# Beyond Optimality: New Trends in Network Optimization

Mung Chiang
Electrical Engineering Department, Princeton

IEEE SAM Workshop
July 2008

## Optimization Beyond Optimality

Very different uses of optimization

- Standard answer: Computing (local, global) optimum

In fact, much more than that:

- I. Modeling: Resource allocation, fairness, reverse-engineering
- II. Architecture: who does what and how to connect
- III. Robustness to stochastic dynamics
- IV. Feedback to engineering assumptions
- V. Complexity-performance tradeoff

## What's Boring By Now

The following kind of results are no longer fresh:

- Dual decomposition of utility maximization
- Asymptotic convergence to the global optimum
- Convexity of the problem after log change of variable and approximations
- Session level stability under exponential filesize distribution

Let's move beyond these

## Nature of the Talk and Acknowledgement

Overview talk on key ideas and challenges

Minimize the amount of materials you can get simply from the publications, subject to the constraint of begin self-contained


- Co-authors of the papers mentioned here: A. R. Calderbank, R. Cendrillon, J. Doyle, P. Hande, J. Huang, J. Liu, S. H. Low, M. Moonen, H. V. Poor, A. Proutiere, S. Rangan, J. Rexford, D. Shah, A. Tang, D. Xu, Y. Yi, Z. Zhang
- Discussion: S. Boyd, D. Gao, J. He, B. Johansson, M. Johansson, F. P. Kelly, R. Lee, X. Lin, A. Ozdaglar, P. Parrilo, N. Shroff, R. Srikant, T. Lan
- Industry collaborators from: AT&T, Alcatel-Lucent, Qualcomm Flarion Technologies, Marvell

# Part I

## Modeling Resource Allocation

## Modeling

The mathematical language for constrained decision making

- Design freedoms (variable)
- Given parameters (constants)
- Goals (objective function)
- Constraints (constraint set)

Impacts demonstrated in commercial systems (3 cases in this talk):

- DSL broadband access networks
- Cellular wireless networks
- Internet backbone networks

## Objective Function

- $\sum_i C_i$: cost function that can depend on all degrees of freedom

- $\sum_i U_i$: utility function that can depend on throughput, delay, energy

Often increasing, concave, smooth, but doesn't have to be

Efficiency

Elasticity

User satisfaction

Fairness

## Objective: Fairness

- $x$ is $\alpha$-fair if, for all other feasible $y$:

$$\sum_s \frac{y_s - x_s}{x_s^\alpha} \leq 0$$

- Include special cases such as maxmin fair, proportional fair (Kelly97), throughput max, delay min...

- Maximizing $\alpha$-fair utility functions lead to optimizers that are $\alpha$-fair (MoWalrand00):

$$U^\alpha(x) = x^{1-\alpha}/(1-\alpha), \alpha \neq 1, \text{and} = \log x, \alpha = 1$$

What about suboptimal solutions?

From Optimality gap $\Delta(\mathbf{x})$ to Fairness gap $\beta(\mathbf{x})$?

## Modeling Beyond Performance

- Availability (XuLiChiangCalderbank07)
- Anonymity (SuhasHuangXuChiang07)
- Integrity, confidentiality, non-repudiation

- Scalability
- Manageability
- Evolvability

## Constraints

1. Inelastic, individual QoS constraints

2. Technological and regulatory constraints

3. Feasibility constraints

- Capacity region (information theory)
- Stability region (queuing theory)
- Achievability region under particular physical phenomena

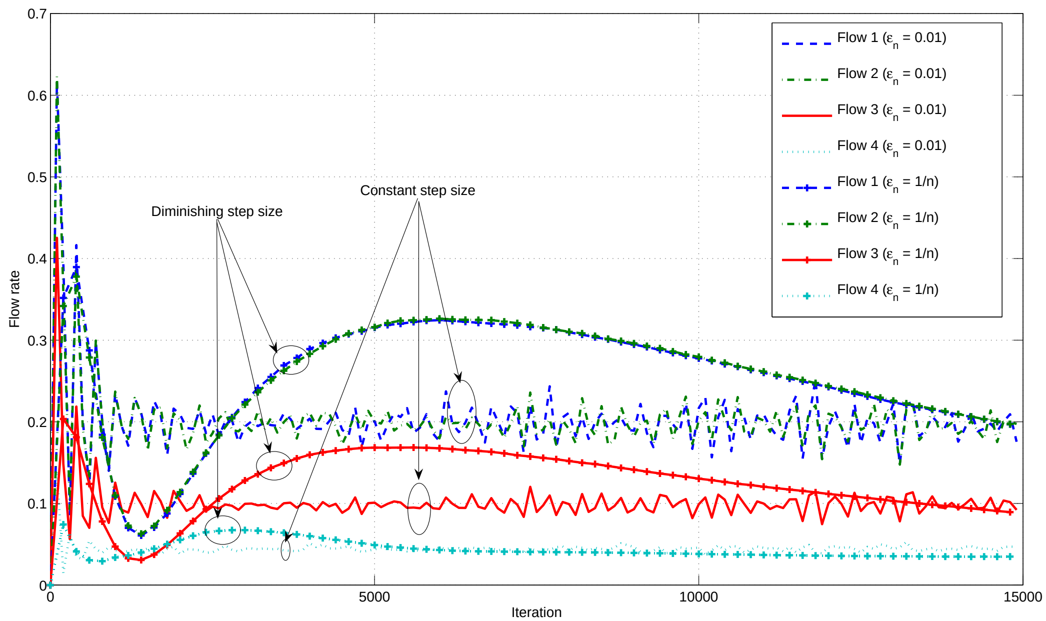
## Constraints: Resource Competition and Allocation

| | Congestion | Collision | Interference |
|---|---|---|---|
| *Constraint* | $x + y \leq 1$ | $x + y \leq 1, x, y \in \{0, 1\}$ | $x/y \leq 1$ |
| *Freedom* | Source rate | Transmit time | Transmit power |
| *Early work* | Jacobson 1988 | Aloha 1970s | Qualcomm 1980s |
| *Key framework* | Kelly 1998 | TE 1992 | Foschini 1993 |
| *Optimization* | max $U(\mathbf{x})$<br>s.t. $\mathbf{Ax} \leq \mathbf{c}$ | max $\boldsymbol{\mu}^T \mathbf{R}$<br>s.t. $\mathbf{R} \in \mathcal{R}$ | min $\mathbf{1}^T \mathbf{p}$<br>s.t. SIR$(\mathbf{p}) \geq \boldsymbol{\gamma}$ |
| *Main method* | Primal-dual update | Max weight match | Fixed point update |

## Feedback in Networks

| | Congestion | Collision | Interference |
|---|---|---|---|
| *Implicit* | Loss, delay in TCP | Collision in contention MAC | SIR |
| *Explicit* | ECN, XCP, RCP | Queue length | Load spillage |
| *Limited* | Some recent works | A lot of works | Not much |

# Stochastic Noisy Feedback




Convergence properties when feedback suffers packet level corruption (ZhangZhengChiang07)

## Modeling By Reverse Engineering

Optimization of network or by network

Given a solution, what is the problem?

Forward engineering also carried out

## Summary of Reverse Engineering

- TCP congestion control

One protocol: Basic NUM (LowLapsley99, RobertsMassoulie99, MoWalrand00, YaicheMazumdarRosenberg00, KunniyurSrikant02, LaAnatharam02, LowPaganiniDoyle02, Low03, Srikant04...)

Multiple protocols: Nonconvex equilibrium problem (TangWangLowChiang05,06)

- IP routing:

Inter-AS routing: Stable Paths Problem (GriffinSheperdWilfong02)

- MAC backoff contention resolution: Non-cooperative Game (LeeChiangCalderbank06)

## Modeling of Topology

- Optimization-based model of network functionality on top of random-graph models (Li Alderson Doyle Willinger 2004)
- Explanatory, rather than descriptive

A "dual" direction in Part III

# Part II

## Quantifying Architecture

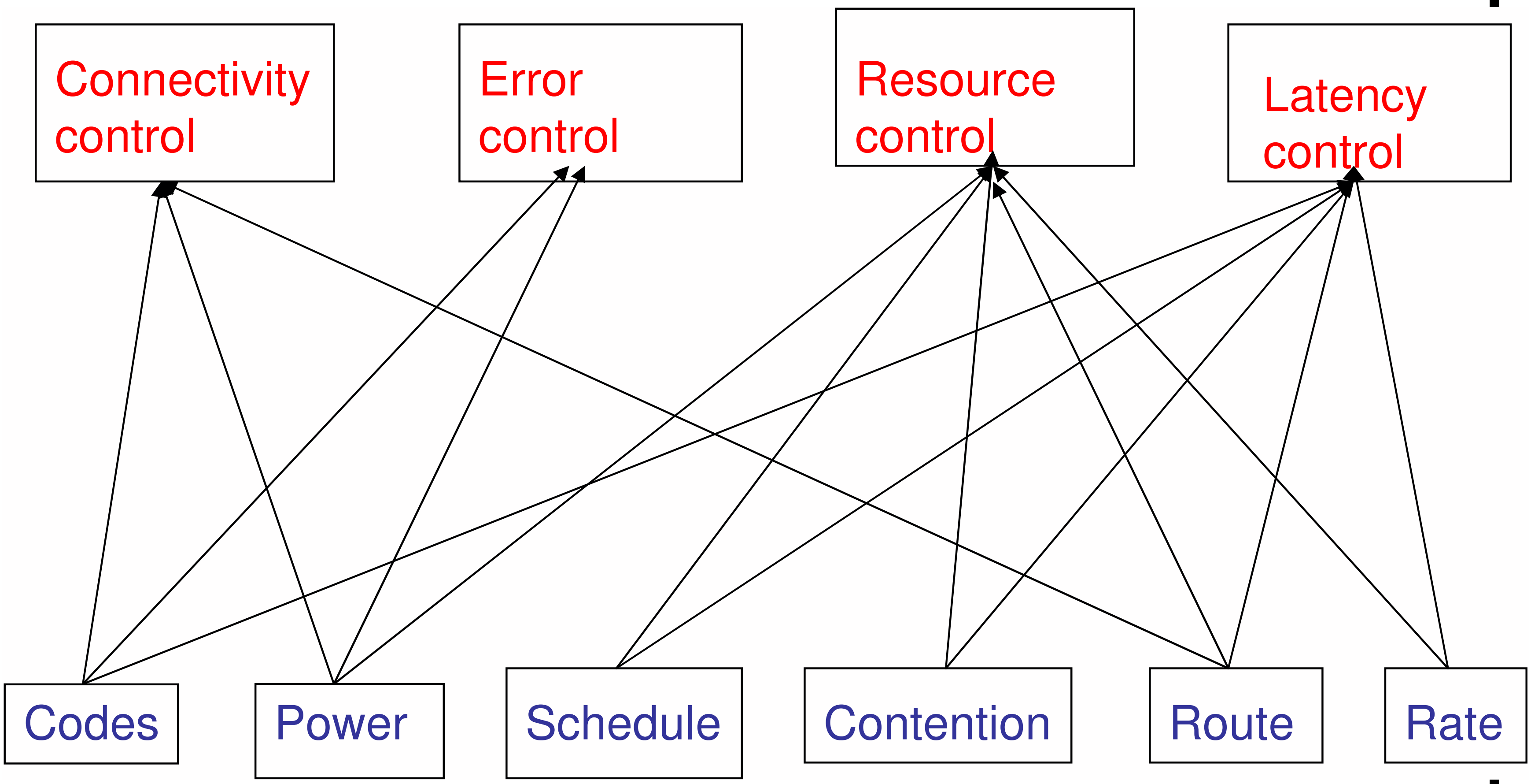
## Architecture: Functionality Allocation
## Who Does What and How to Connect Them

How to contain error?

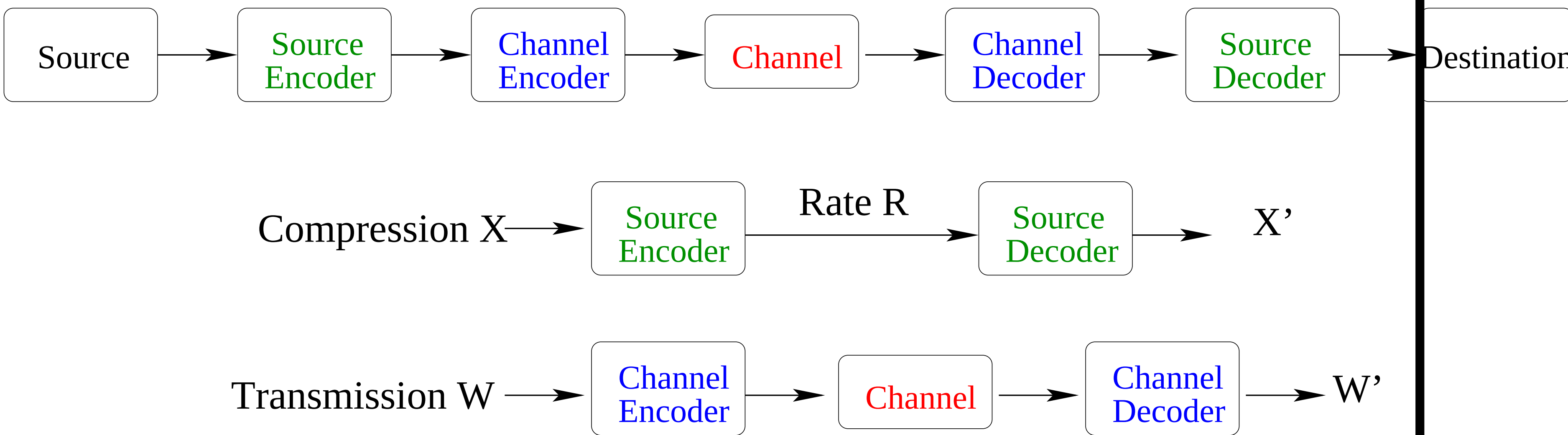
How to resolve bottleneck?

Which stock to buy: Microsoft, Cisco, Qualcomm?

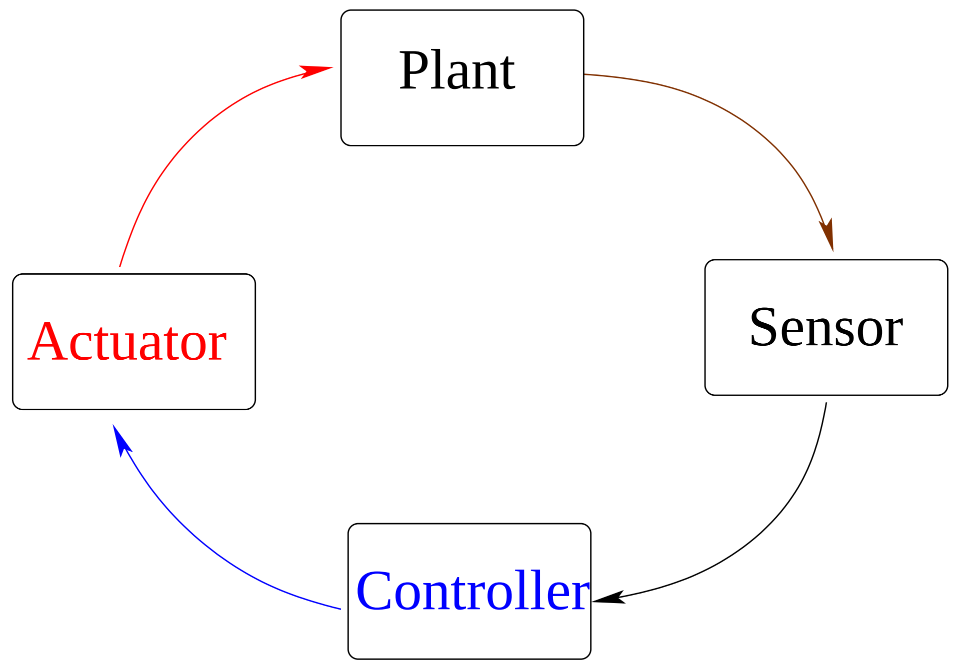
## Some Examples of Functionalities and Freedom



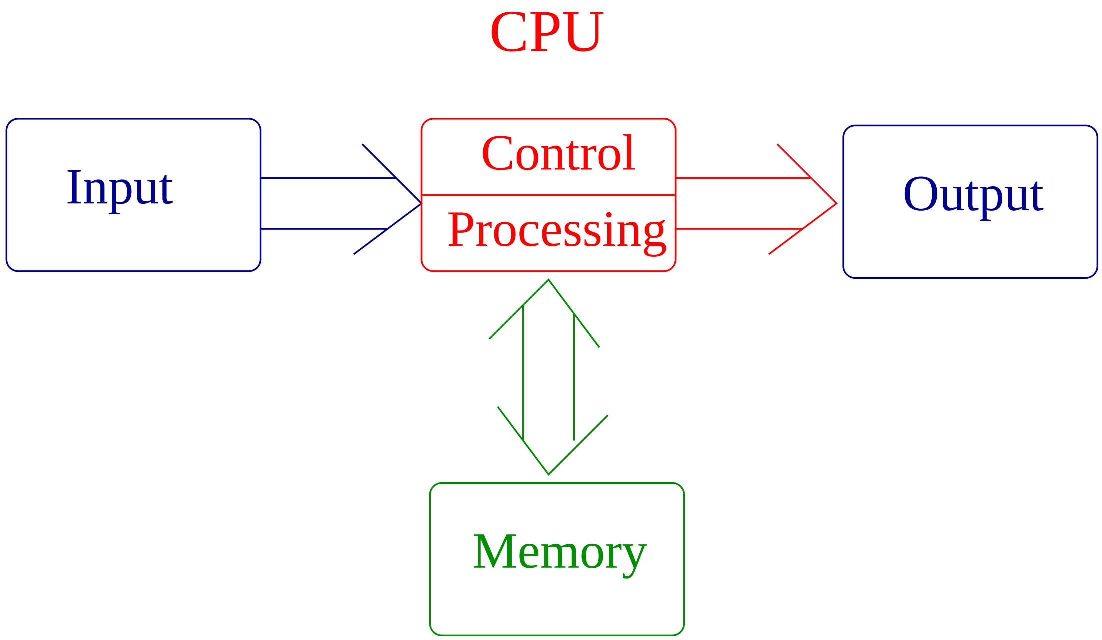
## Architecture in Communication: Well-established



## Architecture in Control: Well-established



## Architecture in Computation: Well-established



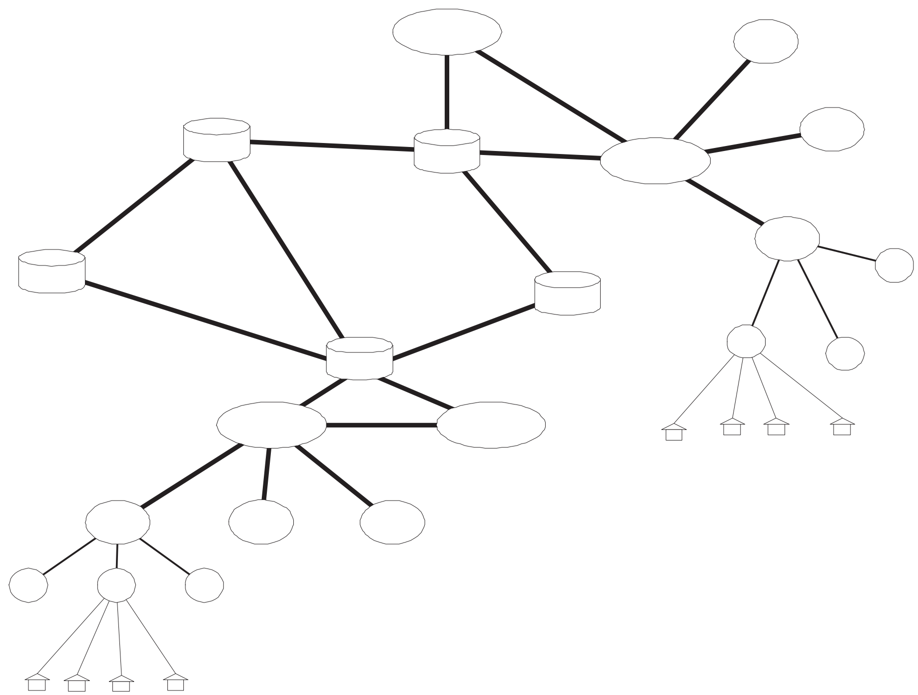
# Architecture in Networking: Not Sure

Layer or not layer?

| |
|---|
| Application |
| Presentation |
| Session |
| Transport |
| Network |
| Link |
| Physical |

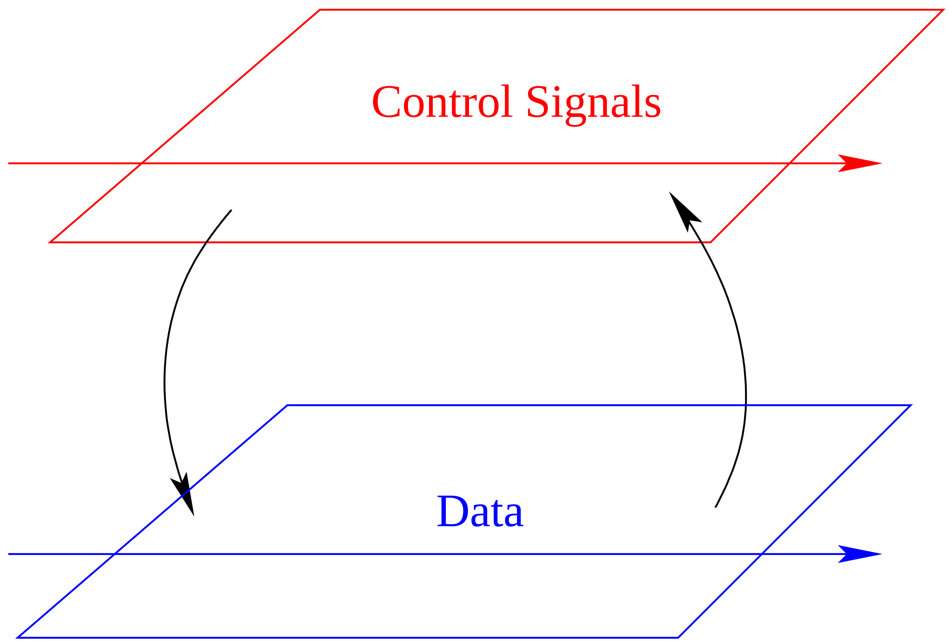
# Architecture in Networking: Not Sure

End-to-end or in-network?



# Architecture in Networking: Not Sure

Control plane or data plane?



## Math Foundation for Network Architecture

Layering As Optimization Decomposition

Network: Generalized NUM

Layering architecture: Decomposition scheme

Layers: Decomposed subproblems

Interfaces: Functions of primal or dual variables

Horizontal and vertical decompositions through

- implicit message passing (e.g., queuing delay, SIR)
- explicit message passing (local or global)

3 Steps: G.NUM $\Rightarrow$ A solution architecture $\Rightarrow$ Alternative architectures

## Two Cornerstones for Conceptual Simplicity

Networks as optimizers

We've seen this in Part I

Layering as decomposition

Common language for comparing architectural alternatives

Suboptimality is fine, as long as architecture is "right"

Survey of key messages, methods, and open problems in

Proceedings of the IEEE: ChiangLowCalderbankDoyle07

## Decomposition

Standard techniques of optimization decomposition:

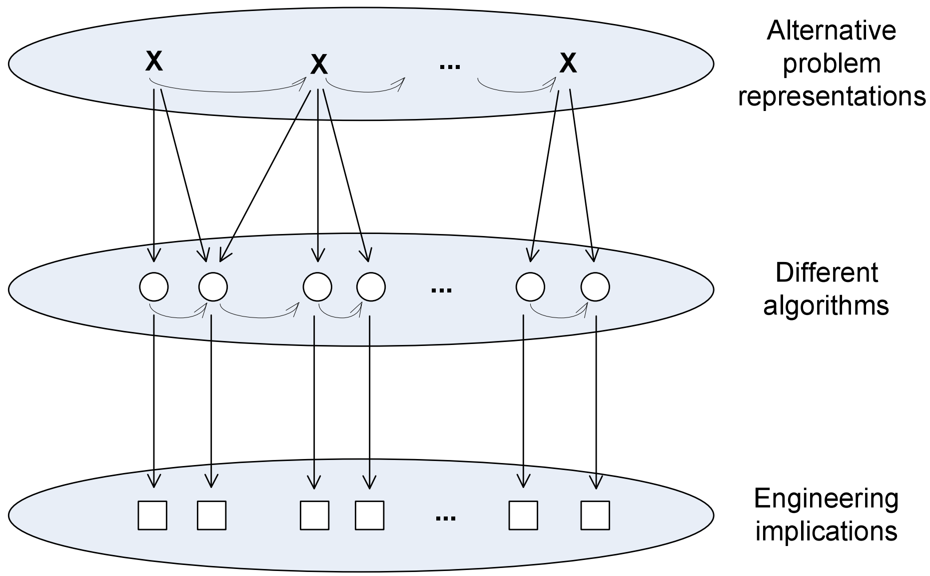
- Dual decomposition (most widely used today)
- Primal decomposition
- Primal penalty function approach

There're various combinations:

- Hierarchical
- Partial
- Timescale choices

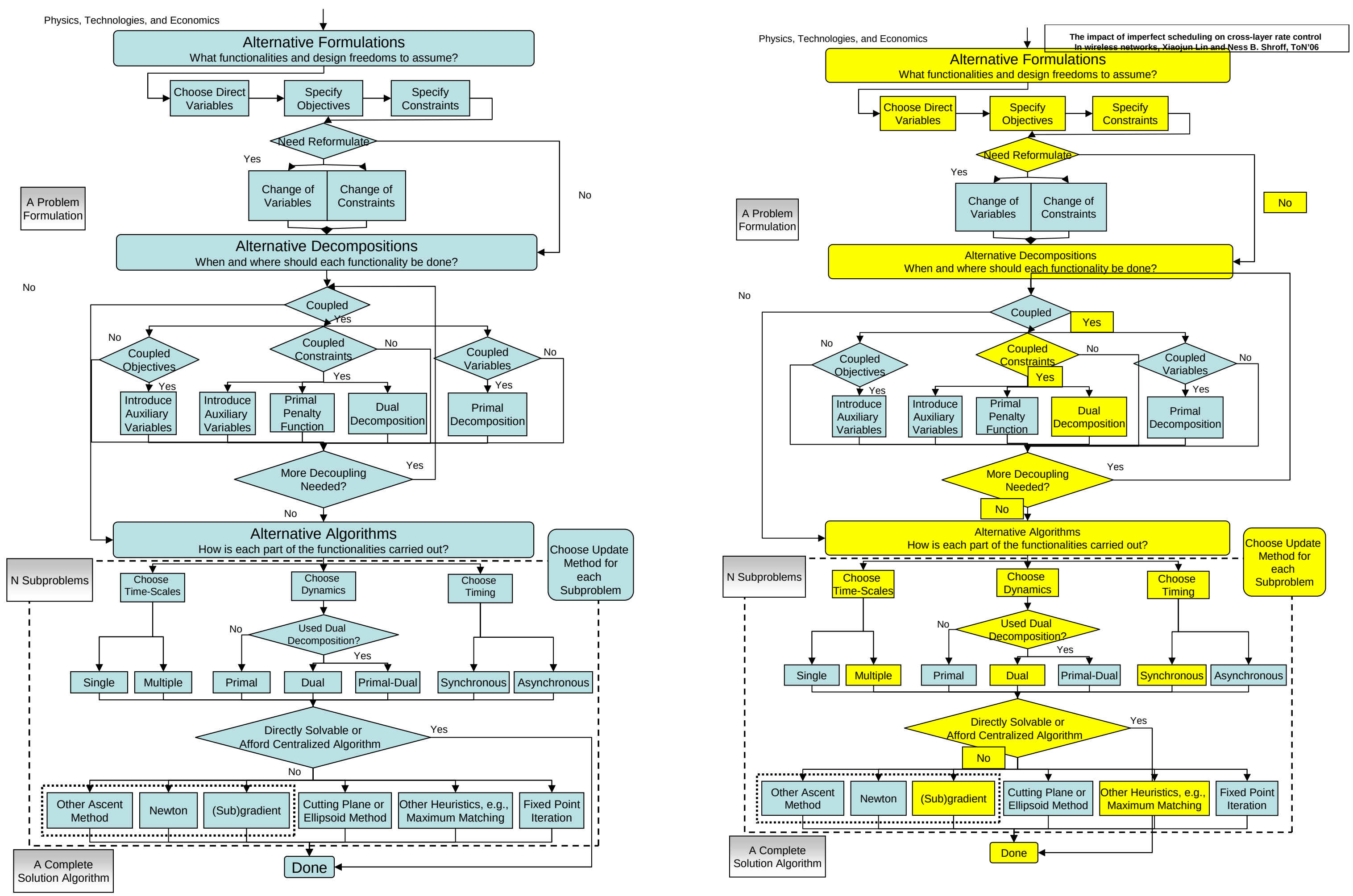
User Manual for decomposition alternatives

# Alternative Decompositions




Need to explore the space of alternative decompositions

# Alternative Decomposition Flowchart



## CAD Tool

Automate the enumeration of alternative decompositions:

Automate the comparison of alternative decompositions:

- Speed of convergence
- Robustness (errors, failures, network dynamics)
- Message passing (amount, locality, symmetry)
- Local computation (amount, symmetry)
- Ease of relaxing to simpler heuristics
- Ease of modification as new applications arise

Challenge: Some of the following metrics are not well defined, fully quantified, or accurately characterized

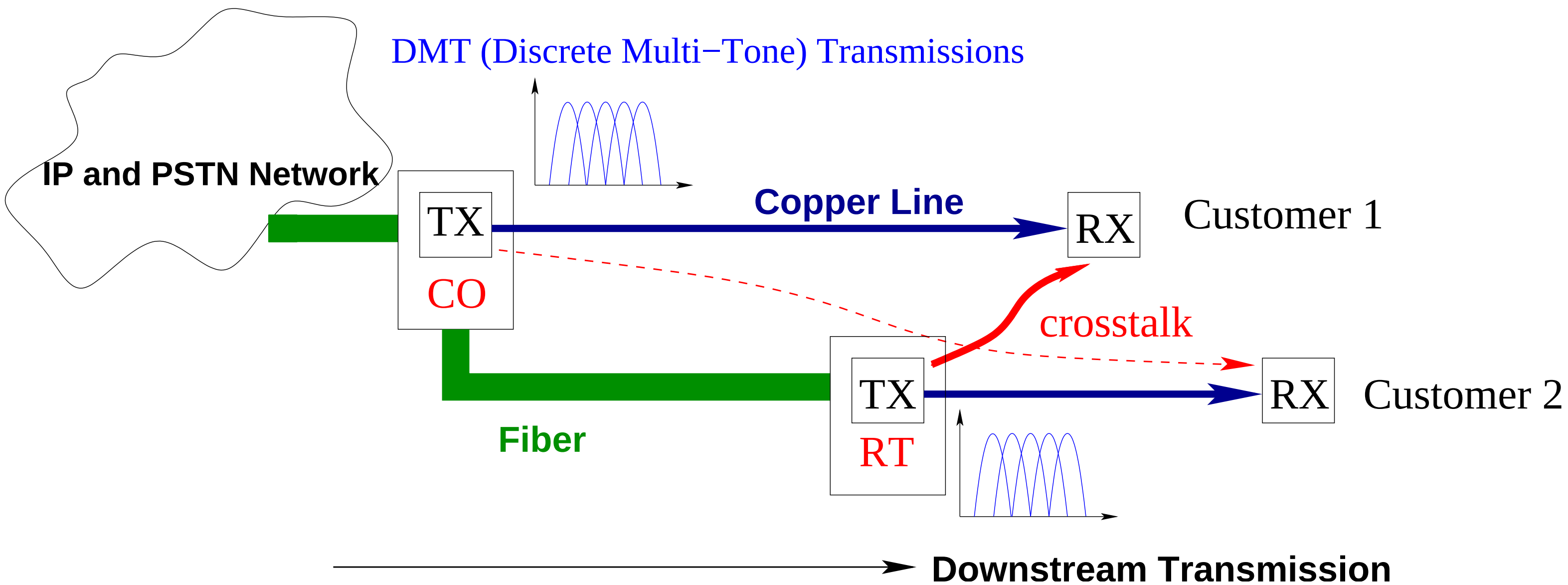
## The Challenge of Coupling

Not every coupling is dual-decomposable

There are much tougher coupling:

- Objective function: network lifetime or coupled utilities
- Constraint: Perron-Frobenius eigenvector in power control

# Case 1: DSL Spectrum Management



## Dynamic Spectrum Management

Problem formulation to characterize rate region

$$
\begin{array}{ll}
\text{maximize} & \sum_n w_n R_n \\
\text{subject to} & R_n = \sum_k \log\left(1 + \frac{p_n^k}{\sum_{m \neq n} \alpha_{n,m}^k p_m^k + \sigma_n^k}\right) \\
 & \sum_k p_n^k \leq P_n^{\max}, \forall n
\end{array}
$$

- Nonconvex
- Coupled across users
- Coupled across tones

## History

- IW: Iterative Water-filling [Yu Ginis Cioffi 02]
- OSB: Optimal Spectrum Balancing [Cendrillon et. al. 04]
- ISB: Iterative Spectrum Balancing [Liu Yu 05] [Cendrillon Moonen 05]
- ASB: Autonomous Spectrum Balancing [Cendrillon Huang Chiang Moonen TransSignalProc06]
- Many other work: BPM, SCALE, IW variants...

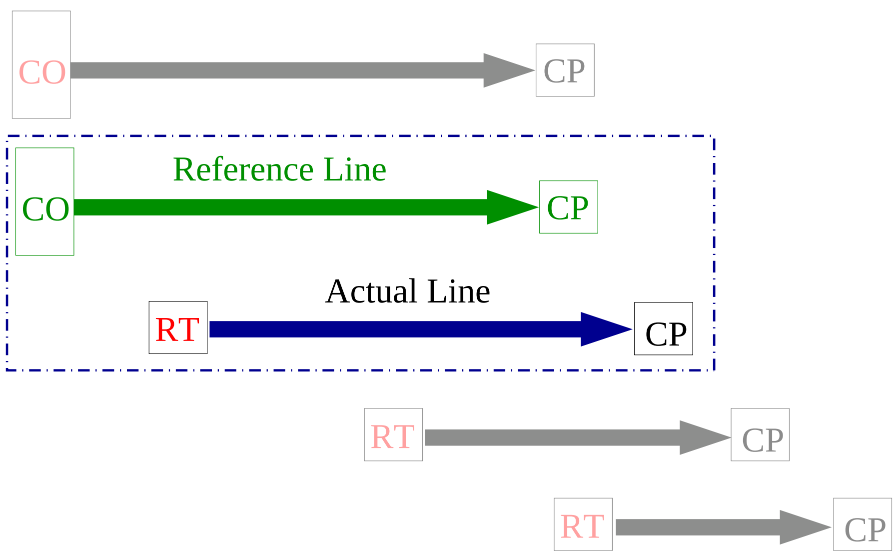
| Algorithm | Operation | Complexity | Performance |
|---|---|---|---|
| IW | Autonomous | $O\left(KN\right)$ | Suboptimal |
| OSB | Centralized | $O\left(Ke^N\right)$ | Optimal |
| ISB | Centralized | $O\left(KN^2\right)$ | Near Optimal |
| ASB | Autonomous | $O\left(KN\right)$ | Near Optimal |

$K$: number of carriers $\quad$ $N$: number of users

# Solution Idea: Static Pricing

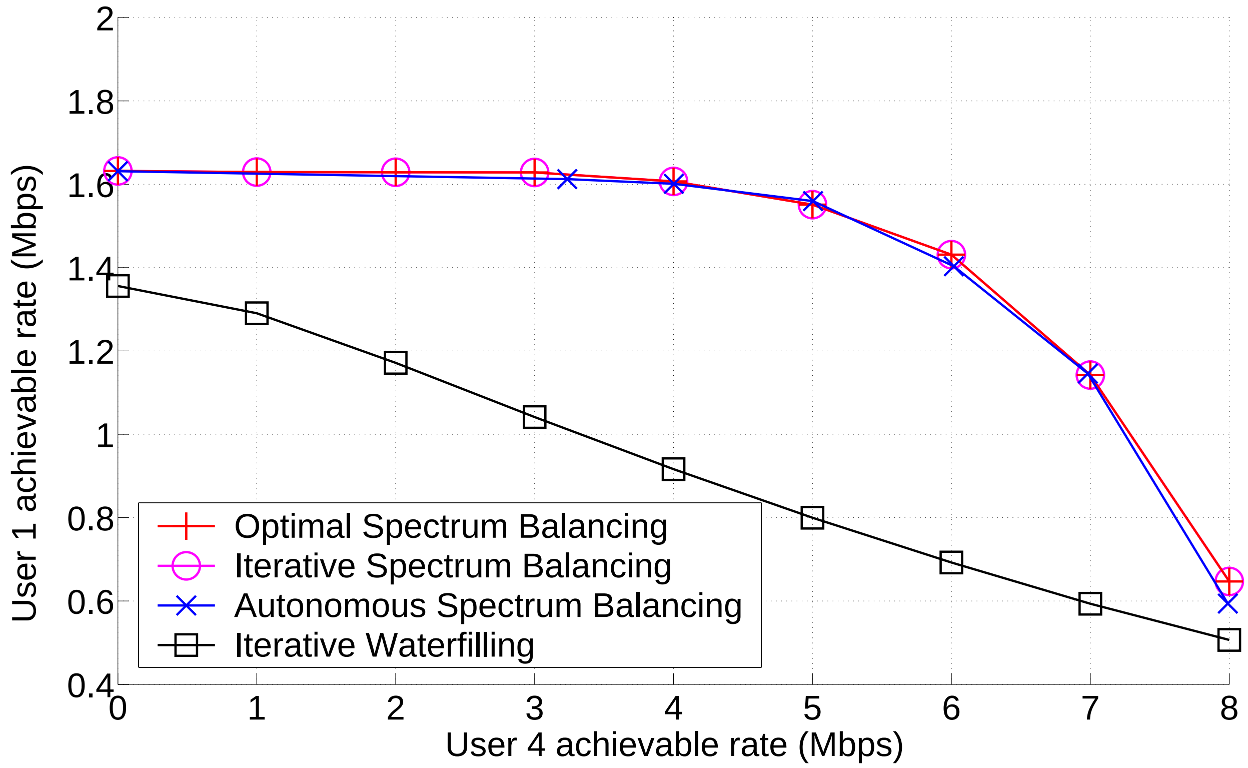
Dynamic pricing for dynamic coupling: decouple tones

Static pricing for static coupling: decouple users

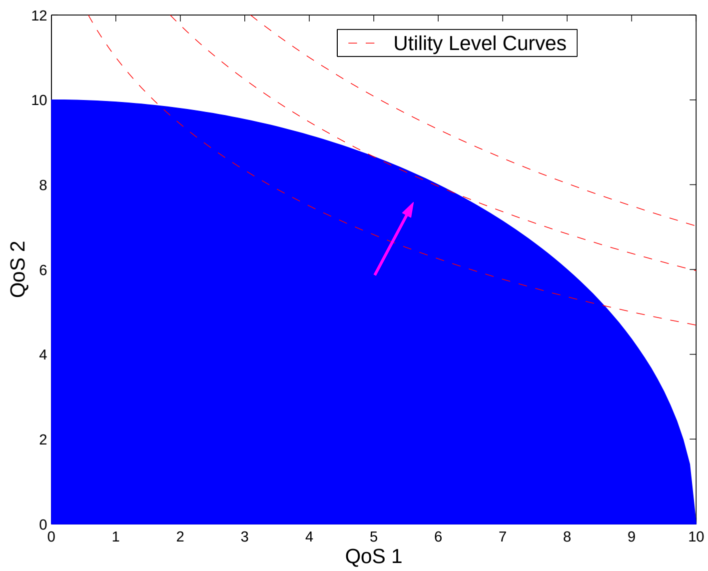



Same convergence conditions as iterative-waterfilling proved

# Much Larger Rate Region (Marvell Simulator)



# Case 2: Wireless Network Power Control




Maximize: utility function of powers and SIR assignments

Subject to: SIR assignments feasible

Variables: transmit powers and SIR assignments

## History

- Late 1980s: Qualcomm's received power equalization for near-far problem

- 1992-2000 Fixed SIR: distributed power control:

Zander 1992, Foschini Miljanic 1993, Mitra 1993, Yates 1995, Bambos Pottie 2000 ...

- Late 1990s: 3G for data wireless networks

- 2001-2004 Nash equilibrium for joint SIR assignment and power control:

Saraydar, Mandayam, Goodman 2001, 2002, Sung Wong 2002, Altman 2004 ...

- 2004-2005 Centralized computation for globally optimal joint SIR assignment and power control:

O'Neill, Julian, and Boyd 2004, Chiang 2004, Boche and Stanczak 2005

- 2006 Distributed and optimal joint control:

Hande Rangan Chiang Infocom06

# Load-Spillage Power Control (LSPC)

Reparameterization: From right eigenvector to left eigenvector:

---

Initialize: Arbitrary $\mathbf{s}[0] \succ 0$.

1. BS $k$ broadcasts the BS-load factor $\ell_k[t] = \sum_{i \in S_k} s_i[t]$.
2. Compute the spillage-factor $r_i[t]$ by $\sum_{j \neq i, j \in S_{\sigma_i}} s_j + \sum_{k \neq \sigma_i} h_{ki} \ell_k$.
3. Assign SIR values $\gamma_i[t] = s_i[t]/r_i[t]$.
4. Measure the resulting interference $q_i[t]$.
5. Update (in a distributed way) the load factor $s_i[t]$:

$$s_i[t+1] = s_i[t] + \delta \Delta s_i[t].$$

where $\Delta s_i = \frac{U_i'(\gamma_i)\gamma_i}{q_i} - s_i$

Continue: $t := t + 1$.

---

## Convergence and Optimality

Theorem: For convex SIR feasibility region, and sufficiently small step size $\delta > 0$, Algorithm converges to the globally optimal solution of

$$\begin{array}{ll} \text{maximize} & U(\boldsymbol{\gamma}) \\ \text{subject to} & \rho(\mathbf{D}(\boldsymbol{\gamma})\mathbf{G}) \leq 1 \end{array}$$

Proof: Key ideas:

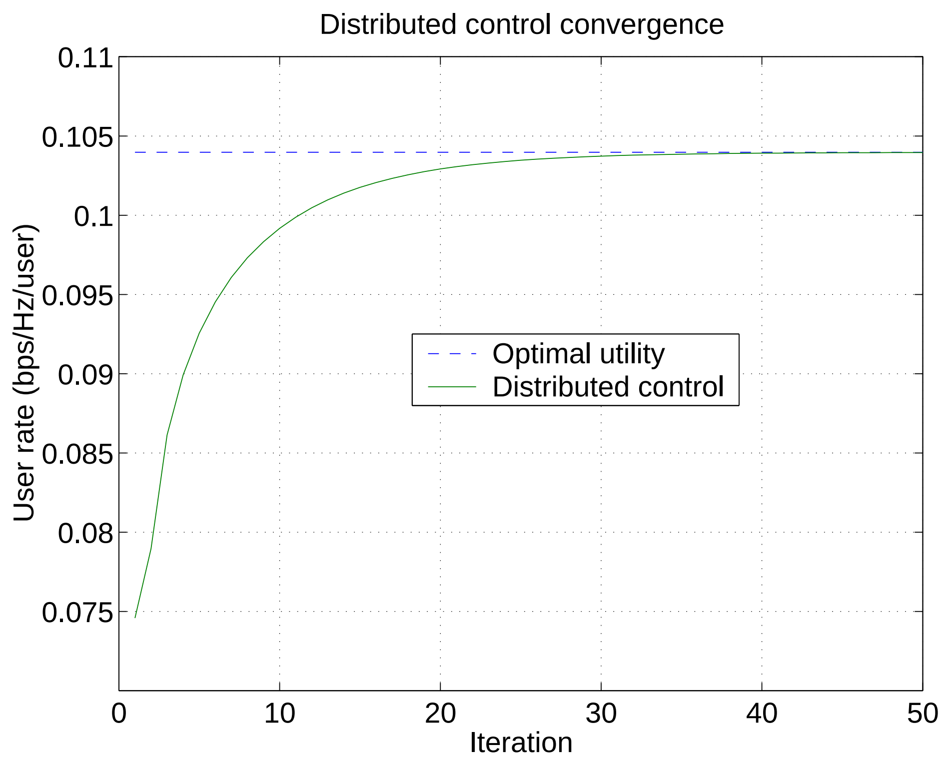
- Develop a locally-computable ascent direction (most involved step)
- Evaluate KKT conditions
- Guarantee Lipschitz condition

Extend to joint beamforming and bandwidth allocation

# Fast Convergence (3GPP2 Simulator)

570 mobile stations over 57 sectors

Fast convergence with distributed control



# Part III

## Robustness to Stochastic Dynamics

## The Bigger Picture of Kelly 1998

Shannon 1948: turn focus from finite blocklength codes to asymptotically large blocklength

- Law of Large Numbers kicks in
- Fundamental limit and digital architecture
- Later finite codewords come back...

Kelly 1998: turn focus from coupled queuing dynamics to deterministic formulations

- Optimization and decomposition view kicks in
- Network protocols as dynamic control systems
- Later stochastics come back...

# Stochastic Network Utility Maximization

Filling in the table with 3 stars would be a long-overdue union between stochastic networks and distributed optimization (survey in YiChiang07)

| | Stability or Validation | Average Performance | Outage Performance | Fairness |
|---|---|---|---|---|
| *Session Level* | $\star\star$ | $\star$ | | $\star$ |
| *Packet Level* | $\star$ | $\star$ | | |
| *Channel Level* | $\star\star$ | $\star$ | | |
| *Topology Level* | | | | |

Timescale of interactions is crucial

Only look at box (1,1) in this talk

# Session Level Stochastic Stability

Dynamic user population with arrivals and departures

$$\begin{array}{ll} \text{maximize} & \sum_s N_s(t) U(\phi_s / N_s(t)) \\ \text{subject to} & \phi \in \mathcal{R} \end{array}$$

• If Poisson ($\lambda$) arrival with exp ($1/\mu$) filesize distribution:

Number of active sources follows Markov chain:

$N_s(t) \rightarrow N_s(t) + 1$ with rate $\lambda_s$

$N_s(t) \rightarrow N_s(t) - 1$ with rate $\mu_s \phi_s(\mathbf{N}(t), \mathcal{R})$

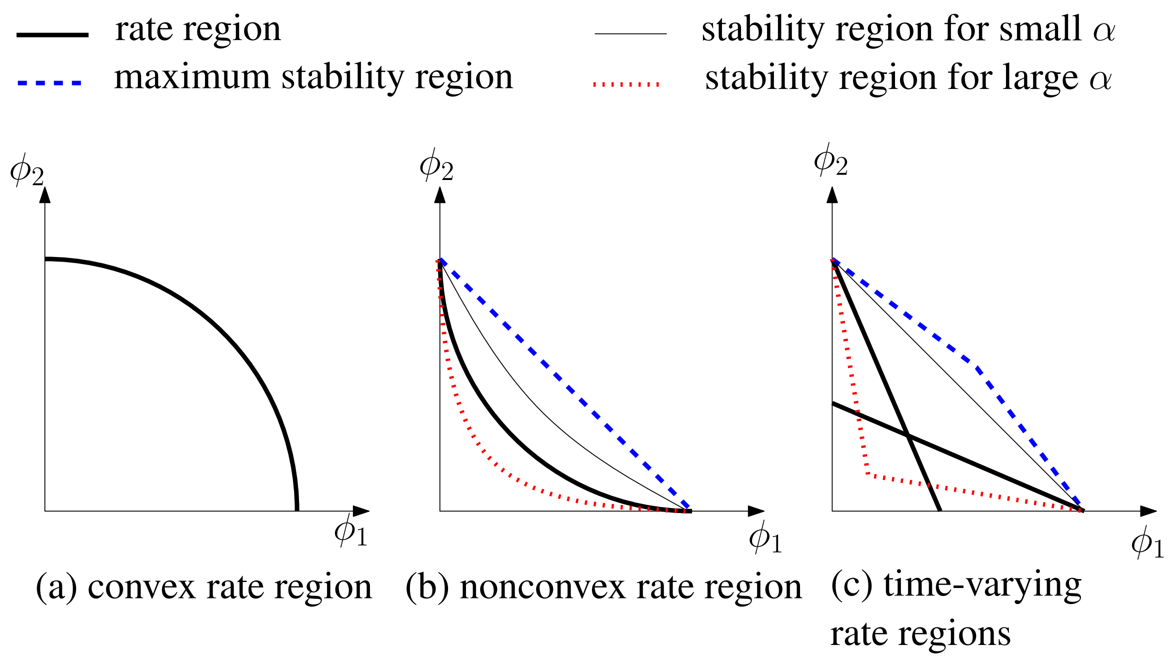
Queue/rate stability of $M/SD/1/\infty$ queuing network

$\lambda/\mu \in \mathcal{R}$ is necessary, is it also sufficient?

## Stability I: Simple Constraint Set

| *Work* | *Arrival* | *Topology* | $U_i$ | $U$ *shape* |
|---|---|---|---|---|
| de Veciana et.al. 99 | Poisson, Exp | General | Same | $\alpha = 1, \infty$ |
| Bonald Massoulie 01 | Poisson, Exp | General | Diff. | General |
| Lin Shroff, Srikant 04 | Poisson, Exp<br>Fast timescale | General | Same | $\alpha > 1$ |
| Ye et.al. 05 | Exp filesize | General | Diff. | General |
| Bramson 05 | General | General | Same | $\alpha = \infty$ |
| Lakshmikantha et.al. 05 | Phase type | $2 \times 2$ grid | Same | $\alpha = 1$ |
| Massoulie 06 | Phase type | General | Same | $\alpha = 1$ |
| Gromoll Williams 06 | General | Tree | Same | General |
| Chiang Shah Tang 06 | General | General | Diff. | A range of $\alpha$ |
| Open | General | General | Diff. | All $\alpha$ |

# Stability II: General Constraint Set



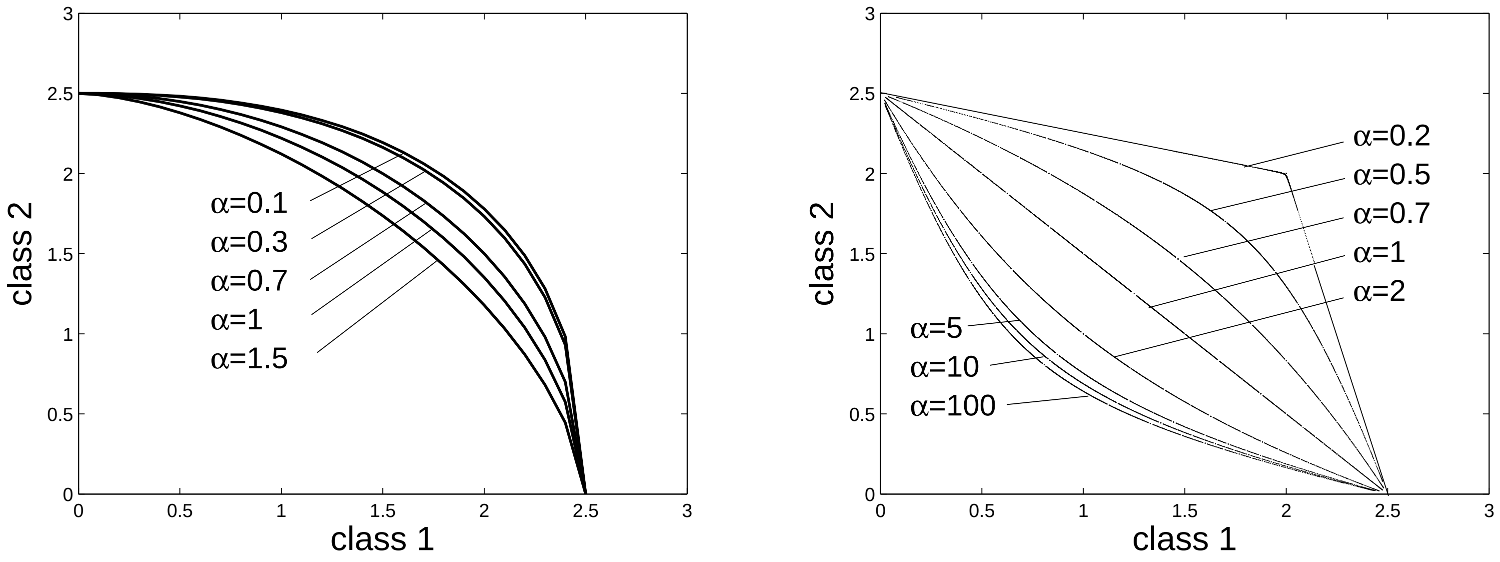

Convex rate region case: stability region is rate region

What about nonconvex or time-varying rate region?

(LiuProutiereYiChiangPoor-Sigmetrics07)

May not be maximum stability region and sensitive to $\alpha$

# Stability-Fairness Tradeoff




More fair allocation has smaller stability region
when rate region is time-varying

## Proof Techniques

- Fluid limit proof
- Laypunov function construction
- Max projection and monotone cone policy

## Open Problems

- Fluid model or fluid limit?
- Does P2P and IPTV traffic require different models?
- How many flows is “many-flow”?
- Design for topology level stochastics?
- From convergence to equilibrium to invariance during transience

# Part IV

DFO

## Design For Optimizability

Nonconvexity happens:

- Nonconcave utility (eg, real-time applications)
- Nonconvex constraints (eg, power control in low SIR)
- Integer constraints (eg, single-path routing)
- Exponentially long description length (eg, certain scheduling)

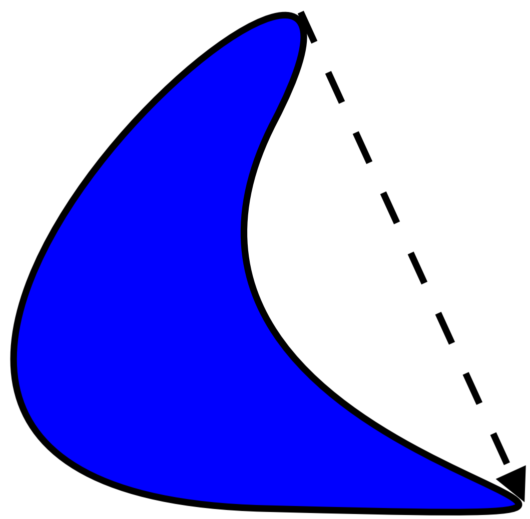
Mathematically, convexity not invariant, so we can have, e.g.,

- Sum-of-squares method (Stengle73, Parrilo03)
- Geometric programming (DuffinPetersonZener67)

More engineering approach: Design for Optimizability

## Tackling Nonconvexity

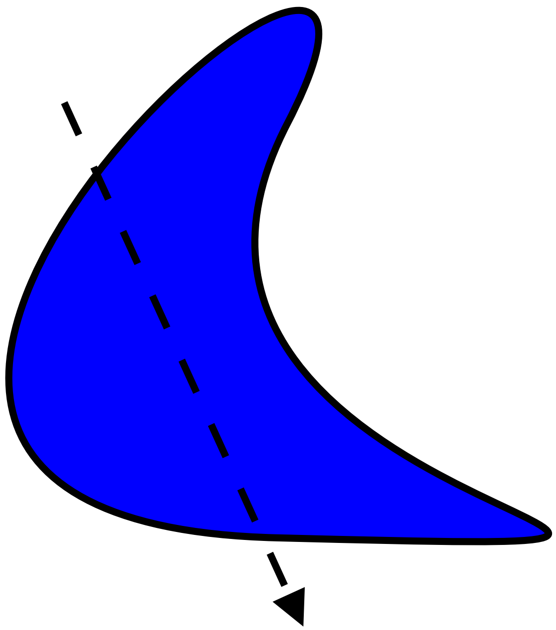
Option 1: Go around nonconvexity



- Geometric Programming, change of variable
- Sufficient condition under which the problem is convex
- Sufficient conditions for uniqueness of KKT points

# Tackling Nonconvexity

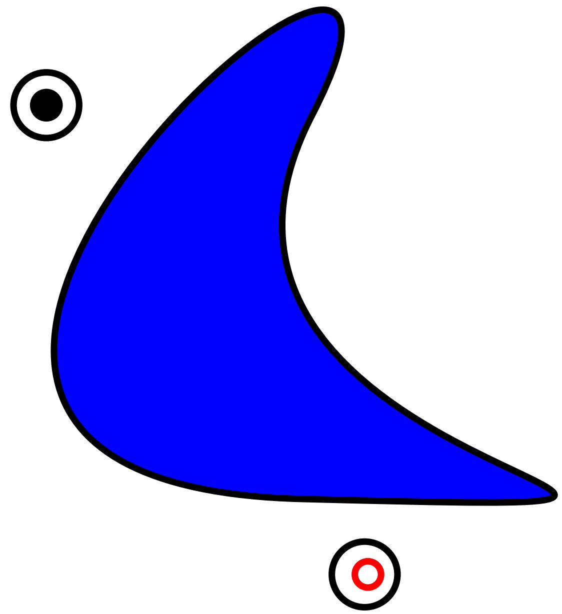
Option 2: Go through nonconvexity



- SOS, Signomial programming, successive convex approximation
- Special structure (e.g., DC, generalized quasiconcavity)
- Canonical duality, Smart branch and bound, etc.

# Tackling Nonconvexity

Option 3: Go above nonconvexity: Design for Optimizability



Change difficult optimization problem, rather than solve it

- Redraw architecture or protocol to make the problem easy to solve
- Need to balance with the cost of making changes to protocols

Optimization as a flag to design issues

# Case 3: Internet Routing and Traffic Engineering

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

- OSPF: a reverse shortest path method

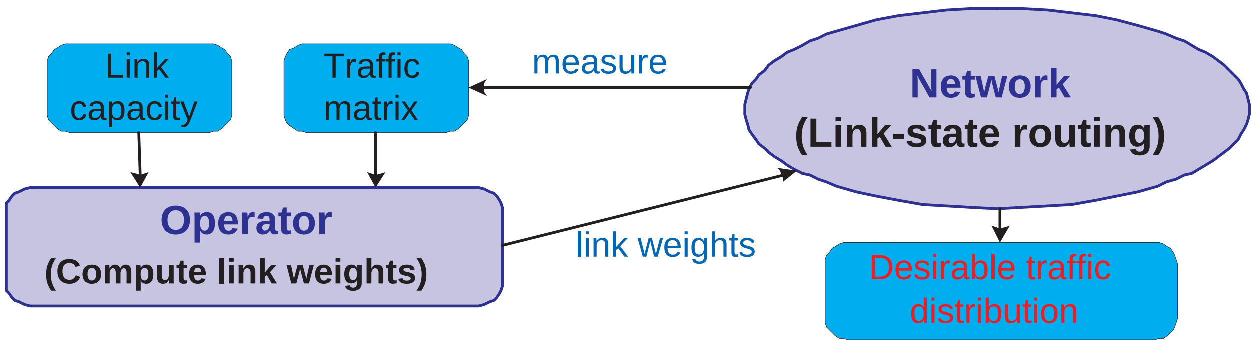
Link-weight-based traffic engineering has two key components:

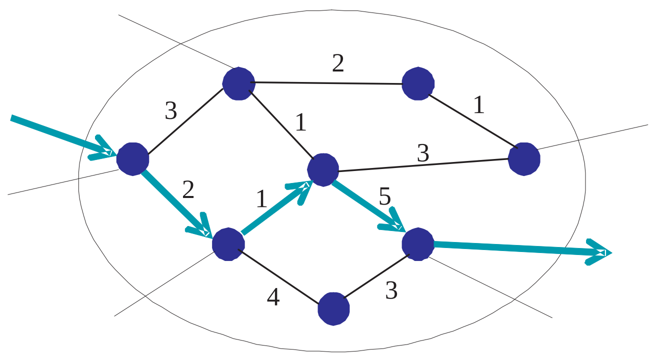
- Centralized computation for setting link weights
- Distributed way of using these link weights to do destination-based packet forwarding

Focus of this talk: Link weight computation:

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

# Internet Routing and Traffic Engineering




Path length= 8

## 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 and show achievability of optimal traffic engineering

## From OSPF to DEFT

A new way to use link weights (XuChiangRexford-Infocom07):

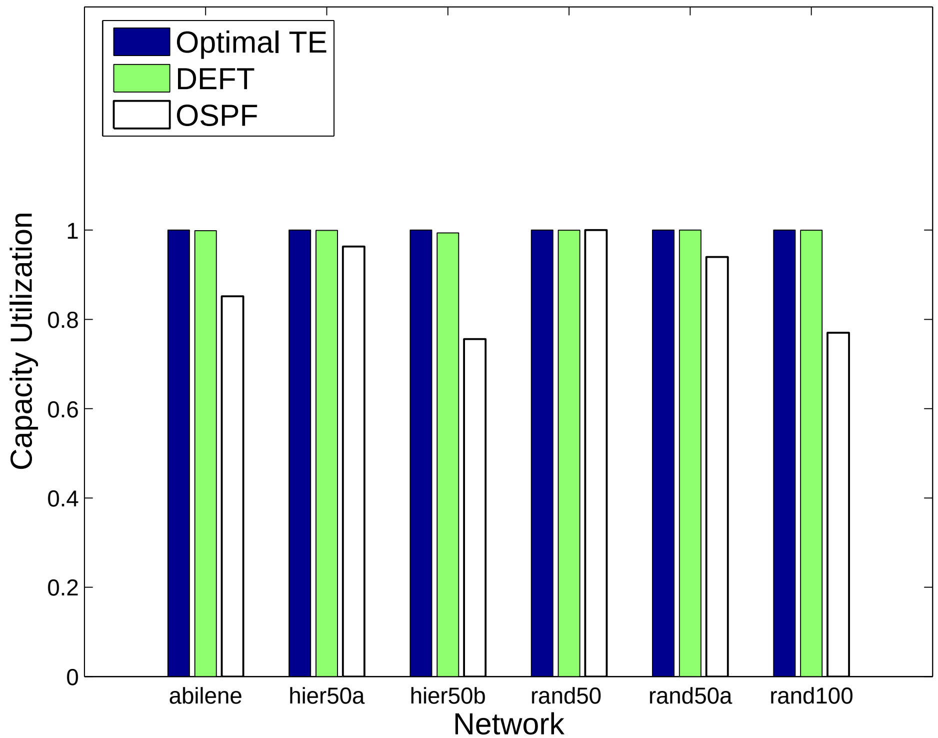
- Use link weights to compute path weights
- Split traffic on all paths
- Exponential penalty on longer paths

Same way to do (destination-based) packet forwarding

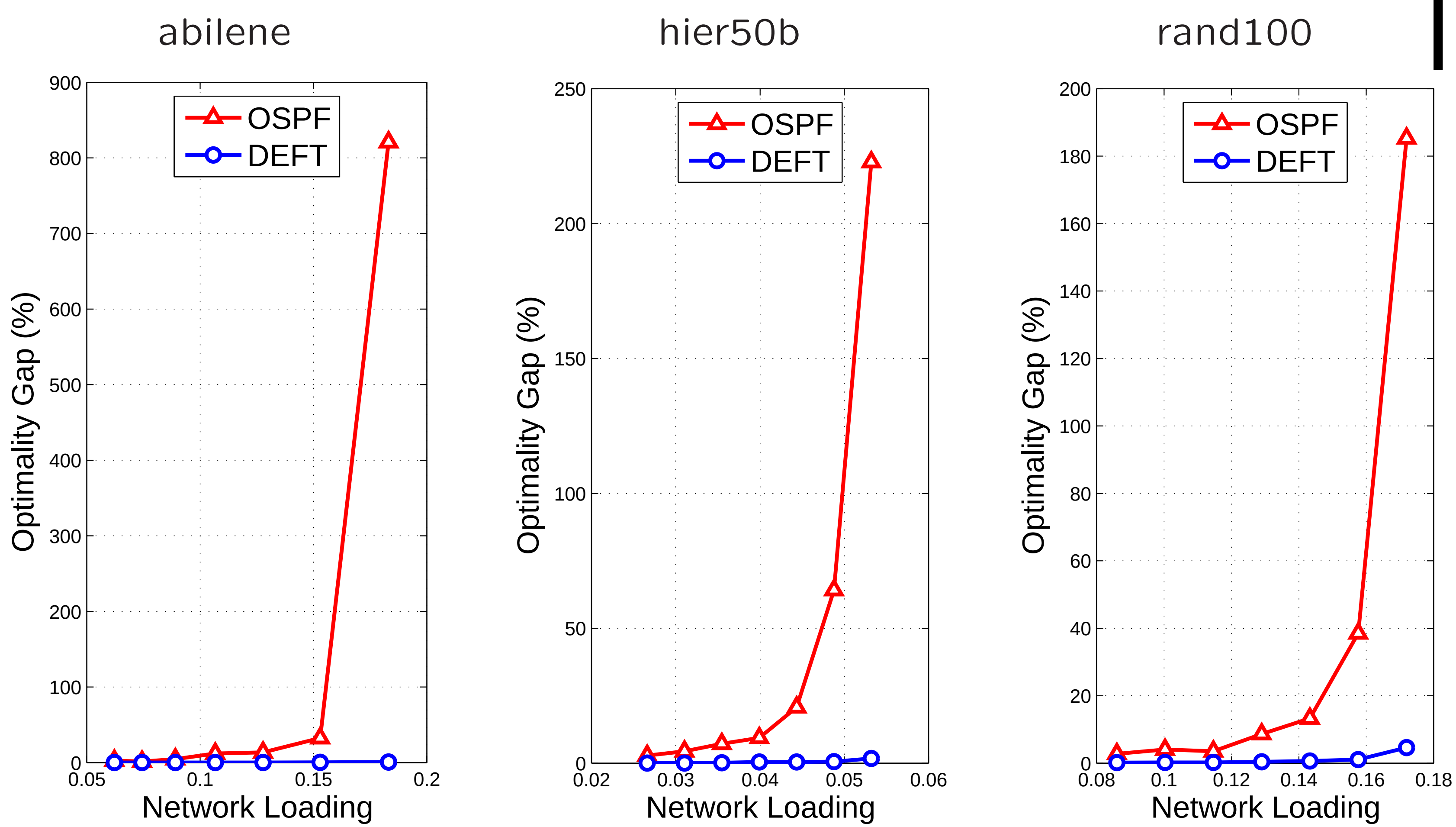
How good can the new protocol be?

How to compute link weights in the new protocol?

# Capacity Improvement (Abilene Traffic Trace)



# Optimality Gap Reduction



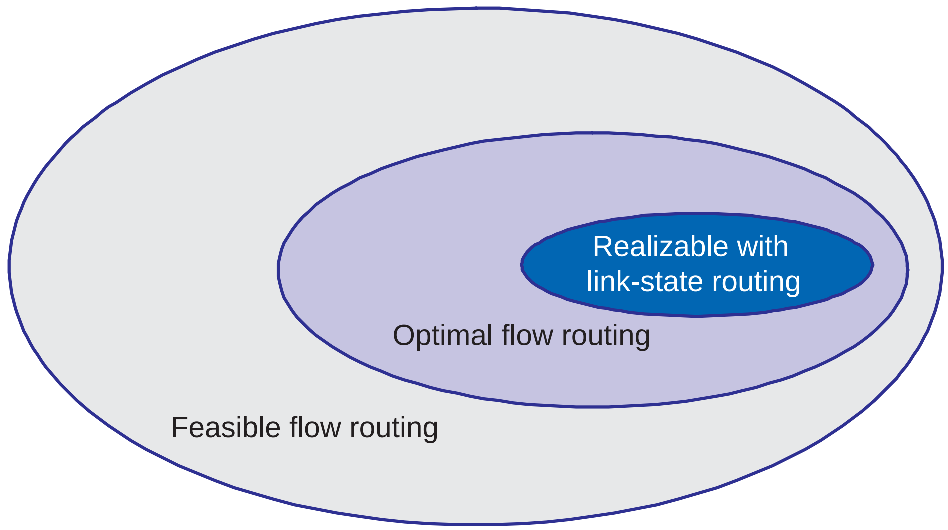
## Simple Routing Can Be Optimal

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

Theorem: Optimal weights can be computed in polynomial time

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

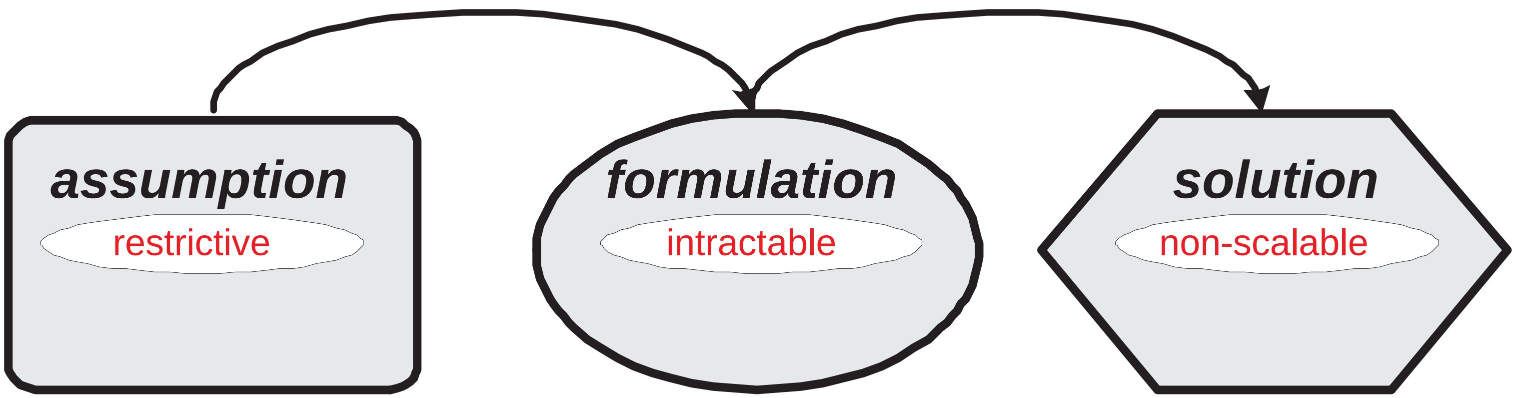
Entropy function is the right choice, and the only one

## Nonconvexity Can Be Sweet

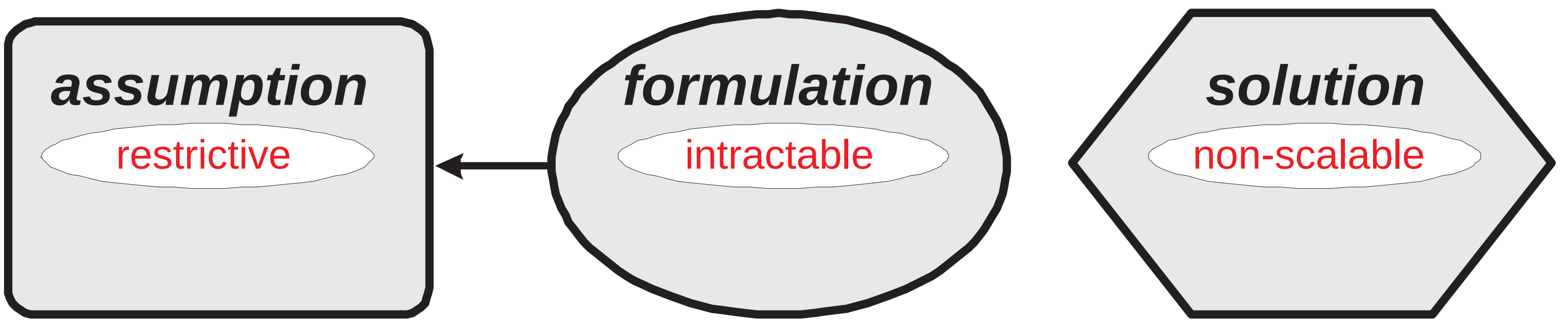
Sometimes, hard problems aren't hard in reality. When?

Sometimes, hard problems don't deserve to exist. How?

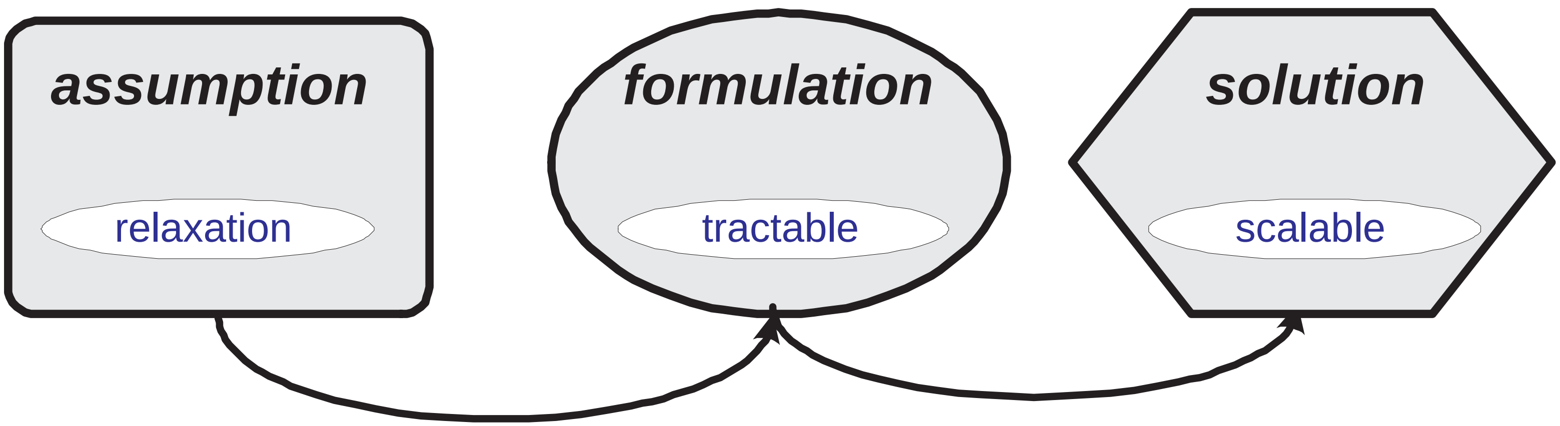
## Solve Hard Problems



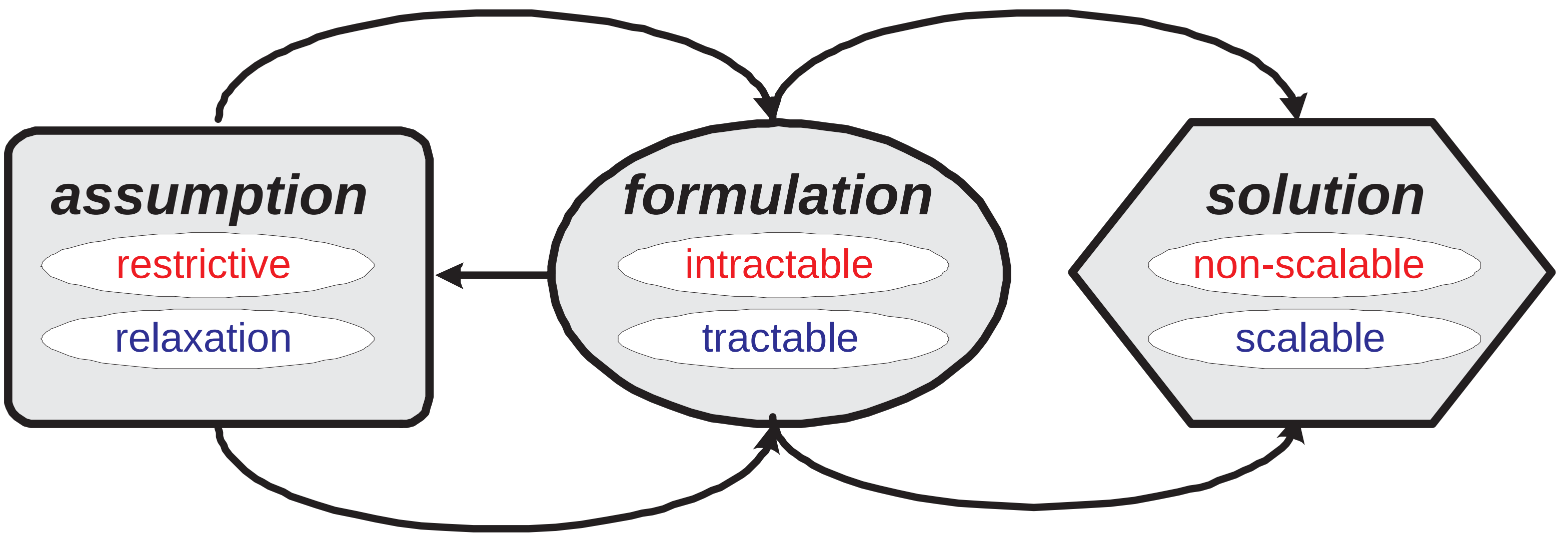
## Don't Solve Hard Problems



# Hard Problems Become Easy



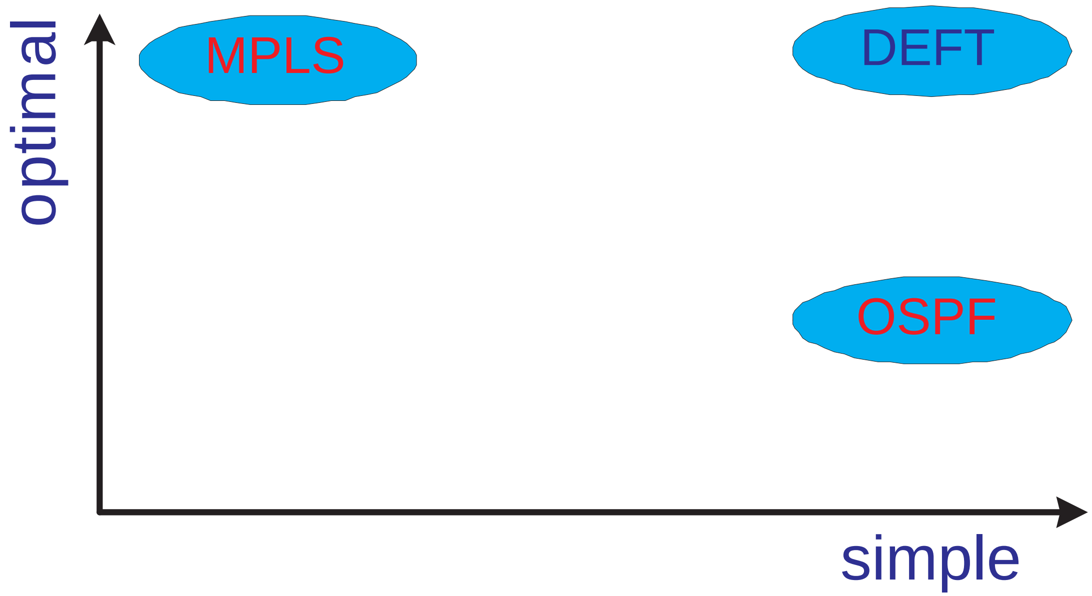
## Feedback in Engineering Process



# Optimizability-Complexity Tradeoff

Often there is a price for revisiting assumptions

In Internet traffic engineering case, DFO provides the best possible tradeoff



# Part V

## Complexity

## Beyond Optimality

I. Modeling: Resource allocation, fairness, reverse-engineering

II. Architecture: who does what and how to connect

III. Robustness to stochastic dynamics

IV. Feedback to engineering assumptions

V. Complexity-performance tradeoff

Optimization as a language to think about network engineering

## Contacts

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