left(t\right)\right] \cdot \dot{x}_{a_i}\left(t\right) > 0$$

satisfied when exit flows are nontrivial.

$\forall p \in P, i \in [1, m(p)]$

$$\begin{aligned}
\frac{dx_{a_1}(t)}{dt} &= h_p(t) - g^p_{a_1}(t) \\
\frac{dx_{a_i}(t)}{dt} &= g^p_{a_{i-1}}(t) - g^p_{a_i}(t) \\
x^p_{a_i}(t_0) &= x^p_{a_i,0} \\
g^p_{a_{i-1}}(t) &= g^p_{a_i}\left(t + D_{a_i}\left[x_{a_i}(t)\right]\right)\left(1 + \dot{D}_{a_i}\left[x_{a_i}(t)\right] \cdot \dot{x}_{a_i}(t)\right) \\
h_p(t) &= g^p_{a_1}\left(t + D_{a_1}\left[x_{a_1}(t)\right]\right)\left(1 + \dot{D}_{a_1}\left[x_{a_1}(t)\right] \cdot \dot{x}_{a_1}(t)\right)
\end{aligned}$$

PQM Network Loading: Flow Propagation (FP) Constraints

- The FP constraints:

$$\text{flow entering} = \text{flow exiting} \times \text{platoon contraction/expansion factor}$$

- Derived from exit-time identity, arc dynamics, arc delay function, and fundamental recusion via the chain rule.
- The FP constraints:

$$g_{a_i}^{p}\left(t + D_{a_i}\left[x_{a_i}(t)\right]\right)\left(1 + \dot{D}_{a_i}\left[x_{a_i}(t)\right] \cdot \dot{x}_{a_i}(t)\right) = g_{a_{i-1}}^{p}(t)$$

- Not good.

PQM Network Loading: Approximate FP Constraints

- 2nd Order Taylor approximation:

$$g_{a_i}^{p}\left(t + D_{a_i}\left[x_{a_i}(t)\right]\right) \approx g_{a_i}^{p}(t) + \frac{dg_{a_i}^{p}(t)}{dt}\left(D_{a_i}\left[x_{a_i}(t)\right]\right) + \frac{1}{2}\frac{d^2 g_{a_i}^{p}}{dt^2}\left(D_{a_i}\left[x_{a_i}(t)\right]\right)^2$$

- Introducing new dummy state variables $r_{a_i}^{p}$, we reduce network loading to a system of ODEs.

$\forall a \in A, p \in P, i \in [1, m(p)]$:

$$\left.\begin{aligned}
\frac{dx_{a_i}^p}{dt} &= g_{a_{i-1}}^p - g_{a_i}^p \\
\frac{dr_{a_i}^p}{dt} &= R_{a_i}^p\left(r_{a_i}^p, g_{a_{i-1}}^p, g_{a_i}^p, x_{a_i}, D_{a_i}\right) \\
\frac{dg_{a_i}^p}{dt} &= r_{a_i}^p \\
\frac{dD_{a_i}}{dt} &= D'_{a_i}\dot{x}_{a_i} \\
&\text{plus initial conditions}
\end{aligned}\right\} \Longrightarrow \Psi_p$$

Numerical Approaches Independent of the Form of Network Loading

- We "go to" the DAE or PDAE version of network loading when we need to know the effective path delay for a given h, within an algorithm for solving $DVI(\Psi, \Lambda, t_0, t_f)$.
- Continuous-time algorithms for $DVI(\Psi, \Lambda, t_0, t_f)$:
  - descent in Hilbert space using gap function
  - fixed-point algorithm in Hilbert space
  - these may also be implemented in discrete time, subject to certain caveats
- We have not encountered difficulties with the fixed point algorithm, despite conventional wisdom to the contrary.

Descent With Gap Function in Continuous Time, Slide 1

- Where $G(u)$ is a differentiable gap functional, let us illustrate the essentials of mathematical programming in continuous time by considering an unconstrained problem:

$$\begin{aligned} \min G(u) &= K\left[x(t_f), t_f\right] + \int_{t_0}^{t_f} f_0\left((x, u, t\right) dt \\ \frac{dx}{dt} &= f(x, u, t) \\ x(t_0) &= x_0 \end{aligned}$$

- Constraints are handled in the usual way by penalties, barriers and projection.
- Assertion: descent in Hilbert space overcomes the two point boundary nature of DVIs.

Step 0. Initialization. Set $k = 0$ and pick $u^0(t) \in \left(L^2[t_0, t_f]\right)^m$.
Step 1. Find State Trajectory. Using $u^k(t)$, find $x^k(t)$ that solves the state initial value problem

$$\frac{dx}{dt} = f\left(x, u^0, t\right) \qquad x(t_0) = x_0$$

Step 2. Find Adjoint Trajectory. Using $u^k(t)$ and $x^k(t)$, find $\lambda^k(t)$ that solves the adjoint final value problem

$$(-1)\frac{d\lambda}{dt} = \frac{\partial H\left(x^k, u^k, \lambda, t\right)}{\partial x} \qquad \lambda(t_f) = \frac{\partial K\left[x(t_f), t_f\right]}{\partial x}$$

Fixed Point Formulation and Algorithm

Step 3. Find Gradient. Using $u^k(t)$, $x^k(t)$ and $\lambda^k(t)$ find gradient in $L^2[t_0, t_f]$:

$$\begin{aligned}\nabla_u G(u^k) &= \frac{\partial H\left(x^k, u^k, \lambda, t\right)}{\partial u} \\ &= \frac{\partial f_0(x^k, u^k, t)}{\partial u} + \left(\lambda^k\right)^T \frac{\partial f\left(x^k, u^k, t\right)}{\partial u}\end{aligned}$$

Step 4. Update and Apply Stopping Test. Set $k = k + 1$ and stop or go to Step 1.

- A (D)VI may be re-stated as a fixed point problem:

$$h = P_\Lambda \left[h - \alpha \Psi(t, h)\right]$$

- The associated continuous-time algorithm for arbitrary $\alpha \in \Re^1_{++}$:

$$h^{k+1} = \arg\min_h \left\{ \frac{1}{2} \left\| h^k - \alpha \Psi\left(t, h^k\right) - h \right\|^2 : h \in \Lambda \right\}$$

The Fixed Point Projection Subproblem

- At each iteration k, we must solve a linear-quadratic optimal control problem when $h \geq 0$ is relaxed:

$$\min_{h} J^{k}(h) = \sum_{(i,j)\in W} v_{ij}^{k}\left[Q_{ij} - y_{ij}(t_f)\right] + \int_{t_0}^{t_f} \frac{1}{2}\left[h^k - \alpha\Psi\left(t, h^k\right) - h\right]^2 d$$

subject to

$$\begin{aligned} \frac{dy_{ij}}{dt} &= \sum_{p\in P} h_p(t) \quad \forall (i,j) \in \mathcal{W} \\ y_{ij}(t_0) &= 0 \quad \forall (i,j) \in \mathcal{W} \end{aligned}$$

- When $h \geq 0$ enforced, subproblem may be solved by descent in Hilbert space.
- Note: derivatives of $\Psi\left(t, h^k\right)$ are not needed.
- But how do we find the current dual variables v_{ij}^k?

Computing the Dual Variables

- A critical challenge in optimal control is finding dual variables.
- Fortunately this is easy for our present circumstance:

$$\begin{aligned} h_p^{k+1} &= \arg\left\{\frac{\partial H^k}{\partial h_p} = 0\right\} \qquad\qquad \forall\,(i,j) \in W,\;\; p \in P_{ij} \\ &= \arg\left\{\left[h_p^k - \alpha\Psi_p\left(t, h^k\right) - h_p\right](-1) - v_{ij}^k = 0\right\}_+ \\ &\Downarrow \\ h_p^{k+1} &= \left\{h_p^k - \alpha\Psi_p\left(t, h^k\right) + v_{ij}^k\right\}_+ \end{aligned}$$

- Flow conservation requires:

$$\int_{t_0}^{t_f} \sum_{p \in P_{ij}} h_p^{k+1}(t)\,dt = \int_{t_0}^{t_f} \sum_{p \in P_{ij}} \left[h_p^k - \alpha\Psi_p\left(t, h^k\right) + v_{ij}^k\right] dt = Q_{ij}$$

- The above is easily solved by line search for each $(i,j) \in \mathcal{W}$; the searches may be performed in parallel.

Convergence and the Effective Delay Operator

- The effective delay operator has been studied by many scholars and its "shape" remains a topic of basic research.
- To apply the classical results on convergence, the effective delay operator must be strongly monotonic.
- Examples of non-monotone delay exist.
- Also monotonic delay operators that are consistent with LWR PDE (hydrodynamic traffic flow theory) have been reported by Perakis and Roels (2007), for certain traffic environments.
- What to do?

Traditional Regularity Conditions Assuring Fixed-Point Convergence

R1. $\Lambda \subset \left(L^2_+ \left[t_0, t_f\right]\right)^{|\mathcal{P}|}$; note in particular that $h \geq 0$.

R2. The unit path delay operator $\Psi(\cdot, h)$ is measurable and strictly positive on Λ.

R3. The unit path delay operator $\Psi\left(\cdot, h\right)$ is continuous on Λ.

R4. The unit path delay operator obeys the Lipschitz condition

$$\left\|\Psi^{k+1} - \Psi^k\right\| \leq \sqrt{K_0}\left\|h^{k+1} - h^k\right\|$$

on Λ, where

$$\Psi^k \equiv \Psi\left(\cdot, h^k\right)$$

and $K_0 \in \Re^1_{++}$.

R5. The unit path delay operator $\Psi\left(\cdot, h\right)$ is strongly monotone on Λ with constant $K_1 \in \Re^1_{++}$; that is

$$\langle \Psi^{k+1} - \Psi^k, h^{k+1} - h^k \rangle \geq \frac{K_1}{2}\left\|h^{k+1} - h^k\right\|^2$$

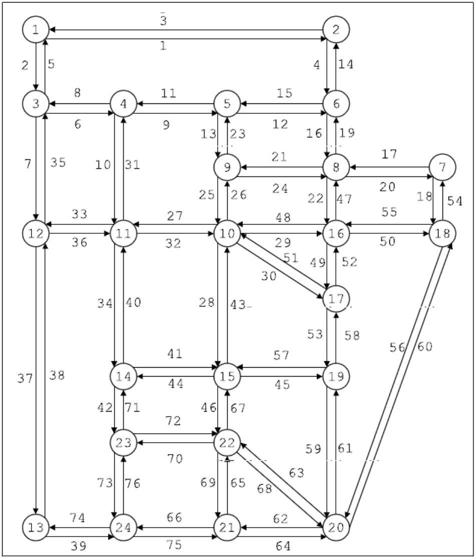
Numerical Example: Sioux Falls Network Description

Sioux Falls Network: 76 arcs, 20 nodes, 10 origin-destination pairs
Fixed travel demand for each OD pairs:

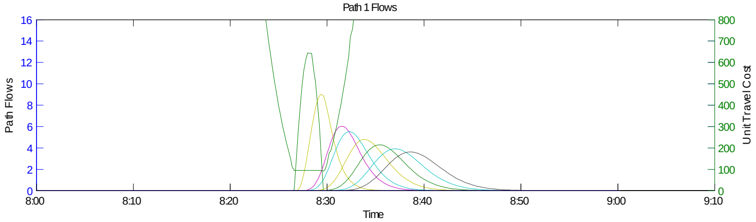
$$Q_{ij} = 100 \quad \forall\,(i,j) \in \mathcal{W}$$

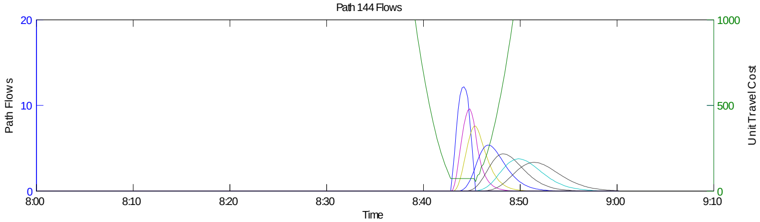
There are 200 paths associated with the 10 origin destination pairs.

Numerical Example: Sioux Falls Network Illustrated

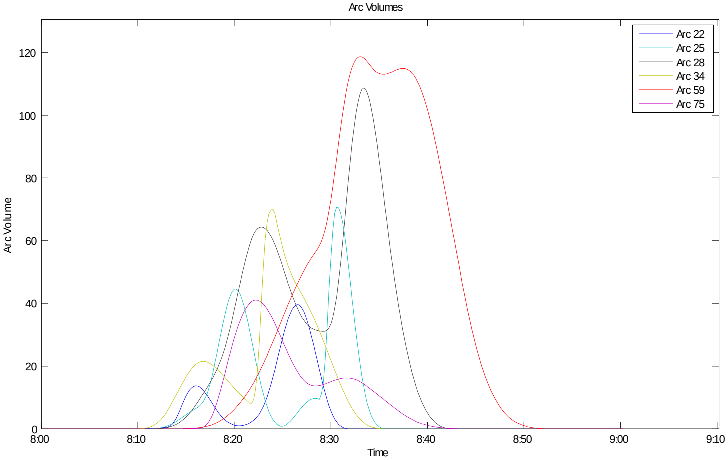


Numerical Example: Path and Arc Exit Flows by Fixed-Point Algorithm, Graphical Presentation





Numerical Example: Arc Volumes by Fixed-Point Algorithm, Graphical Presentation



Numerical Example: Sioux Falls Network, Comparisons by Major Iterations

Network	**PQM**	**KP-LWR**	**CTM**[1]
3 arcs, 4 nodes	13	31	14
6 arcs, 5 nodes	11	-	12
19 arcs, 13 nodes	14	-	10
Sioux Falls	14	-	13

Table: DUE Fixed-Point Major Iterations by Problem Type

[1] The Cell Transmission Model (CTM) is an approximation based on (1) the spatial discretization of each arc, (2) use of finite differences, and (3) a triangular flow vs concentration diagram.

Numerical Example: Sioux Falls Network, Comparisons by CPU Time

Network	PQM	KP-LWR	CTM[2]
3 arcs, 4 nodes	16.1	37.8	21.0
6 arcs, 5 nodes	26.4	-	31.7
19 arcs, 13 nodes	152.2	-	159.1
Sioux Falls	1975	-	3136

Table: Computation Time (seconds) by Problem Type

$$1975 \text{ sec} \approx 33 \text{ min}$$
$$3136 \text{ sec} \approx 52 \text{ min}$$

[2] Cell Transmission Model

Network Loading Based on LWR Theory: The PDAE System

LWR PDE:	$\frac{\partial f_{a_i}(x_{a_i},t)}{\partial x_{a_i}} + \frac{\partial k_{a_i}(x_{a_i},t)}{\partial t} = 0$
Fund'l Diagram:	$f_{a_i}(x_{a_i},t) = u_{a_i}(x_{a_i},t)\,k(x_{a_i},t)$
Recursive ID:	$\xi^p_{a_{i+1}} = \xi^p_{a_i} + D_{a_i}\left(\xi^p_{a_i}\right)$
Bnd'y Condition:	$f^p_{a_i}(L_{a_i},t) = f^p_{a_{i+1}}\left(L_{a_{i+1}}, t + D_{a_{i+1}}(t)\right)$
Propagation:	$h_p(t) = f^p_{a_1}(L_{a_1}, t + D_{a_1})$
Propagation:	$f^p_{a_i}\left(L_{a_i}, \xi^p_{a_i}\right) = f^p_{a_{i+1}}\left(L_{a_{i+1}}, \xi^p_{a_{i+1}}\right)$
Definitional:	$f_{a_i}(x_{a_i},t) = \sum_p \delta_{a_i p} f^p_{a_i}(x_{a_i},t)$
Definitional:	$\frac{\partial D_{a_i}(x_{a_i},t)}{\partial x_{a_i}} = \frac{1}{u_{a_i}}$

Network Loading Based on LWR Theory: Remarks 1

- Alternative, familiar form of LWR PDE:

$$\frac{\partial k_{a_i}}{\partial t} + V_{a_i}\left(k_{a_i}\right)\frac{\partial k_{a_i}}{\partial x_{a_i}} = 0$$

  where the term

$$V_{a_i}\left(k_{a_i}\right) = \frac{\partial f_{a_i}\left(x_{a_i}, t\right)}{\partial k_{a_i}}$$

  requires invoking a so-called "state law." Such laws tend to be specialized by flow regime.
- LWR theory is the preferred foundation for within link traffic dynamics due to its potential to predict shockwaves.
- Multiple Solution types
  - classical solution: smooth
  - weak solution: derivatives may not exist everywhere
  - entropy solution: supplemental conditions to assure uniqueness, especially in hyperbolic equations like the LWR PDE.

Network Loading Based on LWR Theory: Remarks 2

- There are no physically meaningful classical solutions !!!!!!!!!!!!
- Extensions to commodity flow approximations for supply chains. Perakis and Kachani (2010)
- Approximate Solution Expressed as the So-Called "Cell Transmission Model" by Daganzo (1995)
- Hax (1973) theorem gives a nonclassical solution in variational form: this is the foundation for a theoretically rigorous network loading model based on LWR theory.

Network Loading Based on LWR Theory: Challenges

- Modeling spillback
- Modelling passing behavior
- Numerical tractability

Network Loading: Other Foundation Models

- Kinetic/Statisical Mechanics theory of traffic by Priogine and Herman (1971)
- Car following theory [Pipes (1953)]

Network Loading: The Way Ahead

- Multi-spatial scale (continuum vs. network)
- Muli-agent (pedestrians, vehicular, freight modes)
- Multi-time scale (within-day, day-to-day, long term)
- Multi-physics (fluid, gas, Newtonian mechanics of individual vehicles)

Unanswered Questions/Research Opportunities

1. There are many opportunities (glass is half full).
2. We need to extend the SRD DUE modeling framework to consider:
   1. Models with uncertain demand & demand learning
   2. Models with uncertain delay
   3. Elastic demand via DVIs ✓
   4. PDAE systems for LWR
   5. Enroute updating
   6. Explicit feedback control/closed loop games
   7. Theory of dual day-to-day and within-day time scales ✓
3. There are unresolved theoretical and computational issues:
   1. Existence without a priori bounds
   2. Algorithm convergence with nonmonotonic operators
   3. Computation with dual time scales, general case
   4. Computation with uncertainty
   5. computation with robustness

Successful Transfer of this DVI Formalism to Other Applications

1. revenue management, especially computation ✓
2. service pricing ✓
3. electric power pricing ✓
4. urban freight and city logistics ✓
5. dynamic congestion tolls ✓ (Discussed in Part III, which follows)
6. models of the Internet and electronic commerce (Ongoing)

Part III. Equilibrium Network Design

The Price of Anarchy

1. The *price of anarchy* is a term proposed by Papadimitriou (2001) to describe the ratio:

$$\rho = \frac{\text{total latency arising from a Nash-like equilibrium flow pattern}}{\text{minimum total latency of a fully cooperative flow pattern}}$$

2. In the language of traffic assignment: the *price of anarchy* will be ratio of total congestion arising from user equilibrium traffic assignment to minimum total congestion arising from system optimal traffic assignment.
3. As such, the price of anarchy capures the inefficiency associated with a user equilibrium among network users.
4. Clearly

$$\rho \geq 1$$

Notation, Slide 1

We will employ the following set notation familiar from Part I:

$\mathcal{W}$ is the set of origin-destation pairs of the network of interest

$\mathcal{N}$ is the set of nodes of the network of interest

$\mathcal{A}$ is the set of arcs of the network of interest

$\mathcal{P}$ is the set of paths of the network of interest

$\mathcal{P}_{ij}$ is the set of paths connecting OD pair $(i, j) \in \mathcal{W}$

Notation, Slide 1

We will also employ the following matrix notation

$$\delta_{ap} = \begin{cases} 1 & \text{if } a \in p \\ 0 & \text{if } a \notin p \end{cases}$$

$$\Delta = [\delta_{ap} : a \in \mathcal{A},\ p \in \mathcal{P}] \text{ is the arc-path incidence matrix}$$

$$\gamma_{ij}^{p} = \begin{cases} 1 & \text{if } p \in \mathcal{P} \text{ connects } (i,j) \in \mathcal{W} \\ 0 & \text{otherwise} \end{cases}$$

$$\Gamma = \left[\gamma_{ij}^{p} : (i,j) \in \mathcal{W},\ p \in \mathcal{P}\right] \text{ is the path-OD incidence matrix}$$

Notation, Slide 3

Other key notation is the following:

$$h \in \Re_{+}^{|\mathcal{P}|} \quad \text{is a vector of path flows}$$

$$f \in \Re_{+}^{|\mathcal{A}|} \quad \text{is a vector of arc flows}$$

$$Q \in \Re_{++}^{|\mathcal{W}|} \quad \text{is a vector of fixed travel demands}$$

$$c(f) \in \Re_{++}^{|\mathcal{A}|} \quad \text{is a vector of arc cost functions}$$

$$c(h) \in \Re_{++}^{|\mathcal{A}|} \quad \text{is a vector of path cost functions}$$

In fact, the network of interest associates the graph $\mathcal{G}\left(\mathcal{N},\mathcal{A}\right)$ with the fixed vector of demands Q and the arc cost function vector $c(f)$; it is denoted by $\left[\mathcal{G}\left(\mathcal{N},\mathcal{A}\right),Q,c(f)\right]$ or simply $[\mathcal{G},Q,c(f)]$ when no confusion will result. We also define the set of feasible solutions in the obvious way:

$$\Omega_0 = \left\{ f : \Gamma h = Q, \quad f = \Delta h, \quad h \geq 0 \right\} \tag{22}$$

As a consequence of the above notation, it is possible to now offer the following formal definition of the price of anarchy:

Definition

Price of anarchy The price of anarchy for network $[\mathcal{G}, Q, c(f)]$ is

$$\rho\left[\mathcal{G}, Q, c(f)\right] = \frac{\sum_{a \in \mathcal{A}} c_a\left(f_a^{ue}\right) f_a^{ue}}{\sum_{a \in \mathcal{A}} c_a\left(f_a^{so}\right) f_a^{so}} \geq 1 \quad (23)$$

where $f^{ue} \in \Omega$ and $f^{so} \in \Omega$ are, respectively, the user equilibrium flow vector and the system optimal flow vector.

Bounding the Price of Anarchy

Theorem

(Roughgarden) Price of anarchy for $[\mathcal{G}, Q, c(f)]$ *with separable linear arc costs. Let* $f \in \Omega$ *and* $f^{so} \in \Omega$*, where* f^{so} *is optimal for* $[\mathcal{G}, Q, c(f)]$*. Then*

$$\rho\left[\mathcal{G}, Q, c(f)\right] = \frac{\sum_{a \in \mathcal{A}} c_a\left(f_a^{ue}\right) f_a^{ue}}{\sum_{a \in \mathcal{A}} c_a\left(f_a^{so}\right) f_a^{so}} \leq \frac{4}{3} \tag{24}$$

Proof.

See any of several works, such as the book Roughgarden (2005) and the paper Roughgarden and Tardos (2002). □

The Braess Paradox

Local enhancement may cause global degradation.

The Braess Paradox and Equilibrium Network Design: An Example, Slide 1

Consider an example presented in LeBlanc (1975); the relevant network is described by the table

arc name	from node	to node
a_1	1	2
a_2	1	3
a_3	2	4
a_4	3	4

That is, we start by considering the following network without arc a_2:

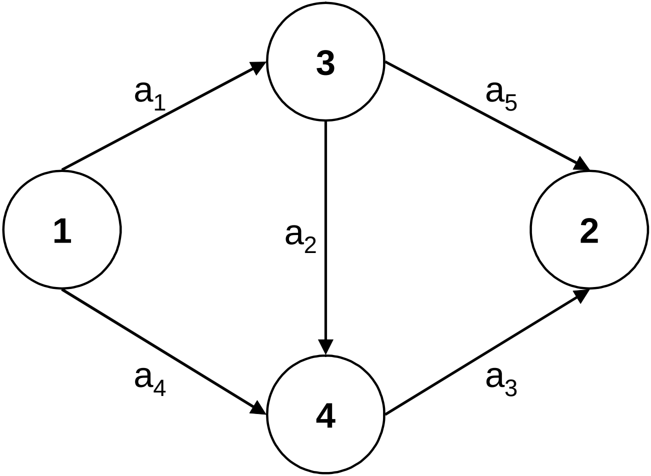


Figure: Five arc network

Example, Slide 3

We presume the following generalized unit cost functions describe congestion on network arcs

$$c_{a_i}\left(f_{a_i}\right) = A_{a_i} + B_{a_i}\left(f_{a_i}\right)^4$$

where

arc	A_{a_i}	B_{a_i}
a_1	40	.5
a_2	185	.9
a_3	185	.9
a_4	40	.5

It is a relatively simple matter to compute, or obtain from symmetry arguments, the user equilibrium flows on this network, when node 1 is the origin and node 2 is the destination and demand obeys

$$h_{p_1} + h_{p_2} = 6 \tag{25}$$

In (25), h_{p_1} and h_{p_2} are the flows on the two paths connecting the single origin-destination pair $(1, 4)$ defined by

$$\begin{aligned} p_1 &= \{a_1, a_3\} \\ p_2 &= \{a_2, a_4\} \end{aligned}$$

Example, Slide 5

In fact the user equilibrium flows for this network configuartion, which we call the *before* configuration, are

$$\begin{aligned} \text{before flow pattern } h^*_{p_1} &= h^*_{p_2} = 3 \\ &\Longrightarrow f_{a_i} = 3 \quad i = 1, 2, 3, 4 \end{aligned}$$

so that the unit path costs are

$$c^*_{p_1} = c^*_{p_2} = 338.4$$

and the total system wide congestion costs for this four arc configuration are

$$TC_{before} = 6\,(338.4) = 2030.4$$

Example, Slide 6

Now we consider improving the network by adding a fifth arc a_5 that directly connects node 2 to node 3. This means that the new network configuration, which we call the *after* configuration, is that given by the table on the right:

arc name	from node	to node
a_1	1	2
a_2	1	3
a_3	2	4
a_4	3	4

$\Longrightarrow$

arc name	from node	to node
a_1	1	2
a_2	1	3
a_3	2	4
a_4	3	4
a_5	2	3

Example, Slide 7

The generalized unit cost for arc a_5 is

$$c_{a_5} = 15.4 + f_{a_5}$$

There is now a third path open between the origin and destination, namely

$$p_3 = \{a_1, a_5, a_4\}$$

The user equilibrium flows for the *after* configuration are

$$\begin{aligned} \textit{after} \text{ flow pattern } h^*_{p_1} &= h^*_{p_2} = h^*_{p_3} = 2 \\ &\Longrightarrow f_{a_i} = 2 \quad i = 1, 2, 3, 4, 5 \end{aligned}$$

since the unit path flow costs are

$$c^*_{p_1} = c^*_{p_2} = c^*_{p_3} = 367.4,$$

which of course means a user equilibrium has been attained

Example, Slide 8

1. The associated total congestion costs of the *after* flow pattern are

$$TC_{after} = 6\,(367.4) = 2204.4$$

2. Thus, we have shown

$$TC_{before} = 2030.4 < 2204.4 = TC_{after}$$

3. Thereby, we have illustrated the Braess paradox.
4. An immediate consequence of the Braess' paradox is that we must include user equilibrium constraints among the constraints of any mathematical formulation of the optimal network design problem when network agents are empowered to make their own selfish routing decisions.

Discrete Network Design: Recall Notation

$$\begin{aligned}
\Delta &= (\delta_{ap}) \\
\delta_{ap} &= \begin{cases} 1 & \text{if } a \in p \\ 0 & \text{if } a \notin p \end{cases} \\
\Gamma &= (\gamma_w^p) \\
\gamma_w^p &= \begin{cases} 1 & \text{if } p \in P_w \\ 0 & \text{if } p \notin P_w \end{cases}
\end{aligned}$$

Discrete Network Design: More Notation

$$M = (M_a : a \in \mathcal{I})$$

$$M_a = \text{a large positive number} \quad \forall a \in \mathcal{I}$$

$$\mathcal{A} = \text{the set of all existing network arcs}$$

$$\mathcal{I} = \text{the set of arcs being considered for insertion into the network}$$

$$B = \text{the available budget}$$

$$y_a = \begin{cases} 1 & \text{if arc } a \in \mathcal{I} \text{ is added to the network} \\ 0 & \text{otherwise} \end{cases}$$

$$y = (y_a : a \in \mathcal{A})$$

$$Q_{ij} = \text{travel demand between OD pair } (i,j) \in \mathcal{W}$$

$$Q = (Q_{ij} : (i,j) \in \mathcal{W})$$

Discrete Network Design, Budgeting Constraints

$$
\begin{aligned}
\text{flow conservation} &: \quad \Gamma h = Q \\
\text{logical constraints} &: \quad My \geq f \\
\text{nonnegative flow} &: \quad h \geq 0 \\
\text{budget constraint} &: \quad \sum_{a \in \mathcal{I}} \beta_a y_a \leq B \ \forall a \in \mathcal{I} \\
\text{discrete improvements} &: \quad y_a = (0, 1)
\end{aligned}
$$

Note that the logical constraints assure flow is zero on any arc not inserted.

Let us presume that we are dealing with discrete capacity additions and that the arc cost functions are separable and strictly monotone increasing while travel demand is fixed.Then

$$\Phi(y) \equiv \arg\left\{\min \sum_{a\in\mathcal{A}\cup\mathcal{I}} \int_0^{f_a} c_a(x_a)\, dx_a \text{ s.t. } f = \Delta h, \right.$$
$$\left. \Gamma h = Q,\ My \geq f,\ h \geq 0 \right\}$$

Discrete Network Design: A Bilevel, Nonconvex Mathematical Program

The complete mathematical formulation of the discrete equilibrium network design model under the assumptions of fixed travel demand and separable arc unit cost functions is the following:

$$\min \sum_{a \in \mathcal{A} \cup \mathcal{I}} c_a \left[f_a \left(y \right) \right] f_a \left(y \right) \tag{26}$$

subject to

$$f \left(y \right) = \Phi \left(y \right) \tag{27}$$

$$y \in \Lambda \equiv \left\{ y : \sum_{a \in \mathcal{I}} \beta_a y_a \leq B,\ y_a = (0,1)\ \forall a \in \mathcal{I} \right\} \tag{28}$$

The above is one instance of a mathematical program with equilibrium constraints (MPEC).

MPEC Defined

We can give the following rather general statement of a finite dimensional mathematical program with equilibrium constraints (MPEC):

$$\min_{y} F(x, y) \tag{29}$$

subject to

$$\begin{aligned} x &\in X(y) \subseteq \Re^n && (30) \\ y &\in Y(x) \subseteq \Re^q && (31) \\ [G(x, y)]^T (x' - x) &\geq 0 \quad \forall x' \in X && (32) \end{aligned}$$

where equilibrium is expressed as a variational inequality in x whose principal operator, $F(., y)$, depends parametrically on y. Furthermore, in problem (29) – (32), the key mappings are

$$\begin{aligned} F(x, y) &: \Re^n \times \Re^q \longrightarrow \Re^1 \\ G(x, y) &: \Re^n \times \Re^q \longrightarrow \Re^n, \end{aligned}$$

No assumptions regarding differentiability or convexity have yet been made.

MPECs and Their Relationship to Stackelberg Games

1. An MPEC of the form (29) – (32) has a one-to-one correspondence with the key features of a *Stackelberg game*.
2. In particular, a Stackelberg game is a type of noncooperative game wherein one of the agents is identifiable as a *leader*, meaning that he/she has the ability to anticipate the reactions of the other agents to his/her strategy decisions.
3. Only the leader is capable of this omniscience; all the other agents are Nash agents whose strategies are nonanticipatory.
4. The objective (29) of our MPEC of course corresponds to the leader's noncooperative play, and it is additionally constrained by considerations (30) that are not directly tied to the followers.
5. Similarly, the Nash followers have certain constraints not directly tied to the leader, as represented by (31). Finally, we note that the leader's play is, to reiterate, constrained by the play of the followers, as represented by (32).

There are at least three principal types of Stackelberg games that are related to a broad interpretation of equilibrium network design as the identification of investments and mechanisms that alleviate congestion while taking into account the Braess paradox; these are:

1. Topological equilibrium network design. The leader of such games minimizes global congestion, subject to budget and equilibrium constraints, by inserting individual arcs or nodes into existing networks. Examples of topological equilibrium network design models are:
    1. the arc addition model (29) – (32) discussed above, whose flows are constrained to be a user equilibrium; and
    2. the equilibrium facility location models developed by Miller et al. (1996) to optimally determine the location, production and shipment activities of a new firm within an existing network economy described by an oligopolistic spatial price equilibrium.

Typology of MPECs, Slide 2

2. Capacity enhancement equilibrium network design. In such games, the leader minimizes global congestion, subject to budget and equilibrium constraints, by increasing the effective capacity of certain existing arcs of the network of interest.
3. Equilibrium pricing. Such Stackeberg games differ from the types described above in that the leader is concerned with the design of a pricing mechanism rather than capital or maintenance investments.
   1. In particular, congestion pricing employs tolls to redistribute traffic in a fashion that diminishes congestion and avoids the Braess paradox. However, they do employ explicit user equilibrium constraints and have a mathematical structure quite similar to capacity enhancement equilibrium design models.
   2. We shall confine our attention to mathematical formulation and numerical solution of congestion pricing models.

The Continuous Equilibrium Network Design Problem, Slide 1

1. Let us turn now to the so called continuous equilibrium network design model and consider how to include elastic travel demand in its formulation.
2. The functions $c_a(f, y_a)$ and $Q_{ij}(u)$ are respectively the nonseparable unit cost of flow on arc a and the nonseparable travel demand for origin-destination pair (i, j). Note that these functions are assumed to be continuous and differentiable. Examples:

$$
\begin{aligned}
c_a(f_a, y_a) &= A_a + B_a \left(\frac{f_a}{k_a + y_a} \right)^4 \\
c_a(f_a, y_a) &= A_a + \frac{B_a}{f_a - (k_a + y_a)} \\
Q_{ij}(u_{ij}) &= \alpha_{ij} - \beta_{ij} u_{ij}
\end{aligned}
$$

3. Of course f_a denotes the flow on arc $a \in \mathcal{A}$.
4. The vector u is the vector formed by concatenating the minimum

The Continuous Equilibrium Network Design Problem, Slide 2

1. Our immediate aim is to develop a means of expressing the notion a user equilibrium as constraints of a mathematical program. Of course, we can always use the complementarity version of the user equilibrium conditions as constraints.

$$\begin{aligned} \left[c_p(h, y) - u_{ij}\right] h_p &= 0 \\ c_p(h, y) - u_{ij} &\geq 0 \end{aligned}$$

2. Alternatively, one can append a variational inequality describing user equilibrium. Note that there are an infinite (uncountable) number of such contraints, as user-equilibrium variational inequality holds for every flow pattern that is nonnegative and satifies flow conservation. Consequently, the explicit use of variational inequality constraints carries computational challenges with it.

The Continuous Equilibrium Network Design Problem, Slide 3

The following theorem presents an alternative mathematical characterization of user equilibrium:

Theorem

*A non-negative path flow $h \geq 0$ is a user equilibrium if and only if (**??**) together with*

$$c_p \geq u_{ij} = g_{ij}(h, y) \equiv \frac{\left(\sum_{q \in P_{ij}} h_q c_q\right)}{Q_{ij}} \quad \forall (i,j) \in \mathcal{W},\ p \in \mathcal{P}_{ij} \qquad (33)$$

are satisfied. Moreover, at equilibrium (33) holds as an equality.

Proof: The proof is quite simple and may be found in Tan et al. (1979). The function $g_{ij}(h, y)$ recognizes that minimum travel cost depends on the current flow pattern (h) and those network improvements (y) that

The Continuous Equilibrium Network Design Problem, Slide 4

In Theorem 14 stated on the previous slide, demand may be either elastic (variable) or inelastic (fixed). The usual conservation constraints

$$\sum_{p \in \mathcal{P}_{ij}} h_p - Q_{ij}(u) = 0 \qquad \forall\, (i,j) \in \mathcal{W} \tag{34}$$

are enforced. Theorem 14 is important because it allows a single level conventional mathematical programming representation of the elastic demand equilibrium network design problem entirely in terms of path flows and improvement variables.

The Continuous Equilibrium Network Design Problem, Slide 5

Theorem 14 above allows any need to enforce a Wardropian static user equilibrium to be expressed by constraints (33), together with flow conservation and nonnegativity. Note that these constraints:

1. are in standard mathematical programming form;
2. are finite in number;
3. require explicit use of path variables;
4. generally make the feasible region nonconvex; and
5. require invocation of a constraint qualification.

The Consumers' Surplus Line Integral Defined

Usually the equilibrium network design problem is studied fo fixed demand or separable because of a substantial complication that arises when nonseparable demand functions are employed. To undersatnd this point let

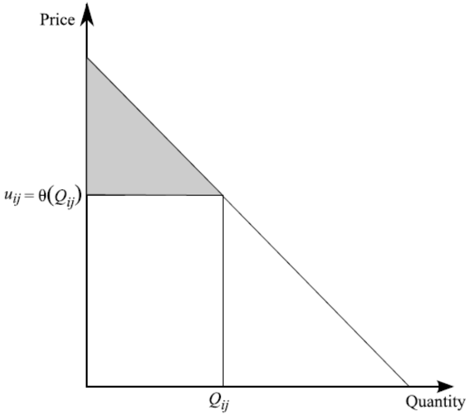
$$\begin{aligned} \Theta_{ij}(Q) &= \text{the inverse travel demand corresponding to } Q \in \Re_{+}^{|\mathcal{W}|} \\ \Theta(Q) &= \left[\Theta_{ij}(Q) : (i,j) \in \mathcal{W}\right] \end{aligned}$$

The net benefits associated with price-quantity pair $(\Theta(Q), Q)$ are given by a line integral for consumers' surplus net of congestion costs; that is, we have

$$CS(Q) = \sum_{(i,j)\in\mathcal{W}} \left[\oint_{0}^{Q} \Theta_{ij}(v)\, dv_{ij} - \Theta_{ij}(Q)\, Q_{ij} \right], \qquad (35)$$

The separable case is illustrated on the next slide.

Consumers' Surplus Based on Invertible Deamnd



Remarks Re Consumers' Surplus Based on Invertible Demand

1. The first term in the definition of $CS(Q)$ (previous slide) measures gross economic benefits and the second term is the payment made for benefits, expressed in terms of congestion costs.
2. Note that a line integral *must* be employed in order to give an exact representation of net economic benefits in the presence of elastic, nonseparable demand functions. This is because consumers' surplus necessarily involves the integration of functions of several variables when demand functions are not separable.
3. This is problematic because it is well known that a line integral does not have a unique, unambiguous value unless the Jacobian matrix formed from its integrand is symmetric.
4. Such symmetry restrictions amount to a requirement that cross price elasticities of demand be proportional to one another, which is unlikely in any real world setting. See Jara-Diaz and Friesz (1982).

Consumers' Surplus Based on Demand (Not Inverse Demand)

Using integration by parts

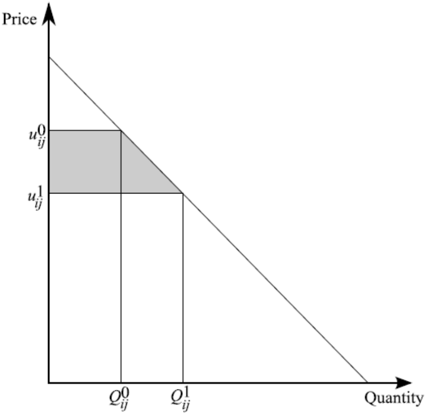
$$CS(u) = \sum_{(i,j)\in\mathcal{W}} \left\{ \left[\Theta_{ij}(u)\, u_{ij}\right]_0^{Q(u)} - \oint_{\tilde{u}}^{u} Q_{ij}(x)\, dx_{ij} - \Theta_{ij}\left[Q(u)\right] Q_{ij}(u) \right\}$$

$$= \sum_{(i,j)\in\mathcal{W}} \oint_{u}^{\tilde{u}} Q_w(x)\, dx_{ij} \qquad (36)$$

where

$$\tilde{u} = \Theta(0) = \text{vector of price axis intercepts (maximal prices)} \qquad (37)$$

The Change in Consumers' Surplus as Our Objective



Line Integral for Change in Consumers' Surplus

Note that

$$\max \Delta CS\left(u, u^0\right) = CS\left(u\right) - CS\left(u^0\right)$$

when the minimum travel costs change is from u^0 to u. In that case

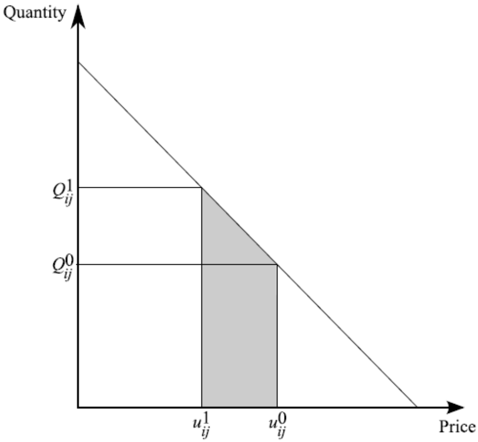
$$\max \Delta CS\left(u, u^0\right) = \sum_{(i,j)\in\mathcal{W}} \oint_{u}^{\tilde{u}} Q_{ij}\left(x\right) dx_{ij} - \sum_{(i,j)\in\mathcal{W}} \oint_{u^0}^{\tilde{u}} Q_{ij}\left(x\right) dx_{ij} \quad (38)$$

which is consistent with the figure presented on the previous slide.

Objective (38) of the previous slide is readily re-expressed as

$$\begin{aligned}\max \Delta CS\left(u, u^0\right) &= \sum_{(i,j)\in\mathcal{W}} \left\{ \oint_{u}^{u^0} Q_{ij}\left(x\right) dx_{ij} + \oint_{u^0}^{\tilde{u}} Q_{ij}\left(x\right) dx_{ij} \right\} \\ &\qquad - \sum_{(i,j)\in\mathcal{W}} \oint_{u^0}^{\tilde{u}} Q_{ij}\left(x\right) dx_{ij} \\ &= \sum_{(i,j)\in\mathcal{W}} \oint_{u}^{u^0} Q_{ij}\left(x\right) dx_{ij} \qquad (39)\end{aligned}$$

Graphical Illustration of the Re-Stated Objective Function



The Network Design Objective Function

By virtue of Theorem 14 we knoq that the Tan et al. inequality (eqejor:1.5) holds as an equality. Therefore, we may express (39) in terms of h and y variables:

$$\max \Delta CS \left(h, y, u^0\right) = \sum_{(i,j)\in\mathcal{W}} \left[\oint_{u}^{u^0} Q_{ij}\left(x\right) dx_{ij} \right]_{u=g(h,y)} \tag{40}$$

where

$$\begin{aligned} g &= \left(g_{ij} : (i,j) \in \mathcal{W}\right) \\ u^0 &= \text{ the known initial vector of minimum travel costs} \end{aligned}$$

It is the form (40) that we use as our objective for optimal design when demand is elastic.

$$\sum_{p\in\mathcal{P}_{ij}} h_p - Q_{ij}(h,y) = 0 \quad \forall (i,j)\in\mathcal{W} \tag{41}$$

$$\sum_{j\in\mathcal{P}_{ij}} h_j c_j(h,y) - c_p(h,y)Q_{ij}(h,y) \leq 0 \quad \forall (i,j)\in\mathcal{W}, p\in\mathcal{P}_{ij} \tag{42}$$

$$c_p(h,y) - \sum_{a\in\mathcal{A}} c_a(\Delta h,y)\delta_{ap} = 0 \quad \forall (i,j)\in\mathcal{W}, p\in\mathcal{P}_{ij} \tag{43}$$

$$\sum_{a\in\mathcal{A}} \psi_a(y_a) \leq B \tag{44}$$

$$h \geq 0 \quad y\geq 0 \tag{45}$$

Ywhere

expression	meaning	expression	meaning
41	flow conservation	44	budget
42	user equilibrium	$h\geq 0$	nonnegativity
43	definitional	$y\geq 0$	nonnegativity

The Set of Feasible Solutions

$$\Upsilon_0 \equiv \left\{ \begin{pmatrix} h \\ y \end{pmatrix} \geq 0 : (41), (42), (43), \text{ and } (44) \text{ hold} \right\}$$

1. Our formulation may be stated in succinct form as

$$\max \Delta CS\left(h,y,u^0\right) = \sum_{(i,j)\in\mathcal{W}} \left[\oint_{u}^{u^0} Q_{ij}\left(x\right) dx_{ij}\right]_{u=g(h,y)} \quad \text{s.t.} \quad \begin{pmatrix} h \\ y \end{pmatrix} \in \Upsilon_0 \tag{46}$$

   which is clearly a *single level mathematical program* (SLMP).

2. We reiterate that constraints (42) are inherently nonconvex, making Ω a nonconvex set. Thus, SLMP is nonconvex regardless of the nature of the expenditure functions $\psi_a(y_a)$.
3. The single level formulation is an alternative to a bi-level formulation of equilibrium network design for which the inner (or lower level) problem is either a mathematical program or a variational inequality or some other problem whose solution is an user equilibrium.

The Elastic Demand Case

When demand is inelastic, constraints (41) and (42) become

$$\sum_{p \in \mathcal{P}_{ij}} h_p - Q_{ij}(u) = 0 \quad \forall (i,j) \in \mathcal{W} \tag{47}$$

$$\sum_{j \in \mathcal{P}_{ij}} h_j c_j(h,y) - c_p(h,y) Q_{ij}(u) \leq 0 \quad \forall (i,j) \in \mathcal{W}, p \in \mathcal{P}_{ij} \tag{48}$$

where the Q_{ij} are fixed for all $(i,j) \in \mathcal{W}$. Thus, model (46) yields the following formulation:

$$\min \sum_{(i,j) \in \mathcal{W}} \sum_{p \in \mathcal{P}_{ij}} c_p(h_p) h_p \quad \text{s.t.} \quad \begin{pmatrix} h \\ y \\ u \end{pmatrix} \in \Upsilon_1 \tag{49}$$

where

$$\Upsilon_1 = \left\{ \begin{pmatrix} h \\ y \\ u \end{pmatrix} : (43), \ (44), \ (47), \text{ and } (48) \text{ hold} \right\} \tag{50}$$

Remarks About Equilibrium Network Design

1. Note: (46) requires evaluation of a line integral, which is only possible if demand functions satisfy a symmetry restriction [see Jara-Diaz and Friesz (1982)] requiring all cross elasticities of demand be proportional.
2. Note: such a symmetry restriction is needed regardless of how the inner (lower level) equilibrium problem is formulated. As such, it would appear that (46) *cannot be solved* since we have no guidance about what path of integration to employ in evaluating the intrinsic consumers' surplus line integral.
3. Approximating the consumers' surplus line integral is a fool's errand. Jara-Diaz and Friesz (1982) show that the available techniques for approximating this line integral may involve unacceptably large errors and are without any theoretical foundation.
4. In practice, (46) is specialized to the case of fixed demand, for which the consumers' surplus line integral is dropped from the formulation.

Reformulations and Algorithms for MPECS/Network Design

1. heuristics
2. pattern search
3. abstract MPEC
4. alternative expression of the Nash equilibrium
5. probabilistic search
6. reaction functions/sensitivity analysis
7. gap functions

Form of Problem Considered by the EDO Heuristic

Suwansirikul and Friesz (1987) proposed the equilibrium decomposed optimization (EDO) heuristic algorithm. The problem form is

$$Z_a\left(y\right) \equiv c_a\left[f_a(y), y_a\right] f_a\left(y\right) + \theta \psi_a(y_a) \tag{51}$$

where $\theta \in \Re^1_+$ is a dual variable for the budget constraint

$$\sum_{a \in \mathcal{A}} \psi_a(y_a) \leq B$$

Take θ to be a known constant, so the problem of interest is:

$$\min Z(y) = \sum_{a \in \mathcal{A}} Z_a\left[f_a(y), y_a\right] \tag{52}$$

$$\sum_{p \in \mathcal{P}_{ij}} h_p = Q_{ij} \quad \forall\left(i,j\right) \in \mathcal{W} \tag{53}$$

$$f = \Delta h \tag{54}$$

$$h \geq 0 \tag{55}$$

$$y \geq 0 \tag{56}$$

Rationale for the EDO Heuristic, Slide 1

1. Note that for any arc $a \in \mathcal{A}$

$$\frac{\partial Z_a}{\partial y_a} = c_a \frac{\partial f_a(y)}{\partial y_a} + f_a(y) \frac{\partial c_a}{\partial y_a} + \theta \frac{\partial \psi_a}{\partial y_a}$$

   In general, the first two terms on the righthand side cannot be evaluated since $f_a(y)$ is not known explicitly.

2. The method is based on ignoring terms that cannot be evaluated.
3. In particular, we make the following approximation

$$c_a \frac{\partial f_a(y)}{\partial y_a} + f_a(y) \frac{\partial c_a}{\partial y_a} + \theta \frac{\partial \psi_a}{\partial y_a} \approx \theta \frac{\partial \psi_a}{\partial y_a} + f_a(y) \left[\frac{\partial c_a}{\partial y_a} \right]_f \qquad (57)$$

   where

$$\left[\frac{\partial c_a}{\partial y_a} \right]_f$$

   indicates that the partial derivative $\partial c_a / \partial f_a$ is calculated holding all flow variables fixed.

Detailed Statement of The EDO Heuristic, Slide 1

Step 0. Initialization. Determine a closed form approximation of the gradient by forming:

$$Z_a' = \theta \frac{\partial \psi_a}{\partial y_a} + f_a \left[\frac{\partial c_a}{\partial y_a} \right]_f$$

for all arcs $a \in \mathcal{I}$, where $\mathcal{I}$ denotes the list of arcs being considered for improvement. Let $\mathcal{S} \subseteq \mathcal{I}$ denote the set of arcs whose optimal improvements have been determined. Select vectors $L^0 \in \Re_+^{|\mathcal{I}|}$ and $U^0 \in \Re_+^{|\mathcal{I}|}$ such that

$$\begin{aligned} Z_a'(L^0) &< 0 \\ Z_a'(U^0) &> 0 \end{aligned}$$

for each arc $a \in \mathcal{I}$. Set $\mathcal{S} = \emptyset$. Set $j = 1$.Select $\varepsilon \in \Re_{++}^1$.

Detailed Statement of The EDO Heuristic, Slide 2

Step 1. Calculate user equilibrium. Using the costs

$$\begin{array}{rcl} c_a(f_a) & a \in & \mathcal{A} \backslash (\mathcal{I} \cup \mathcal{S}) \\ c_a(f, y^*) & a \in & \mathcal{S} \\ c_a(f, y^j) & a \in & \mathcal{I} \end{array}$$

where

$$y^j = \frac{L^{j-1} + U^{j-1}}{2} \in \Re_+^{|\mathcal{I}|}$$

find the user equilibrium flow vector $f(y^j)$.

Detailed Statement of The EDO Heuristic, Slide 3

Step 2. Perform line search. For each arc $a \in \mathcal{I}$, do the following:

(i) If $Z'_a\left(y^j\right) < 0$, set $L^j_a = y^j_a$ and $U^j_a = U^{j-1}_a$.

(ii) If $Z'_a > 0$, set $L^j_a = L^{j-1}_a$ and $U^j_a = y^j_a$.

(iii) If $Z'_a = 0$, record the approximate optimal solution $y^*_a = y^j_a$, add arc a to $\mathcal{S}$, and remove arc a from $\mathcal{I}$.

Step 3. Stopping test. For each arc $a \in \mathcal{I}$, check whether $U^j_a - L^j_a \leq \varepsilon$. Remove each such arc from $\mathcal{I}$, and add it to $\mathcal{S}$, while recording its approximate optimal value $y^*_a = 1/2\left(L^j_a + U^j_a\right)$. If the improvement set $\mathcal{I}$ is not empty, set $j = j + 1$ and go to Step 1. Otherwise stop.

The Abstract MPEC, Again

Let us now introduce algorithms for the abstract MPEC:

$$\min_{y} F(x, y) \tag{58}$$

subject to

$$x \in X(y) \subseteq \Re^n \tag{59}$$

$$y \in Y(x) \subseteq \Re^q \tag{60}$$

$$[G(x, y)]^T (x' - x) \geq 0 \quad \forall x' \in X \tag{61}$$

NCP-Based Formulation of MPEC, Slide 1

Recall that the variational inequality (32) may be restated as a nonlinear complementarity problem (NCP):

$$
\left.
\begin{array}{c}
[G(x,u,y)]^T \begin{pmatrix} x \\ u \end{pmatrix} = 0 \\
\\
G(x,u,y) \geq 0 \\
x \geq 0 \\
u \geq 0
\end{array}
\right\} \iff \left\{
\begin{array}{c}
[G(x,u,y)]^T \begin{pmatrix} x \\ u \end{pmatrix} \leq 0 \\
\\
G(x,u,y) \geq 0 \\
x \geq 0 \\
u \geq 0
\end{array}
\right.
$$

where $G(.,.,.)$ is the function created from $F(.,.)$ and the constraints generating X while u is a vector of dual variables associated with those constraints when the NCP restatement is made; the regularity conditions for and details of such a conversion are found in Facchinei and Pang (2003).

NCP-Based Formulation of MPEC, Slide 2

As a consequence of the preceding remarksm the MPEC becomes

$$\left.\begin{array}{c} \min_y F(x,y) \\ \\ [G(x,u,y)]^T \begin{pmatrix} x \\ u \end{pmatrix} \leq 0 \\ \\ G(x,u,y) \geq 0 \\ x \geq 0 \\ u \geq 0 \\ y \in Y \end{array}\right\} \qquad (62)$$

This is us a single-level re-formulation of MPEC, and may be solved by conventional numerical methods for nonconvex mathematical programming problems; of course, applying such algorithms to (62) will not find a global optimum with certainty.

The Gap Function Approach, Slide 1

1. The gap function perspective provides another means for creating a single-level mathematical program whose solutions are also MPEC solutions.
2. In particular, if we replace the variational inequality constraint with an appropriate gap function, we obtain a single-level mathematical program with an additional equality constraint of a special structure.
3. For a given $y \in Y$, using the Fukushima-Auchmuty gap function, the variational inequality subproblem

$$x = \arg VI\left(F, X, y\right)$$

(32), may be replaced by

$$\zeta_\alpha\left(x, y\right) = 0$$

where α is a positive constant and

$$\zeta_\alpha\left(x, y\right) = \max_{z \in X}\left\{\left[F(z, y)\right]^T\left(x - z\right) - \frac{\alpha}{2}\left\|x - z\right\|^2\right\}$$

The Gap Function Approach, Slide 2

1. Using the Fukushima-Auchmuty gap function leads to a single-level reformulation of MPEC (29) – (32):

$$\left.\begin{array}{c} \min_y F(x,y) \\ \\ \zeta_\alpha(x,y) = 0 \\ \\ x \geq 0 \\ x \in X \\ y \in Y \end{array}\right\} \tag{63}$$

2. Meng et al. (2001) were among the first to note the usefulness of gap functions in formulating and solving MPECs.

Consider the following form of MPEC:

$$\min_{y} F(x, y) \tag{64}$$

subject to

$$\begin{aligned} L &\leq y \leq U & (65) \\ y &\in \Re^q & (66) \\ x &\in X(y) \subseteq \Re^n & (67) \\ [G(x, y)]^T (x' - x) &\geq 0 \quad \forall x' \in X(y) & (68) \end{aligned}$$

where $L \in \Re^q_+$ and $U \in \Re^q_+$ are known fixed vectors. Friesz et al. (1990) proposed and tested a family of heuristic algorithms for the above MPEC, that employ variational inequality sensitivity analysis to approximate the reaction function $x(y)$.

Sensitivity Analysis of VIs

1. There is a highly evolved theory of sensitivity analysis of NLPs, due to Fiacco and McCormick (1968), when parameters are perturbed.
2. This theory has been successfuly generalized to address VIs based on differentialble constraint functions and continuous VI mappings.
3. Allows local linear approximation of $x(y)$ when y is considered to be a parameter of

$$[G(x,y)]^T(x'-x) \geq 0 \quad \forall x' \in X(y)$$

Step 0. Initialization. Determine an initial solution y^0; set $k = 0$. Select a value of β.

Step 1. Solve variational inequality. Solve

$$\left[G\left(x, y^k\right)\right]^T \left(x' - x\right) \geq 0 \quad \forall x' \in X$$

Step 2. Calculate derivatives. Using sensitivity analysis approximate

$$\nabla_y x^k(y^k) = \left(\frac{\partial x_j\left(y^k\right)}{\partial y_i} : i \in [1, q], j \in [1, n]\right)$$

Step 3. Calculate direction. Calculate

$$d^k = \nabla_y M(y^k) = \left(\frac{\partial M\left(y^k\right)}{\partial y_i} : i \in [1, q] \right)$$

where

$$\begin{aligned} M(y) &= F[x(y), y] \\ \frac{\partial M}{\partial y_i} &= \sum_{j=1}^{n} \frac{\partial F}{\partial x_i} \frac{\partial x_j}{\partial y_i} + \frac{\partial F}{\partial y_i} \qquad i \in [1, q] \end{aligned}$$

Step 4. Determine step size. Calculate

$$\alpha_k = \frac{\beta}{k+1} \tag{69}$$

Step 5. Updating and stopping. Calculate

$$y^{k+1} = \left[y^k - \alpha_k \nabla_y M(y^k) \right]_L^U \tag{70}$$

where the righthand side of (70) is a projection of each $y_i^k - \alpha_k \left[\nabla_y M(y^k) \right]_i$ onto the real interval $[L_i, U_i]$. If

$$\left\| y^{k+1} - y^k \right\| \leq \varepsilon$$

stop and declare

$$y^* \approx y^{k+1}$$

Otherwise, set $k = k + 1$ and go to Step 1.

Part IV. Congestion Pricing

Preview

1. Unless otherwise stated, *dynamic congestion pricing* will refer to vehicular networks.
2. Dynamic congestion pricing for vehicular networks is already a reality.
3. Dynamic congestion pricing is presently largely ad hoc.
4. The basic theory of dynamic congestion pricing is simple.
5. To do dynamic congestion pricing correctly, you must know how to calculate dynamic user equilibria.
6. New advances in computing dynamic user equilibria make numerical dynamic congestion pricing nearly practical.
7. Prediction: within 5 to 10 years, medium and large scale numerical dynamic congestion pricing will be possible.

Types of Congestion Pricing

- First best congestion pricing
- Second best congestion pricing

First vs. Second Best Congestion Pricing

- Features of first best dynamic congestion pricing:

$$\textit{DUE flow with tolls} = \textit{DSO flow without tolls}$$

- Second best dynamic congestion pricing:

$$\max J = \text{net present value of societal benefits}$$

subject to

DUE flows, including flow conservation

toll bounds

- The second best problem is an infinite dimensional mathematical program with equilibrium constraints, which we refer to as an H-MPEC since most continuous time applications may be expressed in **H**ilbert space.

Dynamic System Optimization (DSO)

- DSO is a type of DTA that minimizes the total system travel cost

$$\min \ J = \sum_{p \in \mathcal{P}} \int_{t_0}^{t_f} \Psi_p(t, x)\, h_p(t)\, dt$$

where $\Psi_p(t, x)$ is an effective path delay operator obtained from a network loading model.

- The above is subject to the following now familiar constraints

$$\begin{aligned} \sum_{p \in \mathcal{P}_{ij}} \int_{t_0}^{t_f} h_p(t)\, dt &= Q_{ij}, \ \forall (i, j) \in \mathcal{W} \\ h_p(t) &\geq 0, \quad \forall p \in \mathcal{P}_{ij}, \ (i, j) \in \mathcal{W} \end{aligned}$$

- Note that the DSO is easily re-stated as an optimal control problem via its isoperimetric constraints ...

The Dynamic Efficient Toll Problem = First Best Congestion Pricing

- The idea behind the first best pricing is that a toll is attached to every link; the value of that toll is exactly what is needed to cause the tolled DUE problem to have the same solution as the untolled DSO problem.
- The tolled effective delay operator is:

$$\theta_p(t, x, y_p) = D_p(t, x) + F\left[t + D_p(x, t) - T_A\right] + y_p(t) \quad \forall p \in P$$
$$\theta_p(t, x, y_p) = \Psi_p(t, x) + y_p(t)$$

- To make the tolls meaningful

$$y_p(t) \geq 0 \; \forall t \in [t_0, t_f], \;\; p \in P$$

Analysis of First Best Congestion Pricing

- The aforementioned version of DSO is a time-shifted optimal control problem whose necessary conditions are analyzed in Friesz et al. (2004).
- We want

$$h^{DUE}(t) = h^{DSO}(t)$$

- Comparing *DUE* and *DSO* necessary conditions, one obtains the dynamic first best pricing rule:

$$y_p^{DUE}(t) = \left[\frac{\partial \Psi_p \left(t, x^{DSO}\right)}{\partial h_p} h_p^{DSO} \right]^+ \qquad \forall t \in [t_0, t_f]$$

- This result is analogous to the static case (Hearn and Yildrim, 2002), and hardly surprising.

Conceptual Form of the Dynamic Second Best Congestion Pricing Problem

$$\min \text{ present value of total congestion costs}$$

subject to

flows form a dynamic user equilibrium (DUE)

upper and lower bounds on tolls

The Dynamic Optimal Toll Problem with Equilibrium Constraints (DOTPEC)

- The DOTPEC is an instance of the so-called infinite dimensional mathematical program with equilibrium constraints (H-MPEC).
- The DOTPEC is the natural form of the dynamic optimal toll problem with equilibrium constraints.

VI-Based Statement of the DOTPEC

$$\min_{y} J = \int_{t_0}^{t_f} \sum_{p \in \mathcal{P}} \Psi_p\left(t, x\right) h_p\left(t\right) dt$$

subject to

$$0 \leq y_{a_i} \leq M_{a_i} \qquad \forall p \in \mathcal{P}, i \in \left\{1, 2, ..., m\left(p\right)\right\}$$

$$\text{DUE Flows:} \quad \sum_{p \in \mathcal{P}} \int_{t_0}^{t_f} \theta_p\left[t, x\left(h\right), y_p\right]\left(w_p - h_p\right) dt \geq 0 \quad \forall w \in \Lambda$$

where M_{a_i} is the upper bound on tolls for arc a_i, for all $i \in \mathcal{A}$, and

$$\Lambda = \left\{ h : \sum_{p \in P_{ij}} \int_{t_0}^{t_f} h_p\left(t\right) dt = Q_{ij}, \;\; h_p\left(t\right) \geq 0, \; \forall \left(i, j\right) \in \mathcal{W} \right\}$$

Alternative Forms of the DOTPEC

The DOTPEC has several alternative forms due to different ways of expressing the lower level DUE subproblem; these include:

- DVI
- user equilibrium constraints
- NCP
- MiCP
- Gap Function
- other

DVI-Based Form of the DOTPEC

- To express the DOTPEC as a DVI, first restate the set of feasible flows as:

$$\Lambda_0 = \left\{ h \geq 0 : \frac{dy}{dt} = \sum_{p \in \mathcal{P}_{ij}} h_p(t)\, dt, \quad y_{ij}(t_0) = 0, \quad y_{ij}(t_f) = Q_{ij}\ \forall (i,j)$$

- Thereupon we have

$$\min_y J = \int_{t_0}^{t_f} \sum_{p \in \mathcal{P}} \Psi_p(t, x)\, h_p(t)\, dt$$

subject to

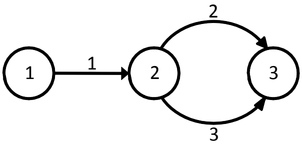
$$0 \leq y_{a_i} \leq M_{a_i} \qquad \forall p \in \mathcal{P}, i \in \left\{1, 2, ..., m(p)\right\}$$

$$\sum_{p \in \mathcal{P}} \int_{t_0}^{t_f} \theta_p\left[t, x(h), y_p\right](w_p - h_p)\, dt \geq 0 \quad \forall w \in \Lambda_0$$

- It is not hard to introduce elastic travel demand.

Numerical Example of Dynamic Congestion Pricing

- We consider a 3-arc, 3-node network.

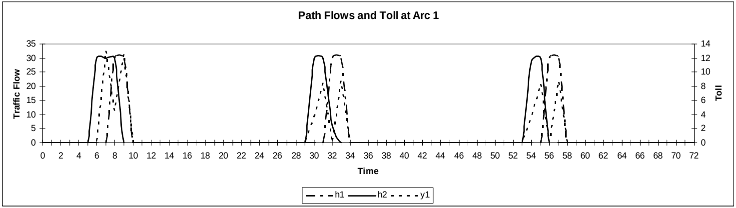


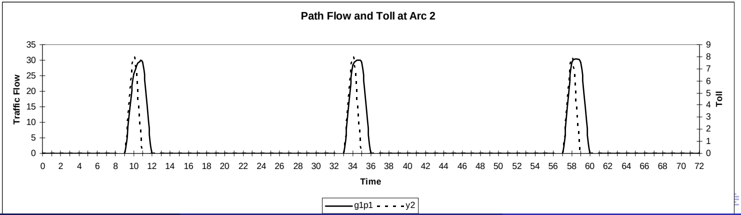
Arc name	From node	To node	Arc delay, $D_a(x_a(t))$
a_1	1	2	$2+\frac{1}{100}x_{a_1}$
a_2	2	3	$1+\frac{1}{150}x_{a_2}$
a_3	2	3	$3+\frac{1}{100}x_{a_3}$

and $\tilde{Q} = 150$, $\chi = 20000$, $s_{13} = 0.7$, $t_A = 12$, and

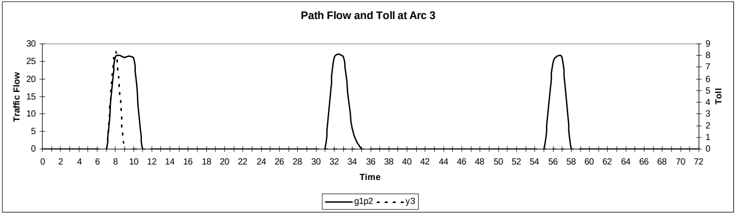
$$F\left[t + D_p(x,t) = t_A\right] = 5\left[t + D_p(x,t) = t_A\right]^2$$

Path flows and toll at each arc:

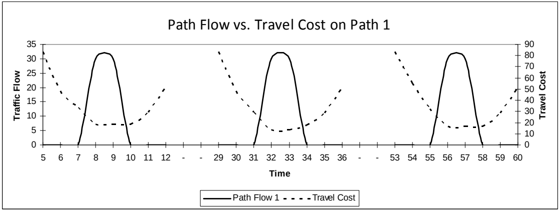


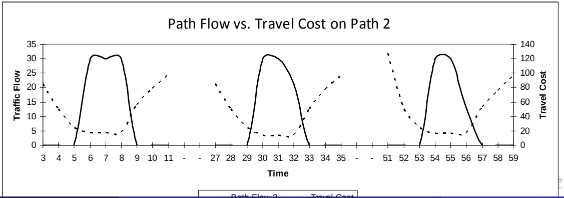


Path flows and toll at each arc:

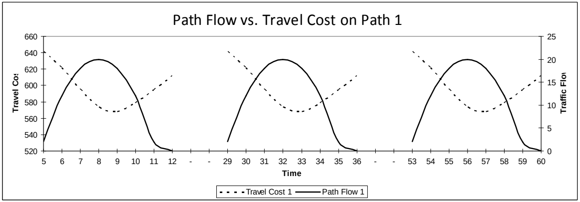


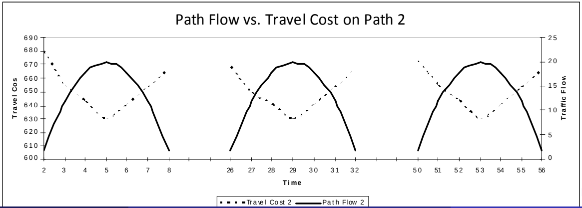
Comparison of path flow and associated unit travel costs on each path:





Comparison of path flow and associated unit travel costs on each path:





The results are

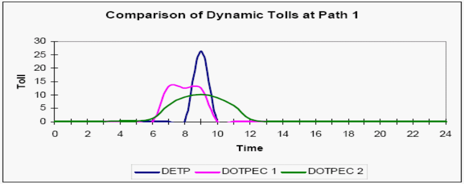


Figure 14: Comparison of Dynamic Tolls by DETP, DOTPEC solved by discrete time approximation (DOTPEC 1), and DOTPEC solved by descent in Hilbert spaces (DOTPEC 2) for path p_1

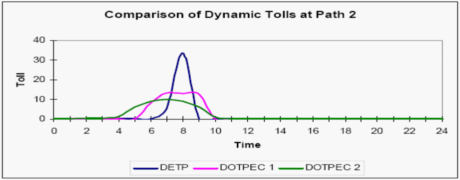


Figure 15: Comparison of Dynamic Tolls by DETP, DOTPEC solved by discrete time approximation (DOTPEC 1), and DOTPEC solved by descent in Hilbert spaces (DOTPEC 2) for path p_2

- Comparison of algorithms
- Less restrictive regularity conditions for convergence
- shape of the DUE delay operators
- Convergence for DUE method and DMPEC method
- Existence for DUE
- Introduction of stochasticity
- introduction of feedback
- rejection of demand
- robust games and robut optimal control
- realistic extension to data networks (the Internet)

Concluding Remarks

- DUE flows are computable
- Hence, the DOTPEC is "computable" – just barely with a desktop, single processor
- My prediction: DOTPEC will be routinely solved for medium to large networks within 5 to 10 years

Last Slide

Thanks !!!

Questions ???