Design for Optimizability
A Case Study in Routing

Mung Chiang
Electrical Engineering Department, Princeton

DIMACS Workshop
November 12, 2009

Internet Routing and Traffic Engineering

Joint work with Dahai Xu and Jennifer Rexford

Most large IP networks run Interior Gateway Protocols in an Autonomous System

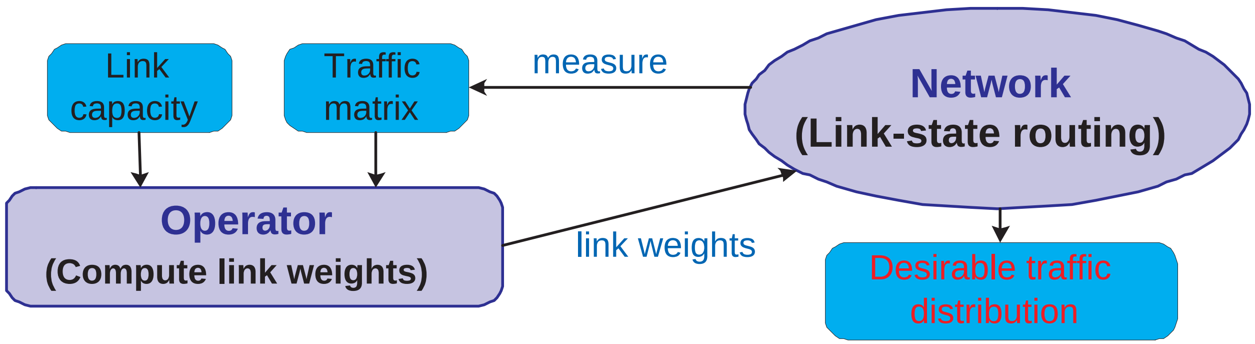
OSPF: a reverse shortest path method

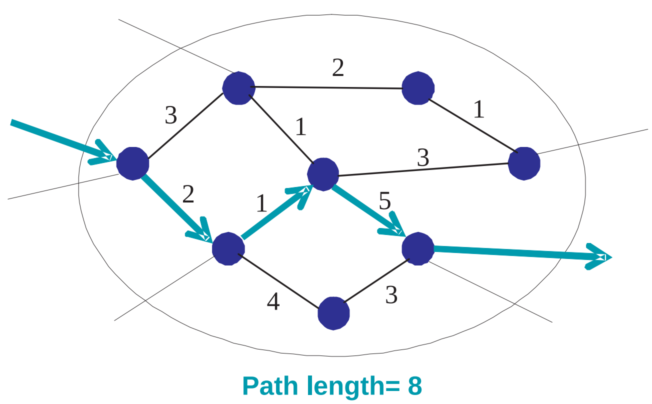
- Take in traffic matrix (constants)
- Vary link weights (variables)
- Hope to minimize sum of link cost function (objective)

3 components of link-state routing for traffic engineering

- Centralized computation for setting link weights
- Distributed way of using these link weights to split traffic
- Hop-by-hop, destination-based packet forwarding

Internet Routing and Traffic Engineering





History

- 1980s-1990s, intra-domain routing algorithms based on link weights
- 1990s, many variants of OSPF proposed and used: UnitOSPF, RandomOSPF, InvCapOSPF, L2OSPF
- Late 1990s, more complex MPLS protocols proposed. (Optimal benchmark: arbitrary splitting of flows on any links in any proportion), but they lose desirable features, eg, distributed determination of flow splitting and ease of management
- 2000, Fortz and Thorup presented local search methods to approximately solve the NP-hard problem in OSPF
- 2003, Sridharan, Guerin, and Diot proposed to select the subset of next hops for each prefix
- 2005, Fong, Gilbert, Kannan, and Strauss proposed to allow flows on non-shortest paths, but loops may be present and performance under multi-destination scenarios not clear
- 2007, Xu, Chiang, Rexford propose DEFT to almost achieve optimal traffic engineering

From OSPF to PEFT

Packet forwarding still destination-based and hop-by-hop

A new way to use link weights:

- Use link weights to compute path weights
- Split traffic on all paths
- Exponential penalty on longer paths

Leads to a new way to compute link weights

How good can the new protocol be?

How to compute link weights in the new protocol?

Problem Statement

Given directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$

Given capacity $c_{u,v}$ for each link (u, v)

Given $D(s, t)$: traffic demand from node s and destined to node t

Link cost function: $\Phi(f_{u,v}, c_{u,v})$ strictly increasing convex function of flow $f_{u,v}$ on link (u, v)

Objective 1: minimize $\max_{(u,v)} \frac{f_{u,v}}{c_{u,v}}$

Objective 2: minimize $\sum_{(u,v)\in\mathbf{E}} \Phi(f_{u,v}, c_{u,v})$

Traffic Splitting Function

$w_{u,v}$: weight for link (u, v)

d_u^t: shortest distance from node u to node t

$d_v^t + w_{u,v}$: distance from u to t when routed through v

$h_{u,v}^t = d_v^t + w_{u,v} - d_u^t$: gap

Link (u, v) is on the shortest path to t if and only if $h_{u,v}^t = 0$

f_u^t: incoming flow at node u for destination t

$f_{u,v}^t$: flow on link (u, v) for destination t

$$f_{u,v}^t = f_u^t \frac{\Gamma(h_{u,v}^t)}{\sum_{(u,j)\in\mathbb{E}} \Gamma(h_{u,j}^t)}$$

OSPF or PEFT

OSPF:

$$\Gamma_O(h^t_{u,v}) = \begin{cases} 1, & \text{if } h^t_{u,v} = 0 \\ 0, & \text{if } h^t_{u,v} > 0. \end{cases}$$

PEFT:

$$\Gamma_P(h^t_{u,v}) = \Upsilon^t_v e^{-h^t_{u,v}}$$

$$\Upsilon^t_u = \sum_{(u,v)\in\mathbb{E}} \left(e^{-h^t_{u,v}} \Upsilon^t_v \right)$$

Routers can direct traffic on non-shortest paths, with an exponential penalty on longer paths

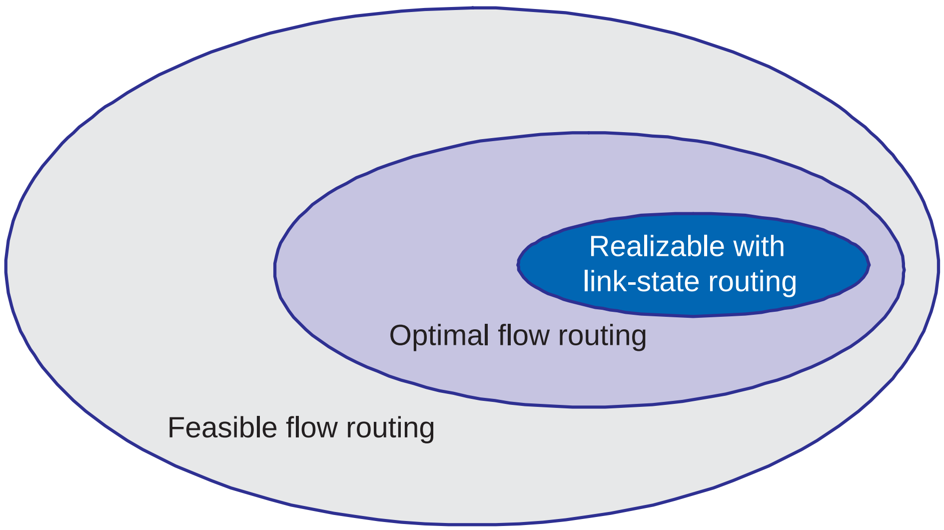
Simple Routing Can Be Optimal

Theorem: Link state routing and destination-based forwarding can achieve optimal traffic engineering

Theorem: Optimal weights can be computed by a convex optimization

Gradient algorithm solves the new link weight optimization problem 2000 times faster than local search algorithm for OSPF link weight computation

Solution Idea: Network Entropy Maximization



Constraint: flow conservation with effective capacity

Objective function: find one that picks out only link-state-realizable traffic distribution

Entropy function is the right choice, and the only one

Network Entropy Maximization

Entropy $z(x^i_{s,t}) = -x^i_{s,t} \log x^i_{s,t}$ for source-destination pair (s,t)

$$
\begin{array}{ll}
\text{maximize} & \sum_{s,t} \left(D(s,t) \sum_{P^i_{s,t}} z(x^i_{s,t}) \right) \\
\text{such that} & \sum_{s,t,i:(u,v)\in P^i_{s,t}} D(s,t) x^i_{s,t} \le \widetilde{c}_{u,v}, \forall (u,v) \\
 & \sum_i x^i_{s,t} = 1, \forall (s,t) \\
\text{variables} & x^i_{s,t} \ge 0.
\end{array}
$$

Characterization of optimality:

$$
\frac{x^{i*}_{s,t}}{x^{j*}_{s,t}} = \frac{e^{-(\sum_{(u,v)\in P^i_{s,t}} w_{u,v})}}{e^{-(\sum_{(u,v)\in P^j_{s,t}} w_{u,v})}}
$$

Link Weight Computation

Compute necessary capacities $\widetilde{\boldsymbol{c}}$ through multi-commodity flow problem
$\mathbf{w} \leftarrow$ Any set of link weights
$\boldsymbol{f} \leftarrow$ Traffic_Distribution($\mathbf{w}$)
while $\boldsymbol{f} \neq \widetilde{\boldsymbol{c}}$ **do**
    $\mathbf{w} \leftarrow$ Link_Weight_Update($\boldsymbol{f}$)
    $\boldsymbol{f} \leftarrow$ Traffic_Distribution($\mathbf{w}$)
end while
Return $\mathbf{w}$ /*final link weights*/

Link Weight Update Function

for each link (u, v) **do**
    $w_{u,v} \leftarrow w_{u,v} - \alpha\,(\widetilde{c}_{u,v} - f_{u,v})$
end for
Return new link weights $\mathbf{w}$

Traffic Distribution Function

For link weights $\mathbf{w}$, construct all-pairs shortest paths and compute $\Gamma_P(h^t_{u,v})$
for each destination t **do**
    Temporarily remove link (u,v) where $d^t_u > d^t_v$
    Do topological sorting on the residual network
    for each source $s \neq t$ in the decreasing topological order **do**
        $f^t_s \leftarrow D(s,t) + \sum_{x:(x,s)\in\mathbb{E}} f^t_{x,s}$
        $f^t_{s,v} \leftarrow f^t_s \frac{\Gamma_P(h^t_{s,v})}{\sum_{(s,j)\in\mathbb{E}} \Gamma_P(h^t_{s,j})}$
    end for
end for
$f_{u,v} \leftarrow \sum_{t\in\mathbb{V}} f^t_{u,v}$
Return $\boldsymbol{f}$ /*set of $f_{u,v}$*/

Simulation

Computational software:

Optimal benchmark: computed using CPLEX 9.1 via AMPL

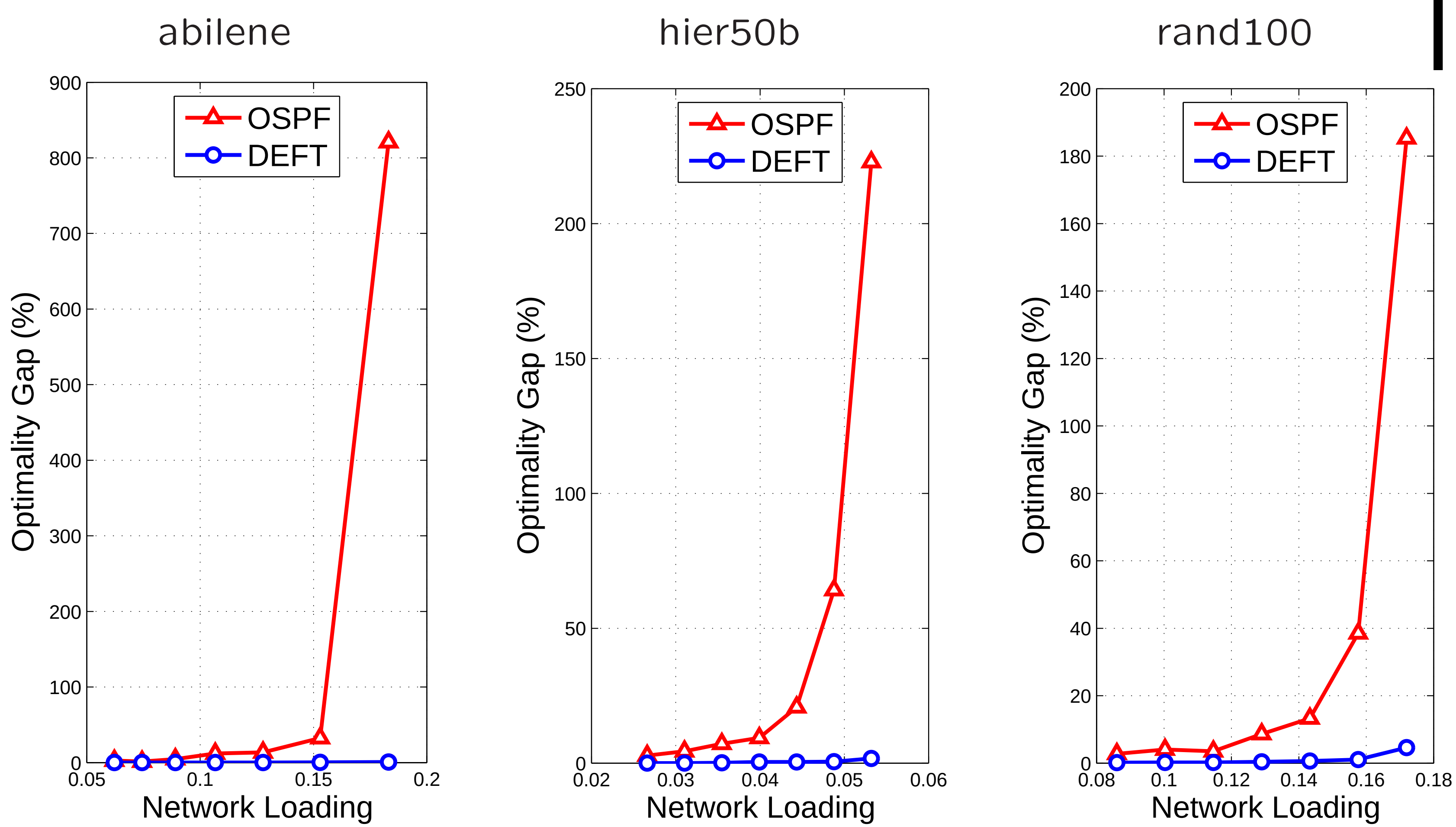
OSPF link weight by local search: Open source software project TOTEM 1.1 with IGP weight optimization

PEFT link weight: our algorithm

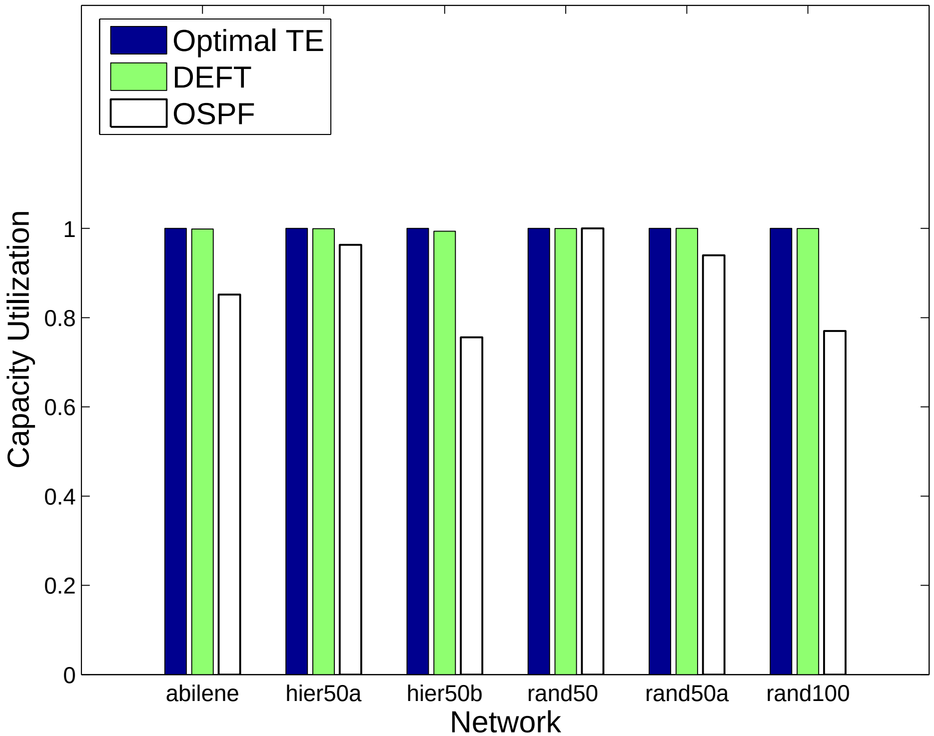
Topology and traffic matrices:

- Abilene on Nov. 15, 2005
- Those well-established in the community

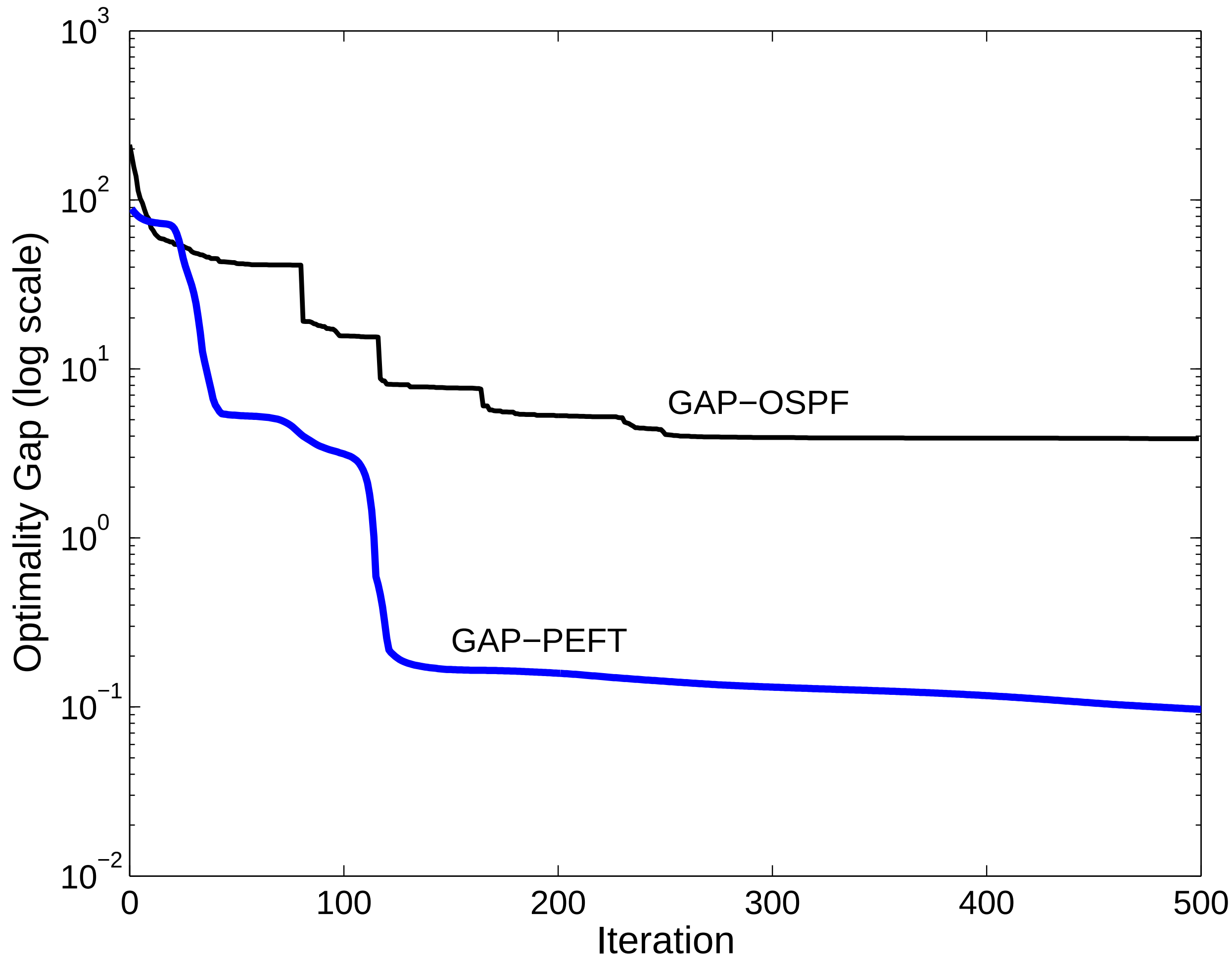
Optimality Gap Reduction



Capacity Improvement



Rate of Convergence



Running Time

Net. ID	Topology	Node #	Link #	Time per Iteration (second)	
				PEFT	OSPF
abilene	Backbone	11	28	0.002	6.0∼13.9
hier50a	2-level	50	148	0.006	6.0∼13.9
hier50b	2-level	50	212	0.007	6.4∼17.4
rand50	Random	50	228	0.007	3.2∼9.0
rand50a	Random	50	245	0.007	6.1∼14.1
rand100	Random	100	403	0.042	39.5∼105.1

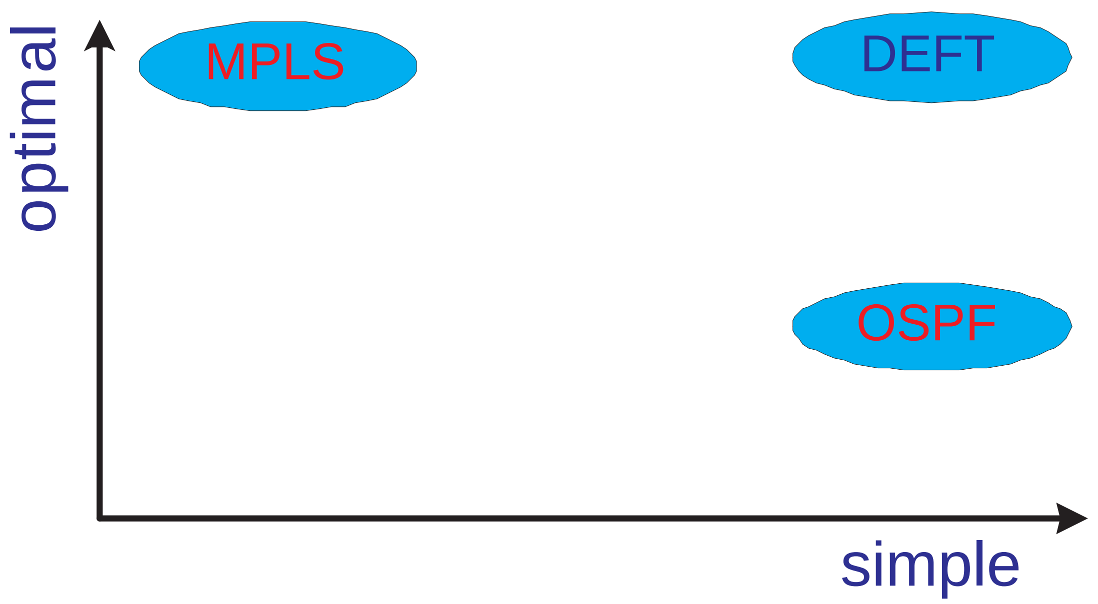
Optimality-Simplicity Tradeoff

	Commodity Routing	**Link-State Routing**	
		OSPF	**PEFT**
Traffic Splitting	Arbitrary	Even	Exponential
Scalability	Low	High	High
Optimal TE	Yes	No	Yes
Complexity Class	Convex Optimization	NP Hard	Convex Optimization

Optimality-Simplicity Tradeoff

Often there is a price for revisiting assumptions

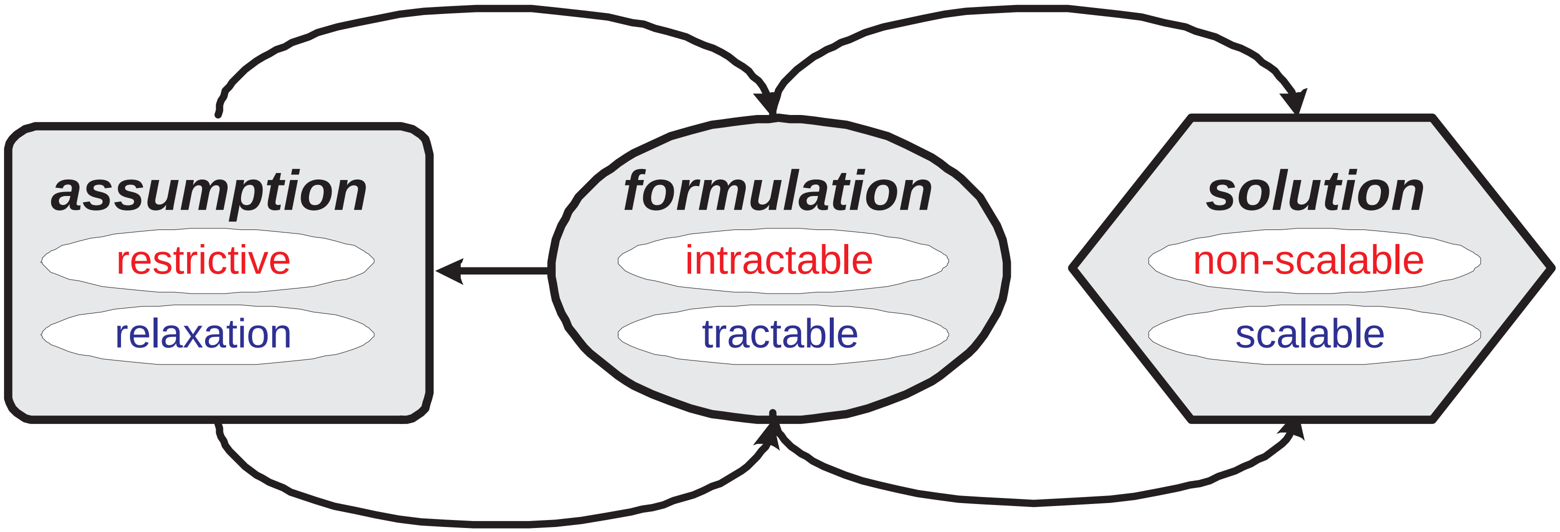
In Internet traffic engineering case, DFO provides an “nice’ tradeoff



NEM and NUM

	Congestion Control	**Traffic Engineering**
Traffic type	Elastic	Inelastic
Flow distribution	Fixed	Variable
Participants	End user and router	Operator and router
Timescale	Seconds	Hours
Framework	NUM	NEM
Multipliers	Feedback prices	Penalty weights
Implications	Stabilized TCP	Optimal LS routing

Feedback in Engineering Process



Contacts

chiangm@princeton.edu
www.princeton.edu/~chiangm