Incentives and Internet Algorithms

Joan Feigenbaum
Yale University
http://www.cs.yale.edu/~jf

Scott Shenker
ICSI and U.C. Berkeley
http://www.icir.org/shenker

Slides: http://www.cs.yale.edu/~jf/IPCO04.{ppt,pdf}
Acknowledgments: Vijay Ramachandran (Yale)
Rahul Sami (MIT)

Outline

- Motivation and Background
- Example: Multicast Cost Sharing
- Overview of Known Results
- Three Research Directions
- Open Questions

Three Research Traditions

- Theoretical Computer Science: complexity
  - What can be feasibly computed?
  - Centralized or distributed computational models

- Game Theory: incentives
  - What social goals are compatible with selfishness?

- Internet Architecture: robust scalability
  - How to build large and robust systems?

Different Assumptions

- Theoretical Computer Science:
  - Nodes are *obedient*, *faulty*, or *adversarial.*
  - Large systems, limited comp. resources

- Game Theory:
  - Nodes are *strategic* (selfish).
  - Small systems, unlimited comp. resources

Internet Systems (1)

- Agents often autonomous (users/ASs)
  - Have their own individual goals
- Often involve “Internet” scales
  - Massive systems
  - Limited comm./comp. resources
- *Both incentives and complexity matter.*

Internet Systems (2)

- Agents (users/ASs) are dispersed.
- Computational nodes often dispersed.
- *Computation is (often) distributed.*

Internet Systems (3)

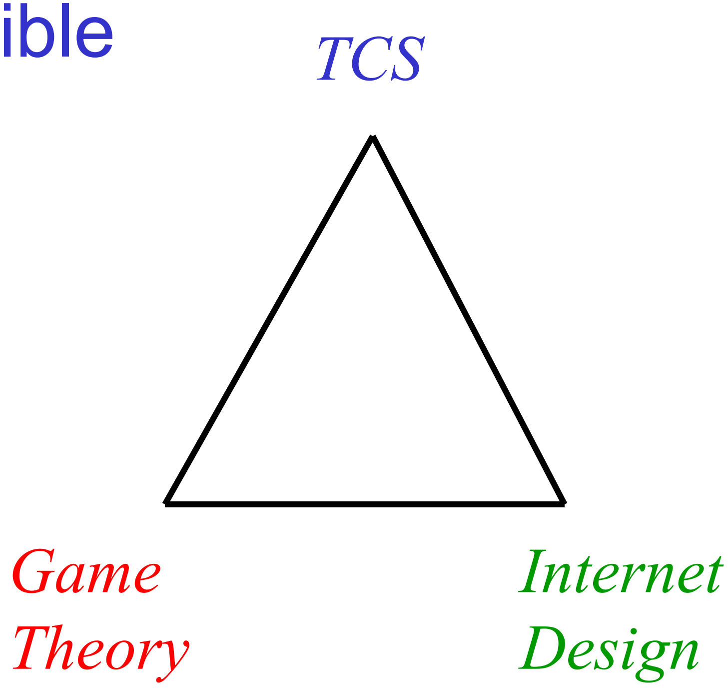
- Scalability and robustness paramount
  - sacrifice strict semantics for scaling
  - many informal design guidelines
  - Ex: end-to-end principle, soft state, *etc.*

- *Computation must be "robustly scalable."*
  - even if criterion not defined precisely
  - *If TCP is the answer, what's the question?*

Fundamental Question

What computations are (simultaneously):

- Computationally feasible
- Incentive-compatible
- Robustly scalable



Game Theory and the Internet

- Long history of work:
  - Networking: Congestion control [N85], *etc.*
  - TCS: Selfish routing [RT02], *etc*.

- Complexity issues not explicitly addressed
  - though often moot

TCS and Internet

- Increasing literature
  - TCP [GY02,GK03]
  - routing [GMP01,GKT03]
  - *etc.*

- No consideration of incentives

- Doesn't always capture Internet style

Game Theory and TCS

- Various connections:
  - Complexity classes [CFLS97, CKS81, P85, *etc.*]
  - Price of anarchy, complexity of equilibria, *etc*. [KP99,CV02,DPS02]

- Algorithmic Mechanism Design (AMD)
  - Centralized computation [NR01]

- Distributed Algorithmic Mechanism Design (DAMD)
  - Internet-based computation [FPS01]

DAMD: Two Themes

- Incentives in Internet computation
  - Well-defined formalism
  - Real-world incentives hard to characterize
- Modeling Internet-style computation
  - Real-world examples abound
  - Formalism is lacking

Outline

- Motivation and Background
  - Mechanism Design
  - Internet Computation
- Example: Multicast Cost Sharing
- Overview of Known Results
- Three Research Directions
- Open Questions

System Notation

Outcomes and *agents*:

- Φ is set of possible *outcomes.*
  - $o \in \Phi$ represents particular outcome.
- Agents have valuation functions v_i.
  - $v_i(o)$ is “happiness” with outcome o.

Societal vs. Private Goals

- System-wide performance goals:
  - Efficiency, fairness, *etc*.
  - Defined by set of outcomes $G(v) \subset \Phi$
- Private goals: Maximize own welfare
  - v_i is private to agent i.
  - Only reveal truthfully if in own interest

Mechanism Design

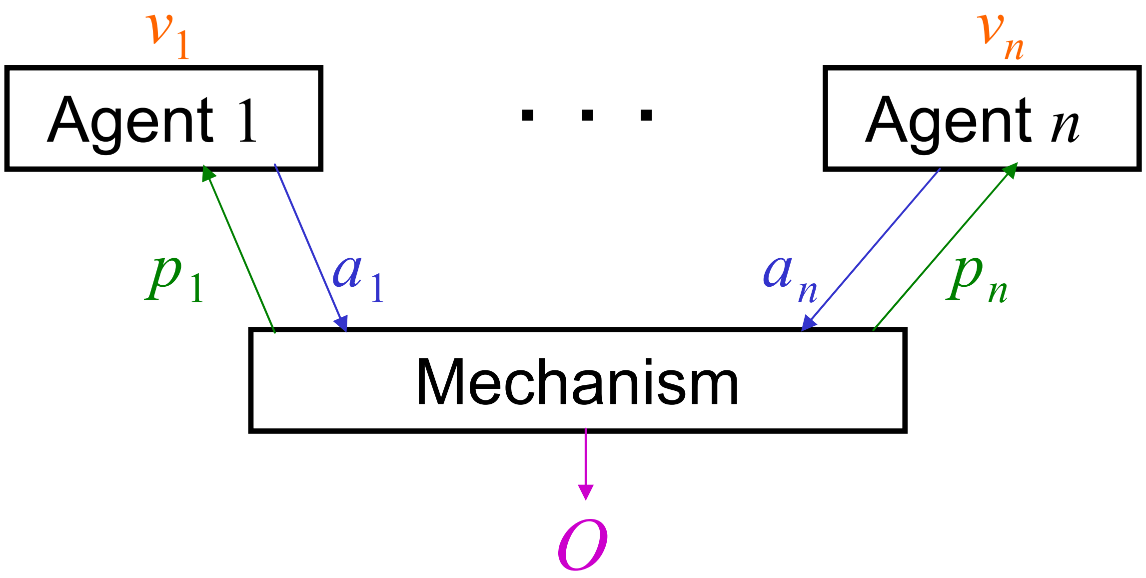
- Branch of game theory:
  - reconciles private interests with social goals

- Involves esoteric game-theoretic issues
  - will avoid them as much as possible
  - only present MD content relevant to DAMD

Mechanisms

Actions: a_i Outcome: $O(a)$ Payments: $p_i(a)$

Utilities: $u_i(a) = v_i(O(a)) + p_i(a)$



Mechanism Design

- $A_O(v)$ = {action vectors} consistent w/selfishness
  - a_i "maximizes" $u_i(a) = v_i(O(a)) + p_i(a)$.
  - "maximize" depends on information, structure, *etc.*
  - *Solution concept:* Nash, Rationalizable, ESS, *etc.*

- Mechanism-design goal: $O(A_O(v)) \subseteq G(v)$ for all v

- Central MD question: *For given solution concept, which social goals can be achieved?*

Direct Strategyproof Mechanisms

- *Direct*: Actions are declarations of v_i.
- *Strategyproof*: $u_i(v) \geq u_i(v_{-i}, x_i)$, for all x_i , v_{-i}
  - Agents have no incentive to lie.
  - $A_O(v) = \{v\}$ *"truthful revelation"*
- *Which social goals achievable with SP?*

Strategyproof Efficiency

Efficient outcome: maximizes Σv_i

VCG Mechanisms:

- $O(v) = \tilde{o}(v)$ where $\tilde{o}(v) = \arg\max_o \Sigma v_i(o)$
- $p_i(v) = \sum_{j \neq i} v_j(\tilde{o}(v)) + h_i(v_{-i})$

Why are VCG Strategyproof?

- Focus only on agent i
  - v_i is truth; x_i is declared valuation
  - $p_i(x_i) = \sum_{j \neq i} v_j(\tilde{o}(x_i)) + h_i$

- $u_i(x_i) = v_i(\tilde{o}(x_i)) + p_i(x_i) = \sum_j v_j(\tilde{o}(x_i)) + h_i$

- Recall: $\tilde{o}(v_i)$ maximizes $\sum_j v_j(o)$

Group Strategyproofness

Definition:

- True: v_i Reported: x_i
- Lying set $S = \{i : v_i \neq x_i\}$

$$\exists\, i \in S\ u_i(x) > u_i(v) \ \Rightarrow \exists\, j \in S \quad u_j(x) < u_j(v)$$

- *If any liar gains, at least one will suffer.*

Outline

- Motivation and Background
  - Mechanism Design
  - Internet Computation
- Example: Multicast Cost Sharing
- Overview of Known Results
- Three Research Directions
- Open Questions

Algorithmic Mechanism Design [NR01]

Require polynomial-time computability:

- $O(a)$ and $p_i(a)$

Centralized model of computation:

- good for auctions, *etc*.
- not suitable for distributed systems

Complexity of Distributed Computations (Static)

Quantities of Interest:

- Computation at nodes
- Communication:
  - total
  - hotspots
- Care about both messages and bits

“Good Network Complexity”

- Polynomial-time local computation
  - in total size or (better) node degree
- $O(1)$ messages per link
- Limited message size
  - F(# *agents*, *graph size*, *numerical inputs*)

Dynamics (partial)

- Internet systems often have “churn.”
  - Agents come and go
  - Agents change their inputs
- “Robust” systems must tolerate churn.
  - most of system oblivious to most changes
- Example of dynamic requirement:
  - $o(n)$ changes trigger $\Omega(n)$ updates.

Protocol-Based Computation

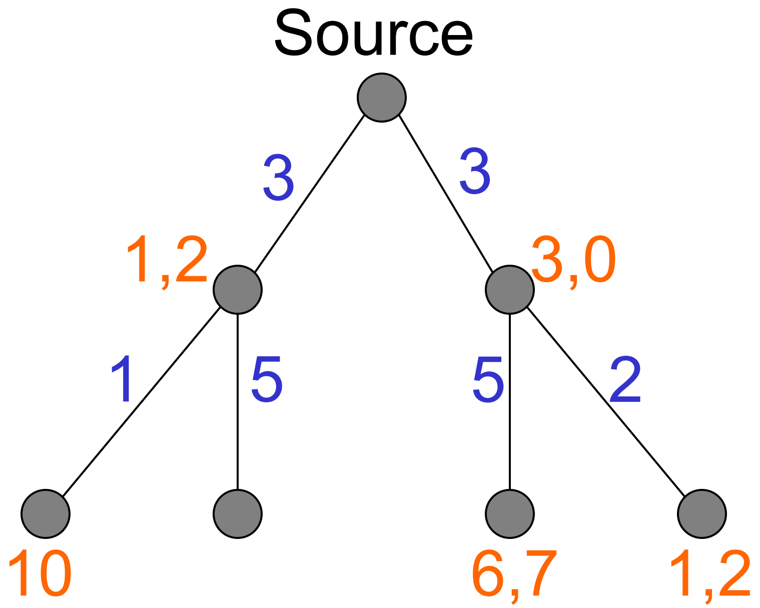
- Use standardized protocol as substrate for computation.
  - relative rather than absolute complexity

- Advantages:
  - incorporates informal design guidelines
  - adoption does not require new protocol
  - example: BGP-based mech's for routing

Outline

- Motivation and Background
- Example: Multicast Cost Sharing
- Overview of Known Results
- Three Research Directions
- Open Questions

Multicast Cost Sharing (MCS)



Users’ valuations: v_i

Link costs: $c(l)$

Receiver Set

Which users receive the multicast?

Cost Shares

How much does each receiver pay?

Model [FKSS03, §1.2]:

- Obedient Network
- Strategic Users

Notation

P	Users (or “participants”)
R	Receiver set ($\sigma_i = 1$ if $i \in R$)
p_i	User i’s cost share (*change in sign!*)
u_i	User i’s utility ($u_i = \sigma_i v_i - p_i$)
W	Total welfare $W(R) \triangleq V(R) - C(R)$

$$C(R) \triangleq \sum_{l \in T(R)} c(l) \qquad V(R) \triangleq \sum_{i \in R} v_i$$

“Process” Design Goals

- No Positive Transfers (NPT): $p_i \geq 0$
- Voluntary Participation (VP): $u_i \geq 0$

- Consumer Sovereignty (CS): For all trees and costs, there is a μ_{cs} s.t. $\sigma_i = 1$ if $v_i \geq \mu_{\mathrm{cs}}$.

- Symmetry (SYM): If i,j have zero-cost path and $v_i = v_j$, then $\sigma_i = \sigma_j$ and $p_i = p_j$.

Two “Performance” Goals

- Efficiency (EFF): $R = \arg\max W$
- Budget Balance (BB): $C(R) = \sum_{i \in R} p_i$

Impossibility Results

Exact [GL79]: No strategyproof mechanism can be both efficient and budget-balanced.

Approximate [FKSS03]: No strategyproof mechanism that satisfies NPT, VP, and CS can be both γ-approximately efficient and κ-approximately budget-balanced, for any positive constants γ, κ.

Efficiency

Uniqueness [MS01]: The only strategyproof, efficient mechanism that satisfies NPT, VP, and CS is the Marginal-Cost mechanism (MC):

$$p_i = v_i - (W - W^{-i}),$$

where W is maximal total welfare, and W^{-i} is maximal total welfare without agent i.

- MC also satisfies SYM.

Budget Balance (1)

General Construction [MS01]: Any cross-monotonic cost-sharing formula results in a group-strategyproof and budget-balanced cost-sharing mechanism that satisfies NPT, VP, CS, and SYM.

- R is biggest set s.t. $p_i(R) \leq v_i$, for all $i \in R$.

Budget Balance (2)

- *Efficiency loss* [MS01]: The Shapley-value mechanism (SH) minimizes the worst-case efficiency loss.

- SH Cost Shares: $c(l)$ is shared equally by all receivers downstream of l.

Network Complexity for BB

Hardness [FKSS03]: Implementing a group-strategyproof and budget-balanced mechanism that satisfies NPT, VP, CS, and SYM requires sending $\Omega(|P|)$ bits over $\Omega(|L|)$ links in worst case.

- *Bad network complexity!*

Network Complexity of EFF

“*Easiness*” [FPS01]: MC needs only:

- One modest-sized message in each link-direction
- Two simple calculations per node

- *Good network complexity!*

Computing Cost Shares

$$p_i \equiv v_i - (W - W^{-i})$$

Case 1: No difference in tree

Welfare Difference = v_i

Cost Share = 0

Case 2: Tree differs by 1 subtree.

Welfare Difference = W^{γ}

(minimum welfare subtree above i)

Cost Share = $v_i - W^{\gamma}$

Two-Pass Algorithm for MC

Bottom-up pass:

- Compute subtree welfares W^{γ}.
- If $W^{\gamma} < 0$, prune subtree.

Top-down pass:

- Keep track of minimum welfare subtrees.
- Compare v_i to minimal W^{γ}.

Computing the MC Receiver Set R

$$W^{\alpha} \equiv v^{\alpha} + \sum_{\substack{\beta \in \mathrm{Ch}(\alpha) \\ \text{s.t. } W^{\beta} \geq 0}} W^{\beta} - c^{\alpha}$$

Proposition:

$\mathrm{res}(\alpha) \subseteq R$ iff $W^{\gamma} \geqslant 0, \forall \gamma \in \{$anc. of α in $T(P)\}$

Additional Notation:

$\{\alpha, \beta, \gamma\} \subseteq P$

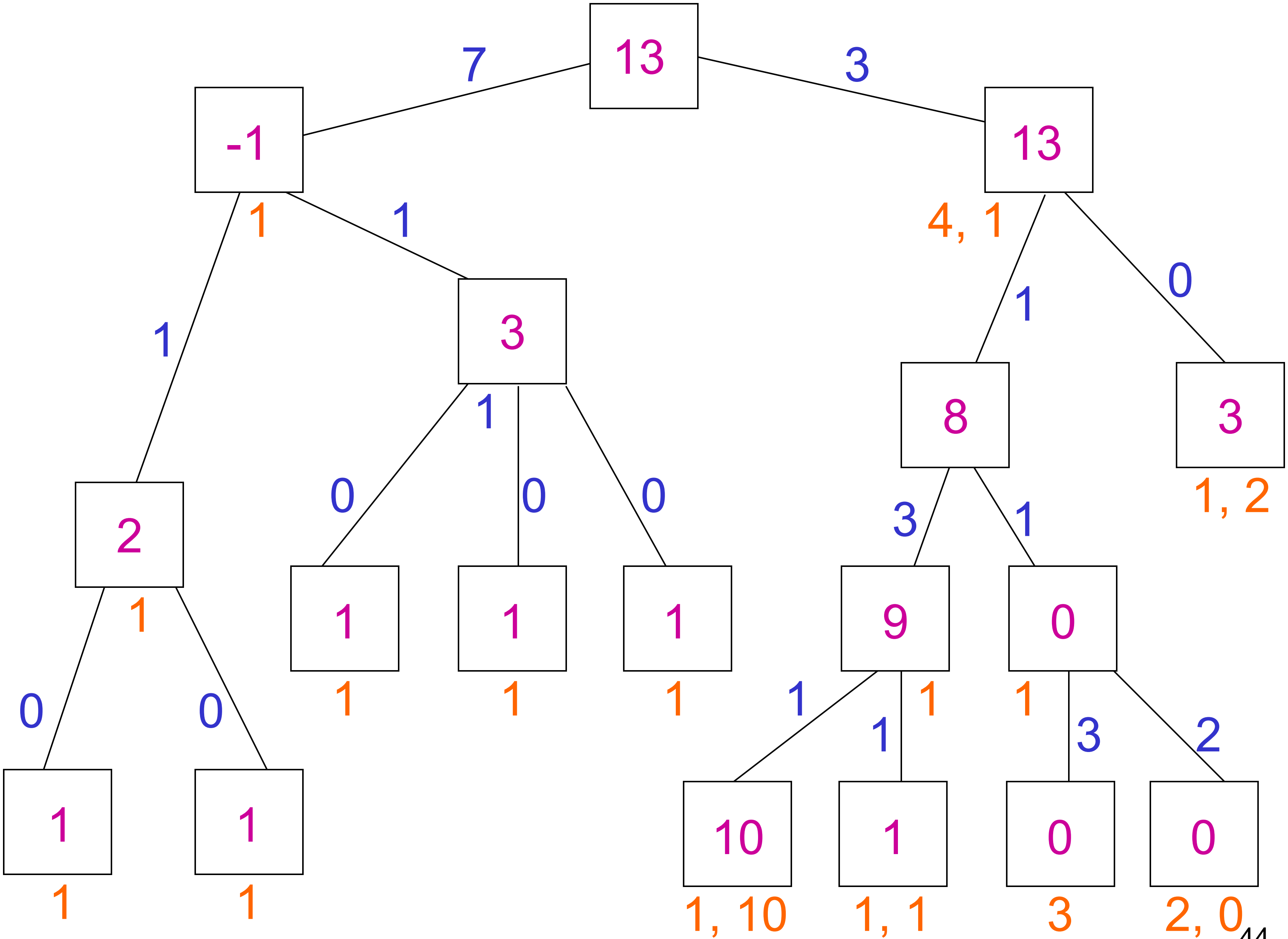
$\mathrm{Ch}(\alpha) \triangleq$ children of α in $T(P)$

$\mathrm{res}(\alpha) \triangleq$ all users “resident” at node α

$\mathrm{loc}(i) \triangleq$ node at which user i is “located”

Bottom-Up Traversal of $T(P)$

```
$\forall \alpha$, after receiving $W^{\beta}$, $\forall \beta \in \mathrm{Ch}(\alpha)$:
{
    COMPUTE $W^{\alpha}$
    IF $W^{\alpha} \geqslant 0$, $\sigma_i \leftarrow 1 \ \ \forall i \in \mathrm{res}(\alpha)$
    ELSE $\sigma_i \leftarrow 0 \ \ \forall i \in \mathrm{res}(\alpha)$
    SEND $W^{\alpha}$ TO $\mathrm{parent}(\alpha)$
}
```


13
7
3
-1
1
1
13
4, 1
1
0
1
3
8
3
1, 2
2
1
0
0
0
1
1
1
1
1
1
1
3
1
9
0
0
0
1
1
1
1
1
1
1
1
3
2
10
1, 10
1
1, 1
0
3
0
2, 0

Computing Cost Shares

$p_i \equiv v_i - (W - W^{-i})$

Case 1: No difference in trees.

Welfare Difference = v_i

Cost Share = 0

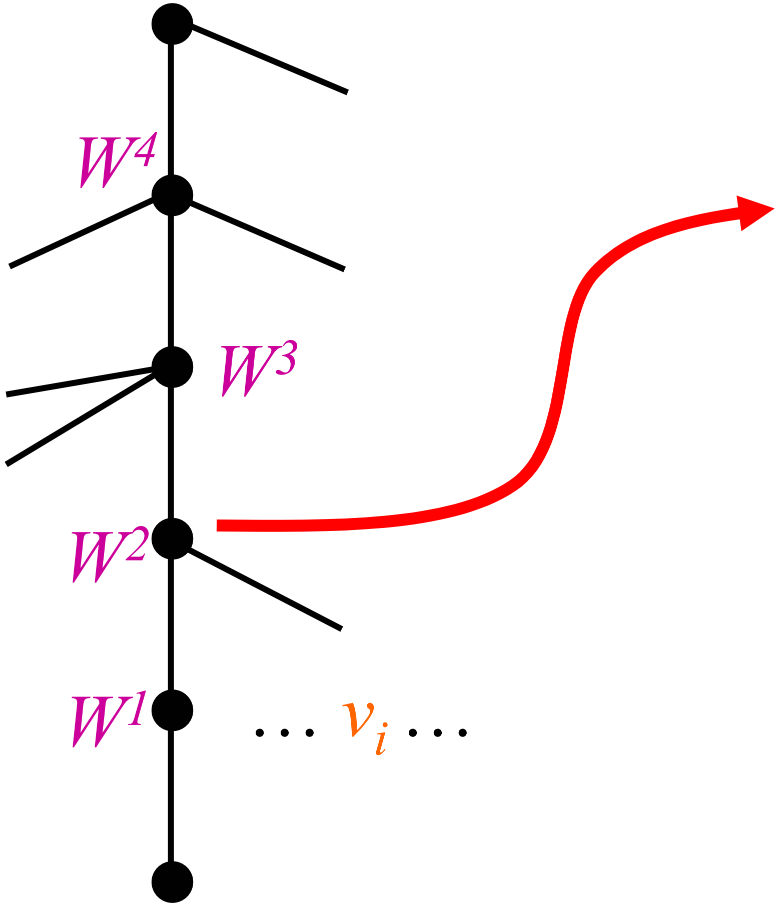
Case 2: Trees differ by 1 subtree.

Welfare Difference = W^{γ}

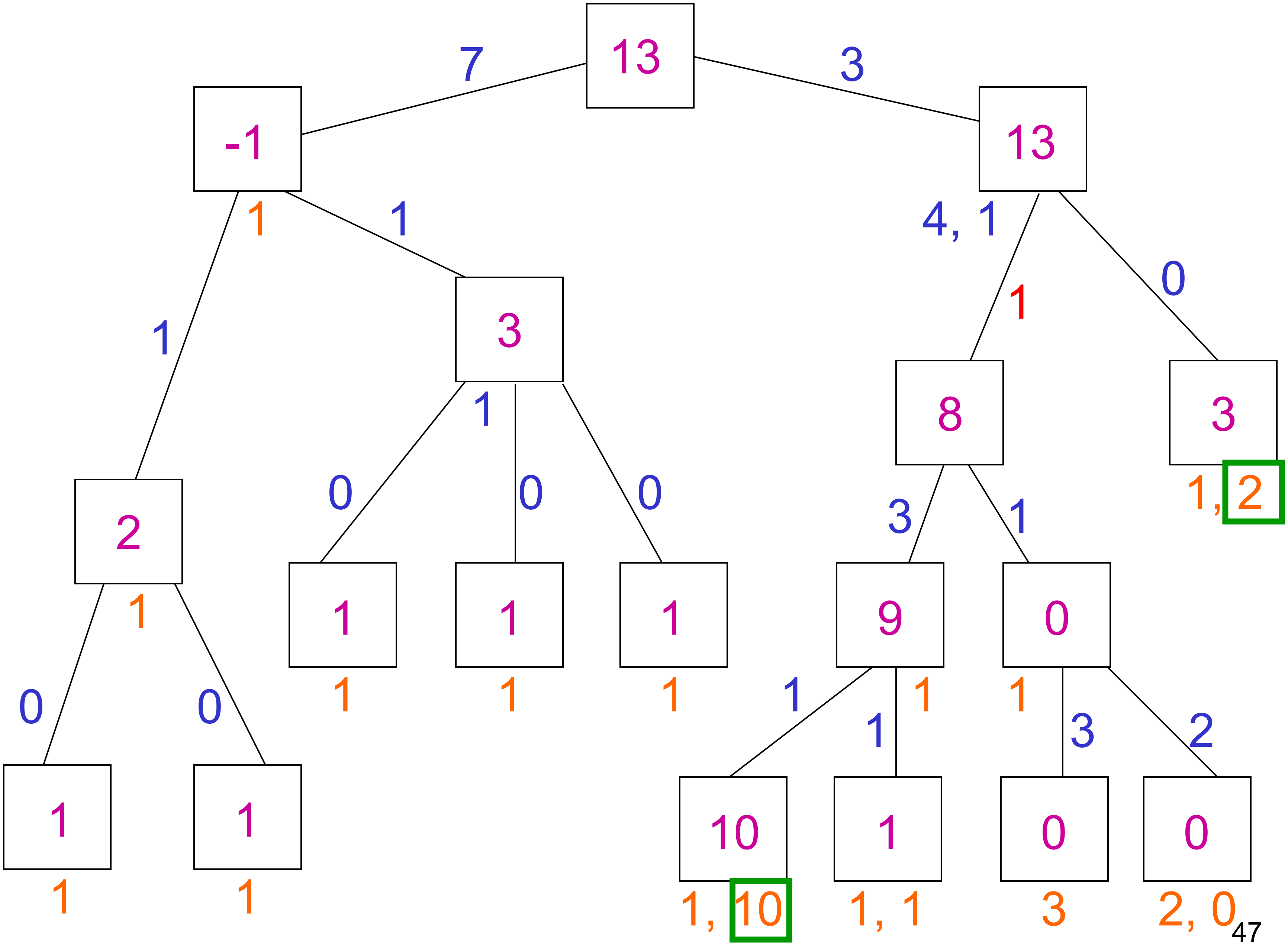
($\gamma \equiv$ minimum welfare anc. of $\mathrm{loc}(i)$)

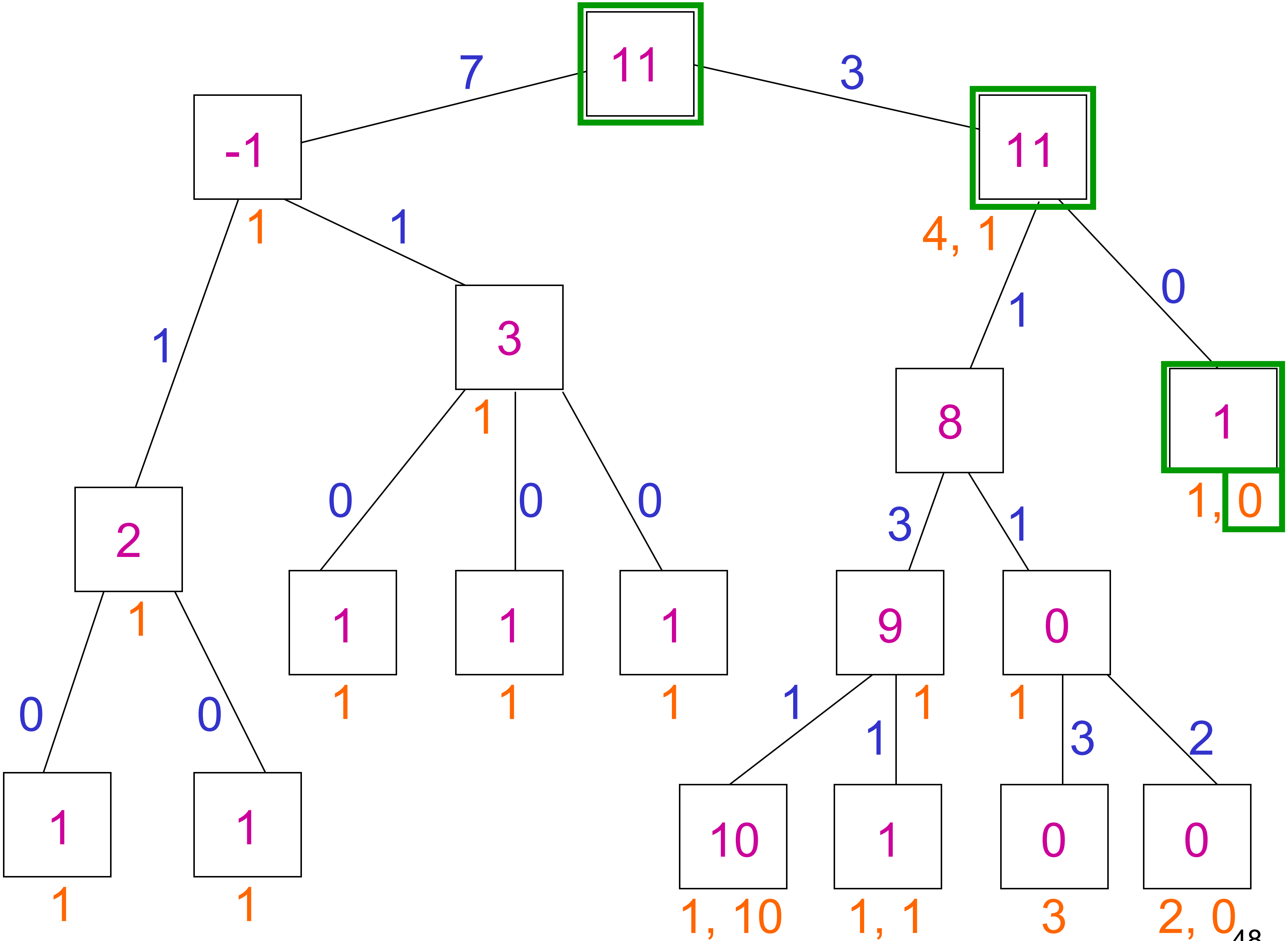
Cost Share = $v_i - W^{\gamma}$

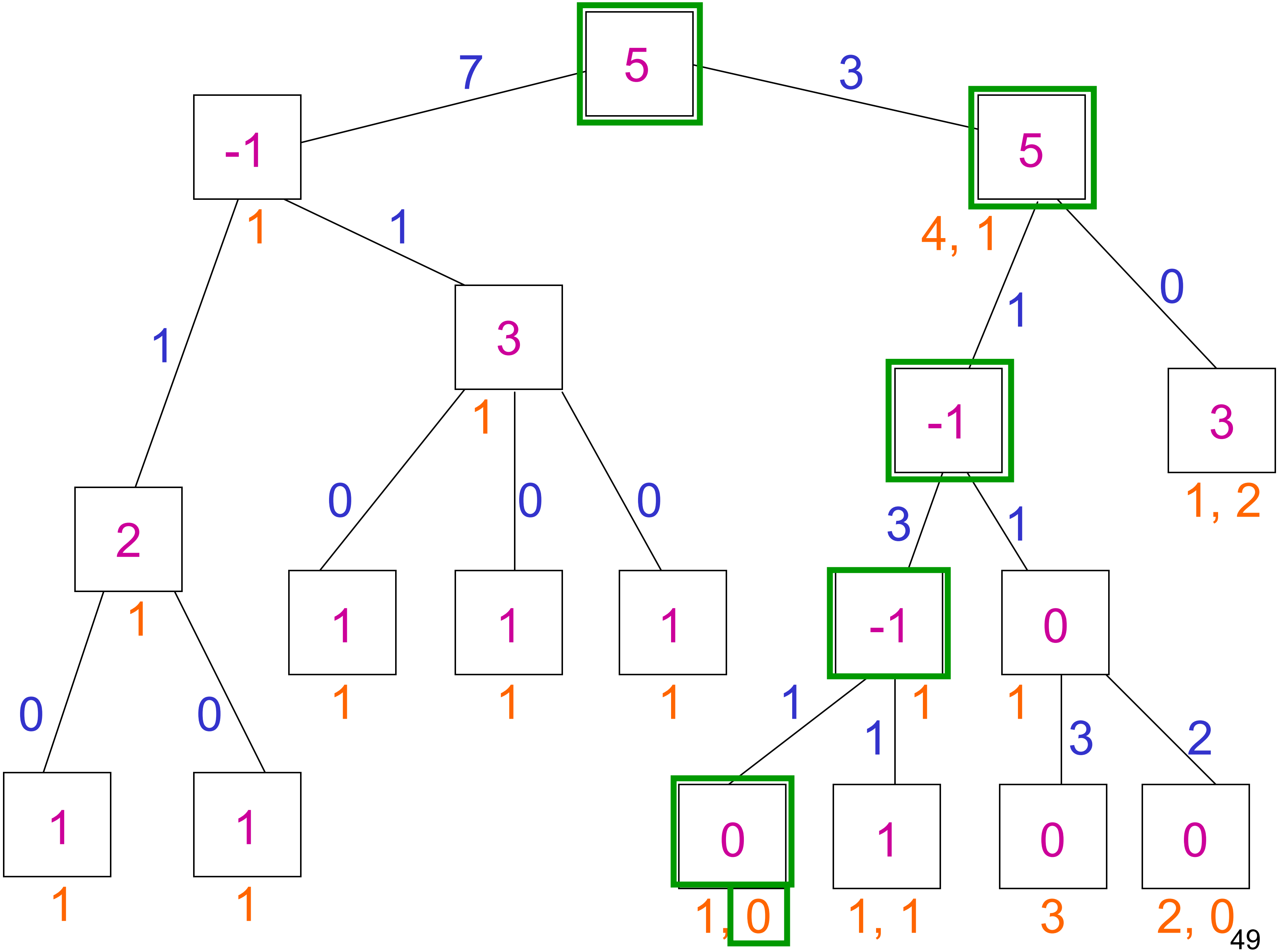
Need Not Recompute W for each $i \in P$



Subtract v_i or W^1 from W^2?


13
7
3
-1
13
1
1
4, 1
1
0
1
3
8
3
1
1, 2
2
0
0
0
3
1
1
1
1
1
1
9
0
1
1
1
1
1
0
0
1
1
1
1
3
2
10
1
0
0
1, 10
1, 1
3
2, 0


11
7
3
-1
11
1
1
4, 1
0
1
3
1
1
8
1
2
0
0
0
1, 0
3
1
1
1
1
1
9
0
1
1
1
1
1
1
1
1
0
0
1
1
3
2
1
1
10
1
0
0
1
1
1, 10
1, 1
3
2, 0

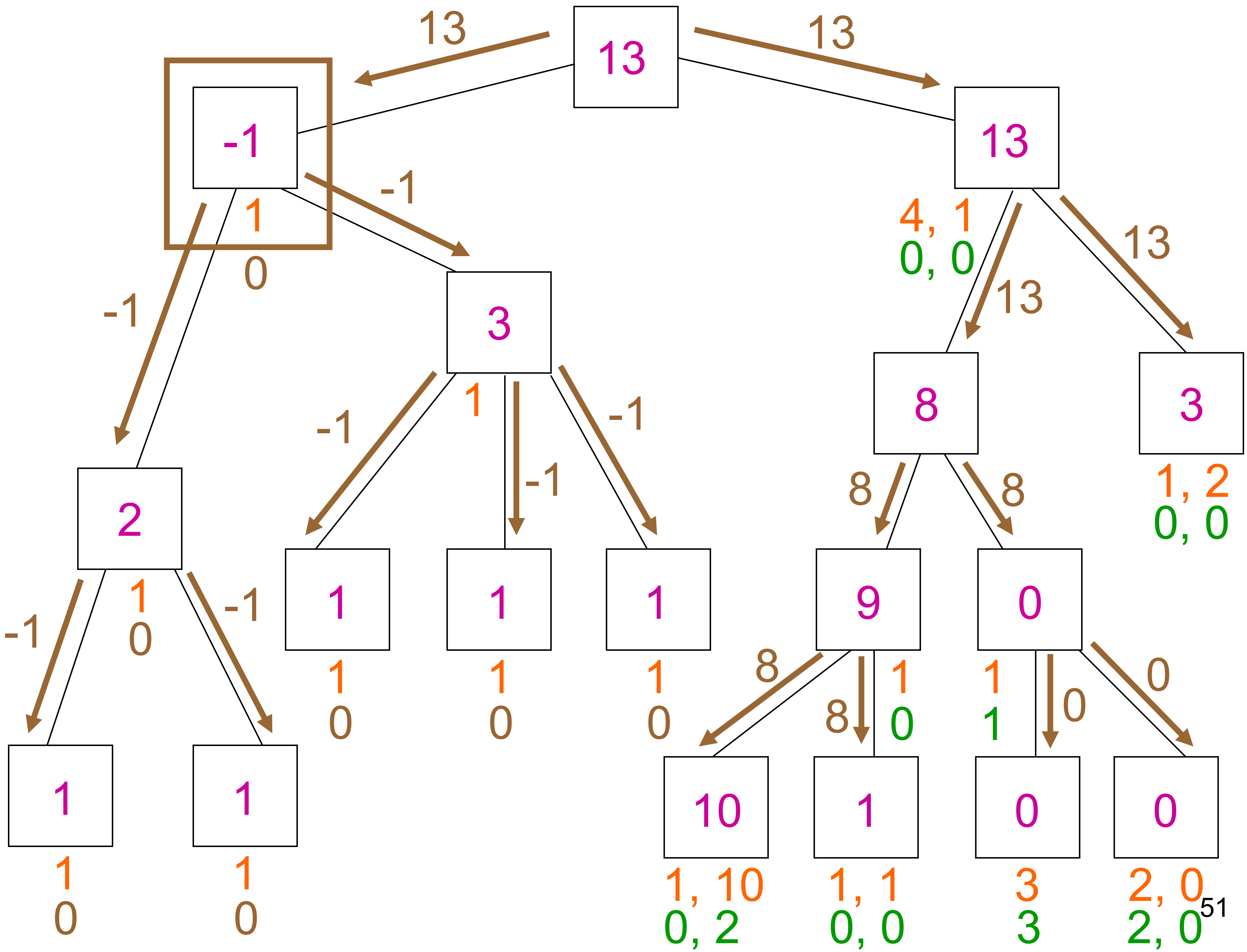

5
7
3
-1
5
1
1
4, 1
0
1
3
1
-1
3
1
0
0
0
1, 2
2
3
1
1
1
1
-1
0
1
1
1
1
1
1
0
0
1
1
3
2
1
1
0
1
0
0
1
1
1, 0
1, 1
3
2, 0

Top-Down Traversal of $T(P)$

(Nodes have “state” from bottom-up traversal)

Init: Root α_s sends W^{α_s} to $\mathrm{Ch}(\alpha_s)$

```
$\forall \alpha \neq \alpha_s$, after receiving $A$ from $\mathrm{parent}(\alpha)$ :
  IF $\sigma_i = 0, \forall i \in \mathrm{res}(\alpha)$, OR $A < 0$ {
      $p_i \leftarrow 0 \wedge \sigma_i \leftarrow 0, \forall i \in \mathrm{res}(\alpha)$
      SEND $-1$ TO $\beta, \forall \beta \in \mathrm{Ch}(\alpha)$ }
  ELSE {
      $A \leftarrow \min(A, W^{\alpha})$
      FOR EACH $i \in \mathrm{res}(\alpha)$
          IF $v_i \leq A, p_i \leftarrow 0$
          ELSE $p_i \leftarrow v_i - A$
      SEND $A$ TO $\beta, \forall \beta \in \mathrm{Ch}(\alpha)$ }
```


13
13
13
-1
13
1
0
-1
-1
3
4, 1
0, 0
13
13
8
3
1, 2
0, 0
2
1
0
-1
-1
-1
1
-1
-1
1
1
1
1
0
1
0
1
0
8
8
9
0
8
8
1
0
1
1
0
0
1
1
1
0
1
0
10
1
0
0
1, 10
0, 2
1, 1
0, 0
3
3
2, 0
2, 0

Profit Maximization [FGHK02]

Mechanism:

- Treat each node as a separate “market.”
- Clearing prices approx. maximize revenue.
- Find profit-maximizing subtree of markets.
- Satisfies NPT and VP but not CS or SYM.

Properties:

- Strategyproof and $O(1)$ messages per link
- Expected constant fraction of maximum profit if
  - maximum profit margin is large (> 300%), and
  - there is real competition in each market

Multiple Transmission Rates [AR02]

r = # rates h = tree height K = size of numerical input

One layer per rate ("layered paradigm"):

- MC is computable with three messages per link and $O(rhK)$ bits per link.
- For worst-case instances, average number of bits per link needed to compute MC is $\Omega(rK)$.

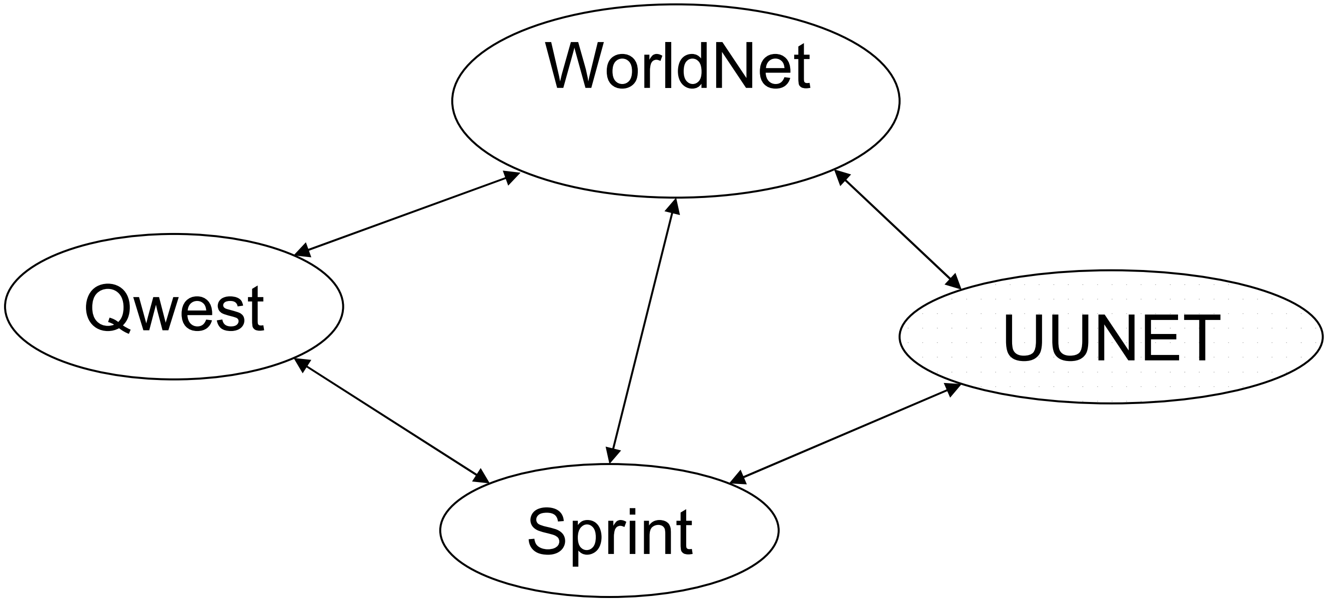
One multicast group per rate ("split-session paradigm"):

- Same MC algorithm has communication and computational complexity proportional to 2^r.
- For variable r, no polynomial-time algorithm can approximate total welfare closely, unless NP=ZPP.

Outline

- Motivation and Background
- Example: Multicast Cost Sharing
- Overview of Known Results
- Three Research Directions
- Open Questions

Interdomain Routing



Agents: Transit ASs

Inputs: Routing Costs or Preferences

Outputs: Routes, Payments

Lowest-Cost Routing

- Agent k's private info: per-packet cost c_k
- Mechanism-design goal: LCPs
- Centralized computation:
  - *P*-time VCG mechanism [NR01]
  - Faster *P*-time VCG mechanism [HS01]
- Distributed computation [FPSS02]:
  - BGP-based algorithm for VCG mechanism
  - All source-destination pairs

Policy-Routing

- Agents have preferences over routes:

$$v_i\colon \{P_{ij}\} \to \Re^{\geq 0}$$

- Goal: routing tree maximizing $\sum_i v_i(P_{ij})$
- Arbitrary preferences [FSS04]:

  NP-hard to approximate w/in factor $O(n^{1/4-\varepsilon})$
- Next-hop preferences [FSS04]:
  - *P*-time (centralized) VCG mechanism
  - No good distributed implementation (dyn.)

Supply-Chain Auctions

- Problem: concurrent auctions where activities must be coordinated across markets
  - Example: Markets for rubber, tires, trucks

- Solution [BN01]: Mechanism that propagates supply and demand curves along the chain
  - Strategyproof and achieves material balance

- Communication complexity:
  - Naïve algorithm sends $\Omega(q)$ prices per link.
  - Use binary search to find traded quantity.
    $\Rightarrow$ $O(\log q)$ prices per link

Spatially Distributed Markets

- Problem: There are multiple markets for a single good, with a cost to transfer the between markets. Find an efficient set of market prices and transfer quantities.

- Solutions [BNP04]:

  1) Mechanism that is efficient and strategyproof

  2) Mechanism that is budget-balanced and strategyproof
- Mechanisms can be computed in polynomial time using a reduction to min-cost flow.

Negotiation-Range Mechanisms

- Classical results in economics show that no strategyproof trade mechanism can be efficient *and* budget-balanced.
- One approach: Mechanism reports a range of prices for each trade, instead of a single price. Then, traders negotiate the final price [BGLM04].
- There is a strategyproof, budget-balanced, and efficient mechanism to match traders and report a price range to each pair [BGLM04].
- Catch: No strategyproof negotiation mechanisms for the second phase

Peer-to-Peer Networks

Distributed rating system [DGGZ03]:

- Constructs “reputation” of each peer
- Prevents lying (strategyproof)

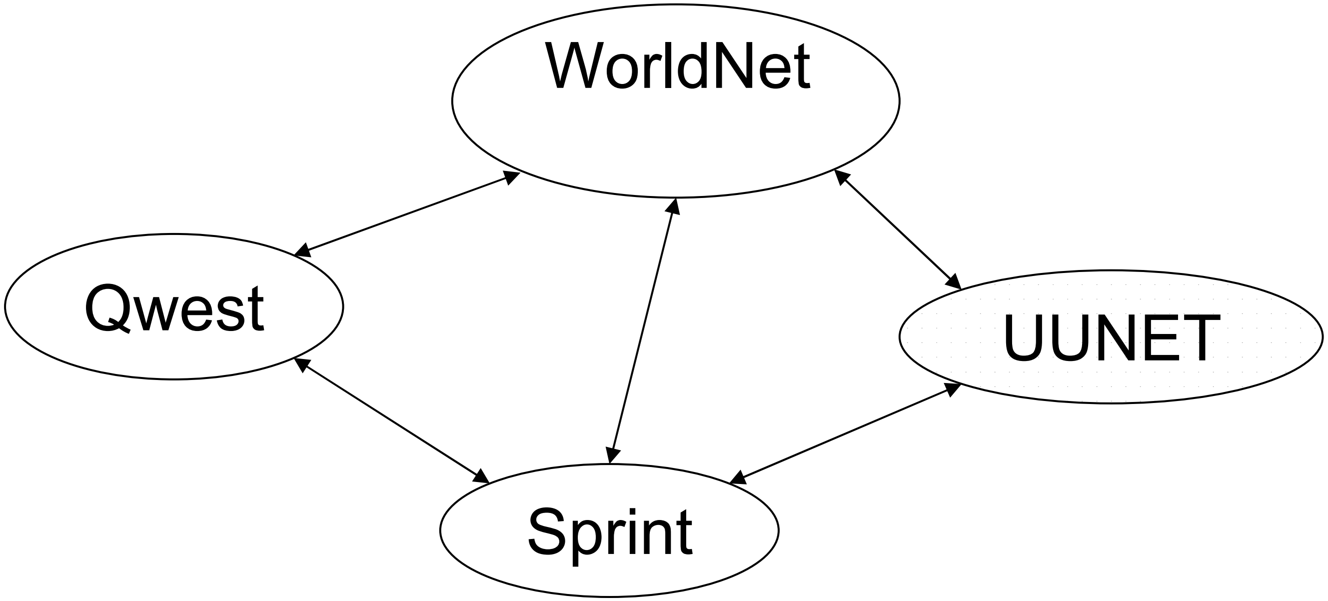
Fair allocation of resources [NWD03]:

- Strategyproof revelation of true usage

Outline

- Motivation and Background
- Example: Multicast Cost Sharing
- Overview of Known Results
- Three Research Directions
  - BGP-based interdomain-routing mechanisms
  - Canonically hard DAMD problems
  - Distributed implementation challenges
- Open Questions

Interdomain-Routing Mechanism-Design Problem



Agents: Transit ASs

Inputs: Routing Costs or Preferences

Outputs: Routes, Payments

Lowest-Cost-Routing MD

Agents' valuations: Per-packet costs $\{c_k\}$

(Unknown) global parameter: Traffic matrix $[T_{ij}]$

Outputs: $\{route(i, j)\}$

Payments: $\{p^k\}$

Objectives:

- Lowest-cost paths (LCPs)
- Strategyproofness
- "BGP-based" distributed algorithm

A Unique VCG Mechanism

Theorem [FPSS02]:

For a biconnected network, if LCP routes are always chosen, there is a unique strategyproof mechanism that gives no payment to nodes that carry no transit traffic. The payments are of the form

$$p^k = \sum_{i,j} T_{ij}\, p_{ij}^k ,$$ where

$$p_{ij}^k = c_k + \text{Cost}\left(P^{-k}(c;\, i, j) \right) - \text{Cost}\left(P(c;\, i, j) \right)$$

Proof is a straightforward application of [GL79].

Features of this Mechanism

- Payments have a very simple dependence on traffic $[T_{ij}]$: Payment p^k is weighted sum of per-packet prices p^k_{ij} .
- Cost c_k is independent of i and j, but price p^k_{ij} depends on i and j.
- Price p^k_{ij} is 0 if k is not on LCP between i, j.
- Price p^k_{ij} is determined by cost of min-cost path from i to j not passing through k (min-cost "k-avoiding" path).

BGP-Based Computational Model (1)

- Follow abstract BGP model of [GW99]:

  Network is a graph with nodes corresponding to ASs and bidirectional links; intradomain-routing issues are ignored.

- Each AS has a routing table with LCPs to all other nodes:

Dest.	LCP				LCP cost
AS1	AS3	AS5	AS1		3
AS2	AS7	AS2			2

Entire paths are stored, not just next hop.

Computational Model (2)

- An AS "advertises" its routes to its neighbors in the AS graph, whenever its routing table changes.

- The computation of a single node is an infinite sequence of stages:

Receive routes from neighbors

Update routing table

Advertise modified routes

- Complexity measures:
  - Number of stages required for convergence
  - Total communication

★Surprisingly *scalable* in practice.

Computing the VCG Mechanism

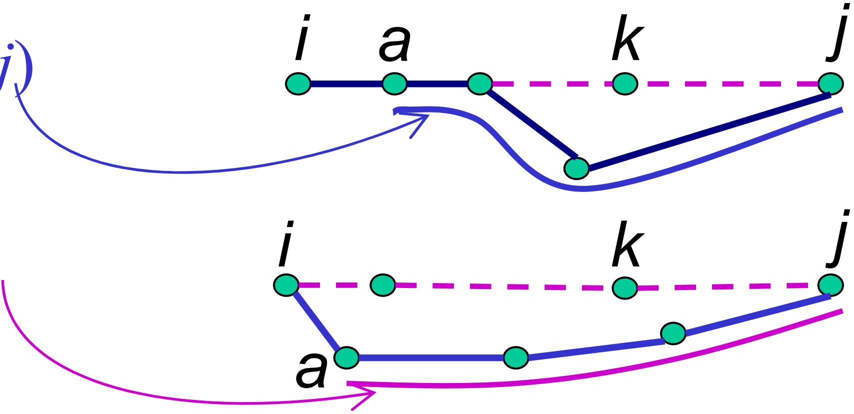
- Need to compute *routes* and *prices*.
- Routes: Use Bellman-Ford algorithm to compute LCPs and their costs.
- Prices:

$$p_{ij}^{k} = c_k + \boxed{\mathrm{Cost}\left(P^{-k}(c;\, i,\, j)\right)} - \mathrm{Cost}\left(P(c;\, i,\, j)\right)$$

⇒ Need algorithm to compute cost of min-cost k-avoiding path.

Structure of k-avoiding Paths

- BGP uses communication between neighbors only
  ⇒ we need to use "local" structure of $P^{-k}(c;\ i,j)$.
- Tail of $P^{-k}(c;\ i,j)$ is either of the form



(1) $P^{-k}(c;\ a,j)$

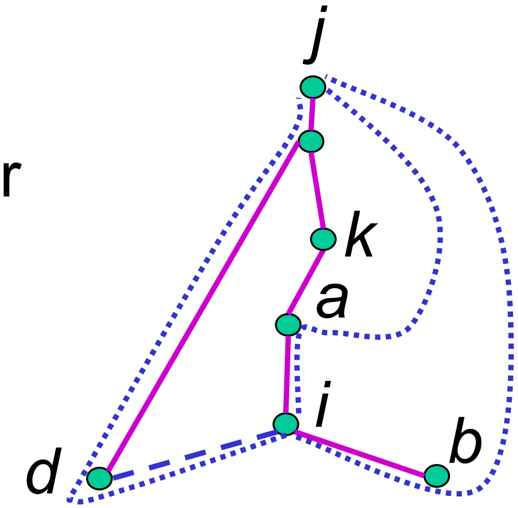
or (2) $P(c;\ a,j)$

- Conversely, for each neighbor a, either $P^{-k}(c;\ a,j)$ or $P(c;\ a,j)$ gives a candidate for $P^{-k}(c;\ i,j)$.

Computing the Prices

- Classifying neighbors:
  - Set of LCPs to j forms a tree.
  - Each of i's neighbors is either
    - (*a*) parent
    - (*b*) child
    - (*d*) unrelated

  in tree of LCPs to j.



- Each case gives a candidate value for p_{ij}^{k} based on neighbor's LCP cost or price, *e.g.*,

  (b) $p_{ij}^{k} \leq p_{bj}^{k} + c_b + c_i$

- p_{ij}^{k} is the minimum of these candidate values

  $\Rightarrow$ compute it locally with dynamic programming.

A “BGP-Based” Algorithm

Dest.	cost	LCP and path prices				LCP cost
AS1		AS3	AS5	AS1		$c(i,1)$
	c_1	p_{i1}^{3}	p_{i1}^{5}			

1. LCPs are computed and advertised to neighbors.
2. Initially, all prices are set to ∞.
3. In the following stages, each node repeats:
   - Receive LCP costs and path prices from neighbors.
   - Recompute candidate prices; select lowest price.
   - Advertise updated prices to neighbors.

Final state: Node i has accurate p_{ij}^{k} values.

Performance of Algorithm

$$d = max_{i,j} \,||\, P(c;\, i, j) \,||$$

$$d' = max_{i,j,k} \,||\, P^{-k}(c;\, i, j) \,||$$

Theorem [FPSS02]:

This algorithm computes the VCG prices correctly, uses routing tables of size $O(nd)$ (a constant factor increase over BGP), and converges in at most $(d + d')$ stages (worst-case additive penalty of d' stages over the BGP convergence time).

Dealing with Strategic Computation

- Restoring strategyproofness: Cost c_k must be the only path information that AS k can manipulate.
- Possible because all other information reported by AS k is known to at least one other party, hence not "private" information of AS k.
- Solution [MSTT]: All information is signed by originating party.
  cost c_i: signed by AS i.
  existence of link ij: signed by AS i and AS j.
  AS k's message has to include all relevant signatures.
- AS k cannot benefit by suppressing real paths to k.

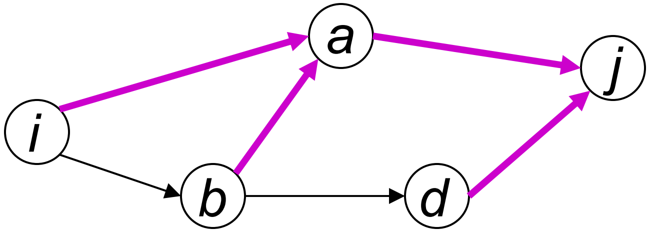
Modified BGP-Update Messages

Update from AS k to AS j for route to AS1:

Dest.	cost	LCP and path prices			LCP cost
AS1		AS3	AS5	AS1	$c(k,1)$
		p_{k1}^{3}	p_{k1}^{5}		
	c_k	c_3	c_5		
	$s_k(c_k)$	$s_3(c_3)$	$s_5(c_5)$		
	$s_k(l_{kj})$	$s_3(l_{3k})$	$s_5(l_{53})$	$s_1(l_{15})$	

General Policy-Routing Problem Statement

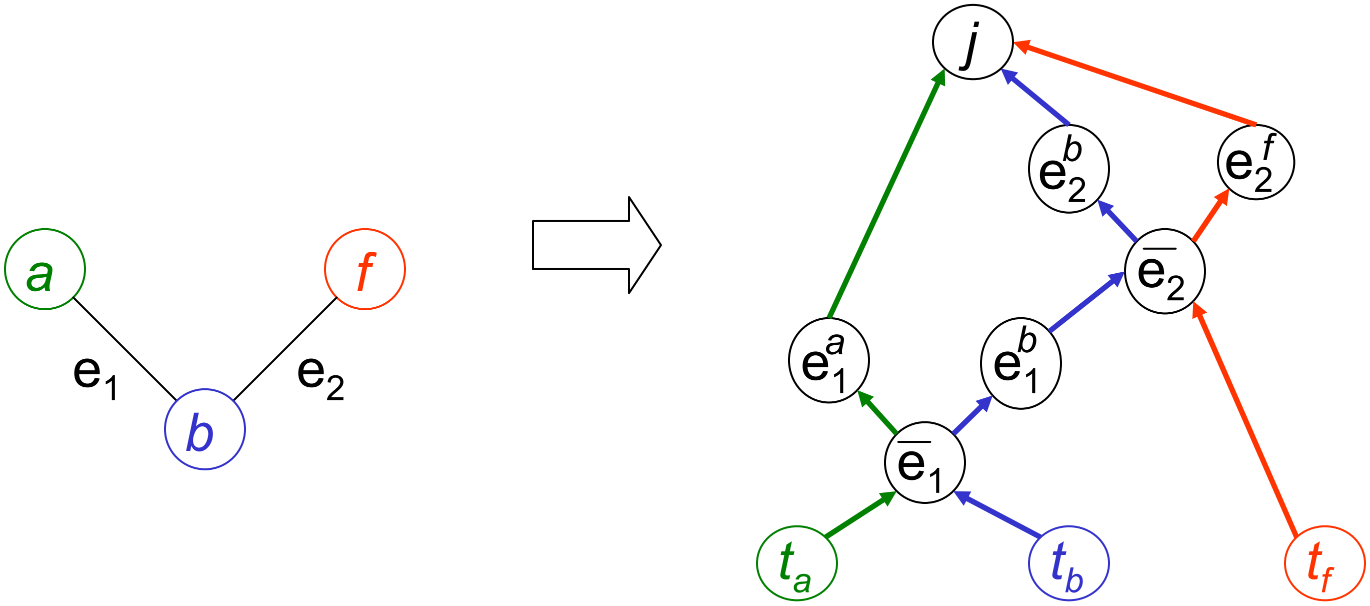
- Consider each destination j separately.
- Each AS i assigns a value $v_i(P_{ij})$ to each potential route P_{ij}.



- Mechanism-design goals:
  - Maximize $W = \sum_i v_i(P_{ij})$.
  - For each destination j, $\{P_{ij}\}$ forms a tree.
  - Strategyproofness
  - BGP-based distributed algorithm

NP-Hardness with Arbitrary Valuations

- Approximability-preserving reduction from Independent-set problem:



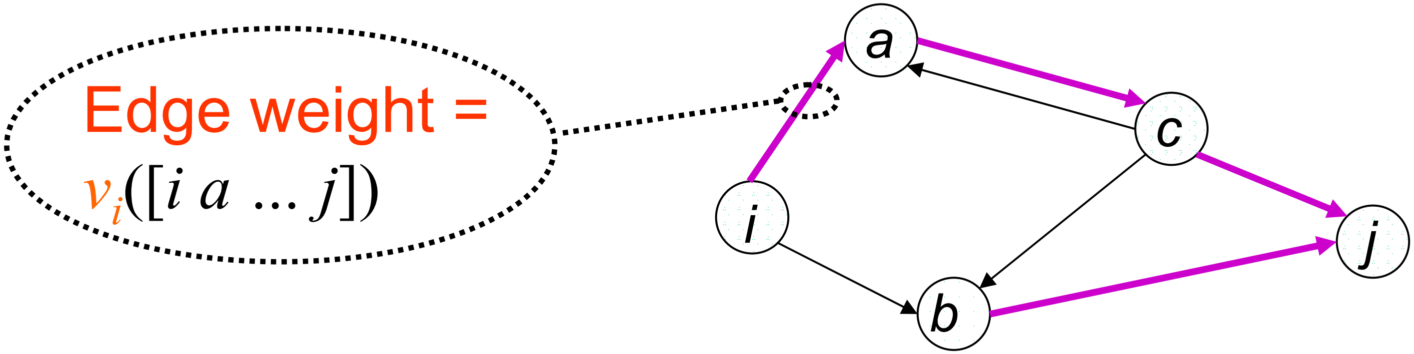
Paths from terminals t_a, t_b, t_f have valuation 1, all other paths 0.

- NP-hard to compute maximum W exactly.
- NP-hard to compute $O(n^{1/4-\varepsilon})$ approximation to maximum W.

Next-Hop Preferences

- $v_i(P_{ij})$ depends only on first-hop AS a.
- Captures preferences due to customer/provider/peer agreements.

For each destination j , optimal routing tree is a Maximum-weight Directed Spanning Tree (MDST):



Edge weight = $v_i([i\ a\ \dots\ j])$

Strategyproof Mechanism

Let

T^* = Maximum weight directed spanning tree (MDST) in G

T^{-i} = MDST in $G - \{i\}$

- For biconnected networks, there is a unique strategyproof mechanism that always picks a welfare-maximizing routing tree and never pays non-transit nodes. The payments required for this mechanism are

$$p^i = W(T^*) - v_i(T^*) - W(T^{-i})$$

- Routes and payments can be computed in polynomial time (in a centralized computational model).

Proving Hardness for "BGP-Based" Routing Mechanisms [FSS04]

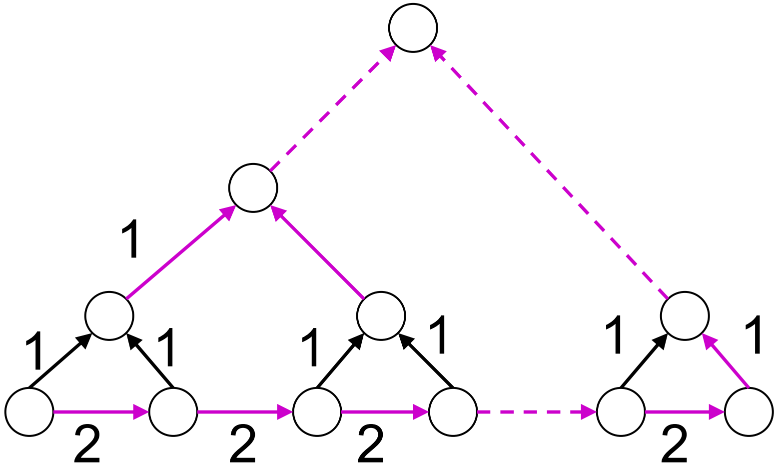
- Need to formalize requirements for "BGP compatibility."
- Hardness results need only hold for:
  - "Internet-like" graphs
    - $O(1)$ average degree
    - $O(\log n)$ diameter and $O(\log n)$ diameter$'$
  - An open set of numerical inputs in a small range

Reasonable Routing-Table Size and Convergence Time

- Each AS uses $O(l)$ space for a route of length l.
- Length of longest routes chosen (and convergence time) should be proportional to network diameter or diameter′.
- See related work on formal models of “path-vector” routing protocols [GJR03].

Long Paths Chosen by MDST [FSS04]

- Example:



- Don’t even know how to compute MDST prices in time proportional to length of longest route chosen.

Reasonably Stable Routing Tables

- Most changes should not affect most routes.

- More formally, there are $o(n)$ nodes that can trigger $\Omega(n)$ update messages when they fail or change valuations.

MDST Does **Not** Satisfy the Stability Requirement [FSS04]

Proof outline:

(i) Construct a network and valuations such that, for $\Omega(n)$ nodes i, T^{-i} is disjoint from the MDST T^*.

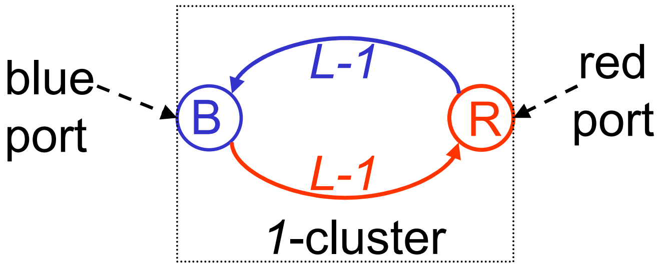
(ii) A change in the valuation of any node a may change

$$p_i = W(T^*) - v_i(T^*) - W(T^{-i}).$$

(iii) Node i (or whichever node stores p_i) must receive an update when this change happens.
$\Rightarrow$ $\Omega(n)$ nodes can each trigger $\Omega(n)$ update messages.

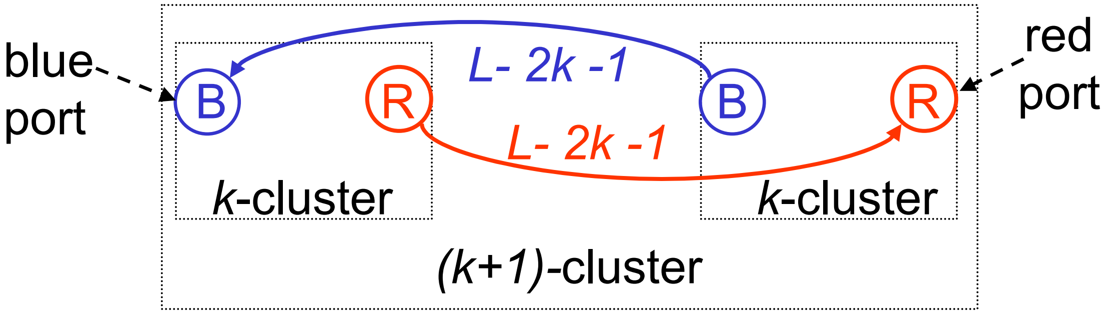
Network Construction (1)

(a) Construct *1-cluster* with two nodes:



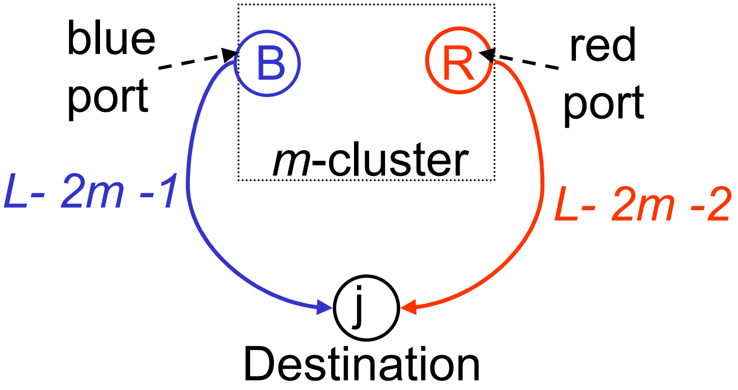
$L \approx 2 \log n + 4$

(b) Recursively construct *(k+1)*-clusters:

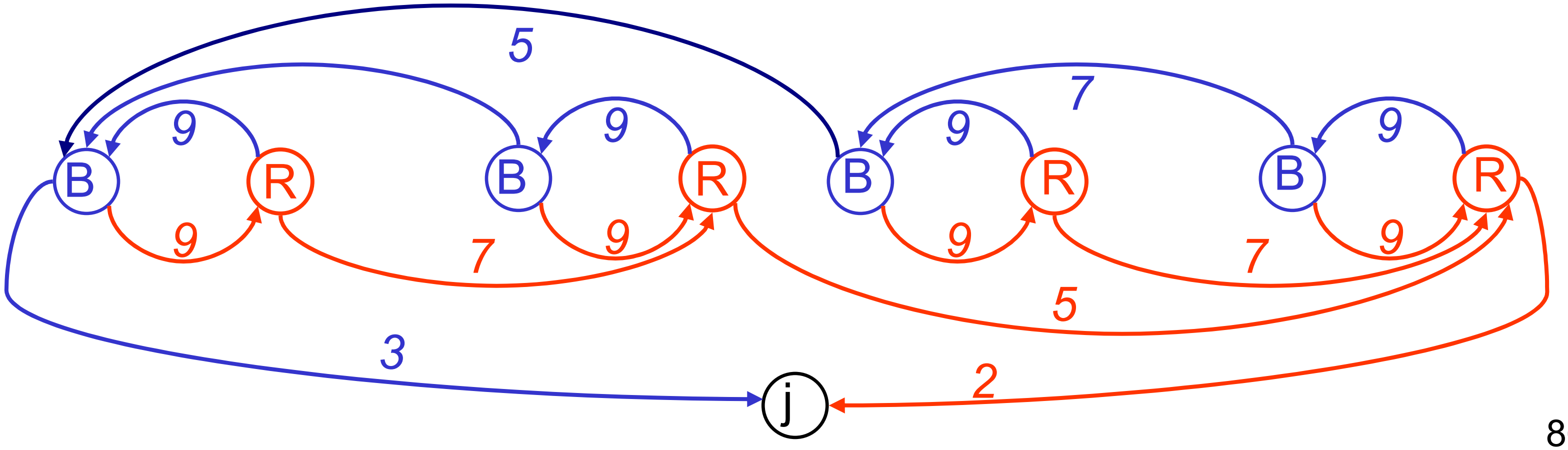


Network Construction (2)

(c) Top level: *m*-cluster with $n = 2^m + 1$ nodes.



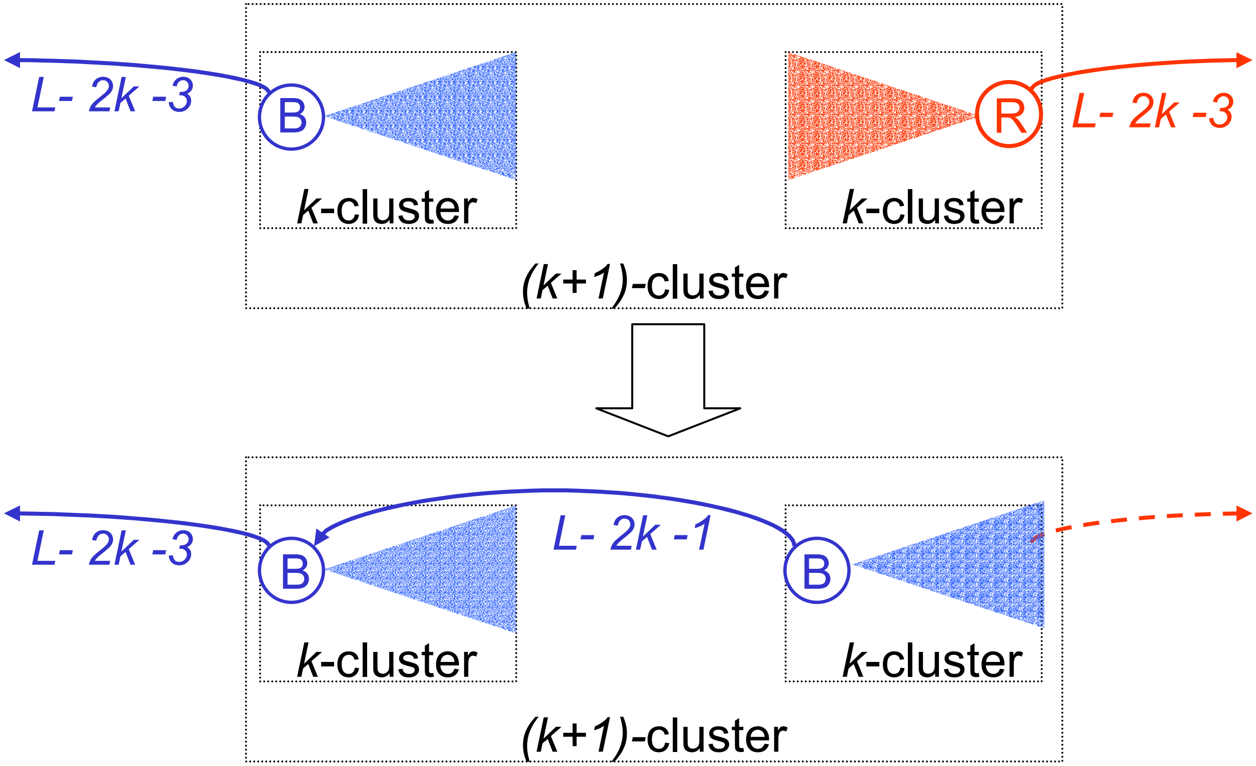
Final network (*m* = *3*):

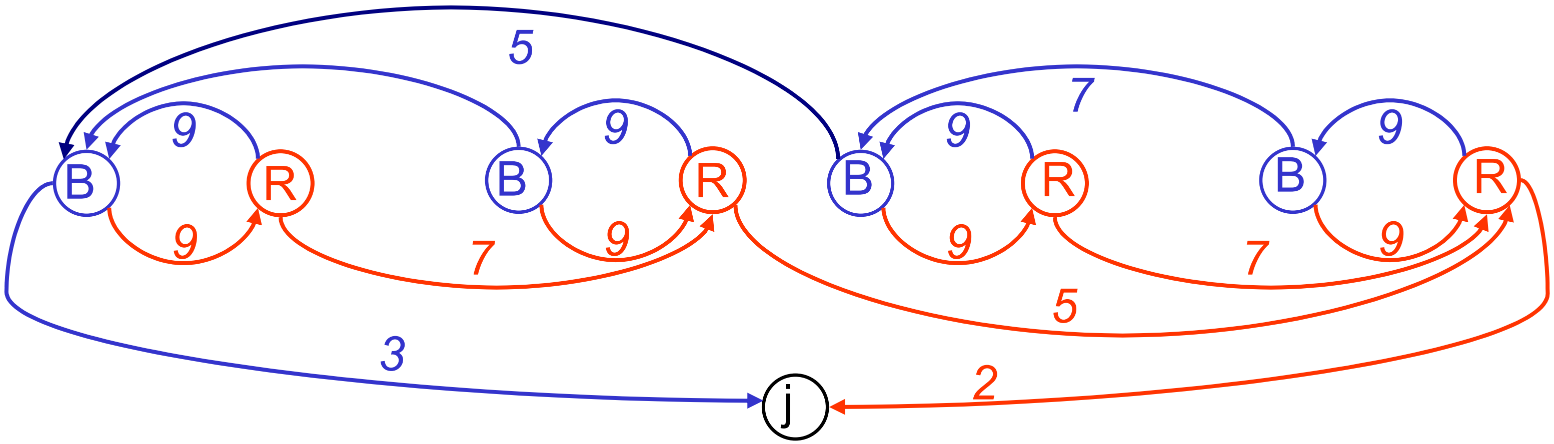


Optimal Spanning Trees

Lemma: *W*(*blue tree*) = *W*(*red tree*) + 1 ≥ *W*(*any other sp.tree*) + 2

Proof: If a directed spanning tree has red and blue edges, we can increase its weight by at least 2:





- MDST T^* is the blue spanning tree.
- For any blue node B, T^{-B} is the red spanning tree on $N - \{B\}$.
- A small change in any edge, red or blue, changes

$$p^B \quad = \quad W(T^*) - v_B(T^*) - W(T^{-B})$$

⇒ Any change triggers update messages to all blue nodes!

Alternative Policy Class: Subjective Costs

- AS *i* assigns a cost $c_i(k)$ to AS *k*.

  AS *i*'s subjective cost for route P_{ij} is

  $C_i(P_{ij}) = \Sigma_{k \in P_{ij}} c_i(k)$
- Overall goal: minimize total subjective cost to destination = $\Sigma_i C_i(P_{ij})$
- Natural generalization of Lowest-Cost Routing
- Expresses a broad range of policies.
- Question: Which subclasses of Subjective-Cost

  Policies lead to strategyproof, BGP-based mechanisms?

Forbidden-Set Policies

- AS *i* has a set S_i of ASes it does not want to route through.
- Goal: Find a routing tree in which no AS *i* uses a route through any AS in S_i.
- *0-1* subjective cost model:

$$c_i(k) = 1 \quad \text{if } k \in S_i$$

$$c_i(k) = 0 \quad \text{if } k \notin S_i$$

- Theorem [FMKS]: It is NP-hard to find a routing tree that even approximately minimizes total subjective cost, within any factor.

1-2 Subjective costs

- Restricted subclass of subjective-cost policies with $c_i(k) \in \{1,2\}$ *for all i,k.*
- It is NP-hard to find a minimum subjective-cost routing tree with *1-2* subjective costs [FKMS].
- It is also APX-hard, *i.e.*, $(1+\varepsilon)$-approximation is hard.
- Easy *2*-approximation: Shortest path tree
- This approximation does not use private information at all. ⇒ No interesting mechanism design problem.

Question: Can we do better than *2*-approximation with a non-trivial approximation algorithm?

Open Questions about Subjective-Cost Routing

- ASes "almost" agree about the cost of node *k*:

  Subjective costs are randomly distributed about an (unknown) objective value.

  Question: How does the hardness change with the degree of subjectivity?
- Differences in cost arise because ASes value different objective metrics (*e.g.*, length *vs.* reliability).
-
-

Open Questions

- BGP-compatible special case of next-hop-preferences routing
- Fully fleshed-out BGP-based computational model
  - Incremental computation
  - "Smooth" convergence?
- New DA principle: Use an Internet protocol as a "computational substrate."

Outline

- Motivation and Background
- Example: Multicast Cost Sharing
- Overview of Known Results
- Three Research Directions
  - BGP-based interdomain-routing mechanisms
  - Canonically hard DAMD problems
  - Distributed implementation challenges
- Open Questions

“Hard to Solve on the Internet”

Intuitively, this means

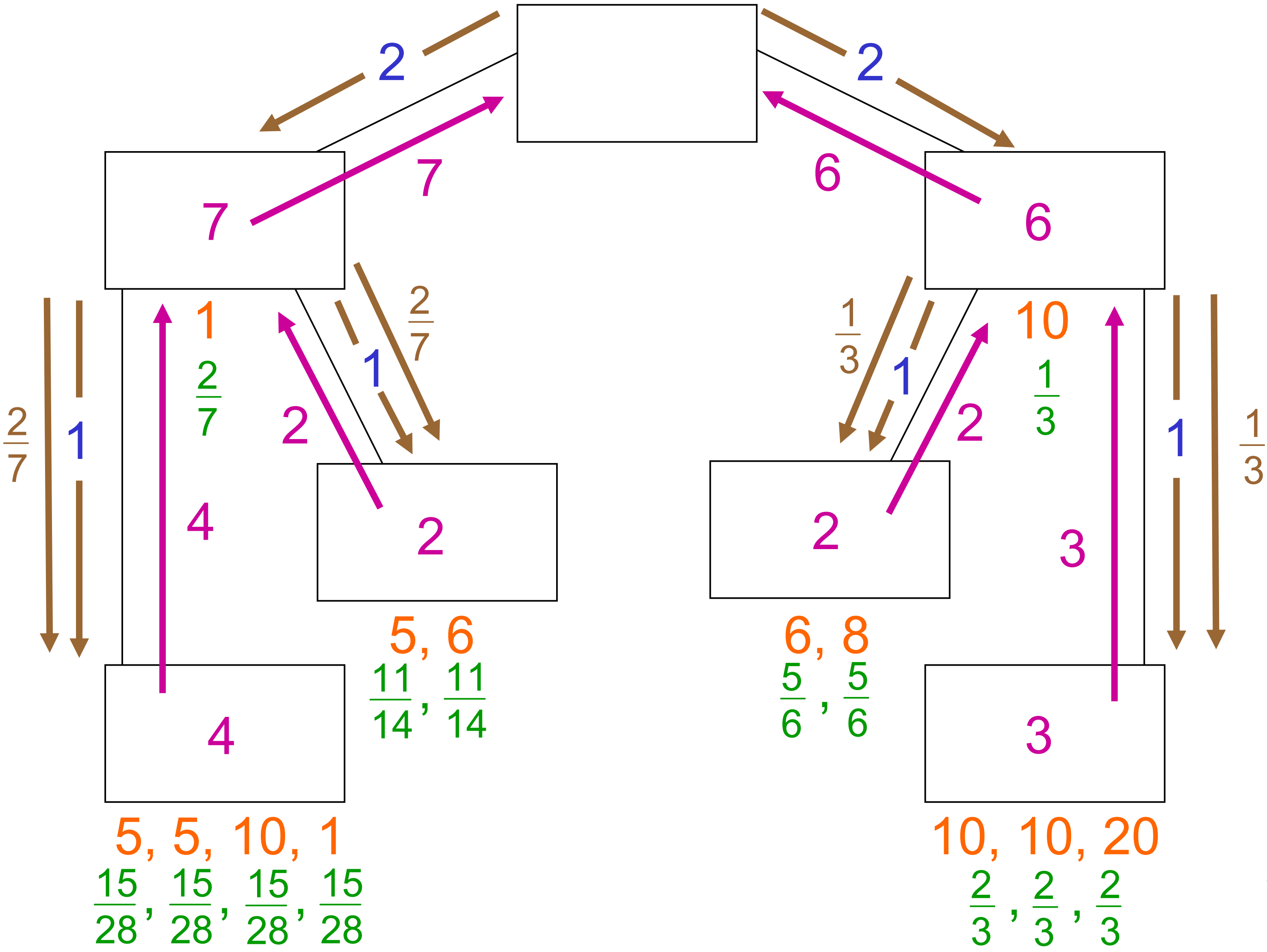
- Cannot simultaneously achieve
  - Robust scalability
  - Incentive compatibility
- Can achieve either requirement separately

Recall that BB multicast cost sharing is hard.

Scalability $\triangleq$ low (absolute) network complexity

Incentive compatibility $\triangleq$ GSP’ness

GSP’ness Without Scalability



Iterative SH Algorithm

- Start with $R = P$.
- Calculate cost shares as above.
- Eliminate from R all i s.t. current $p_i > v_i$.
- Repeat until $R \neq \varnothing$ or no i eliminated.

Worst case: $|P|$ iterations.

Lower bound in [FKSS03] shows that bad network complexity is unavoidable.

Scalability Without GSP'ness

Bottom-up pass: Compute

$$C = \sum_{l \in L} c(l) \quad \text{and} \quad V = \sum_{i \in P} v_i$$

Top-down pass:

If $C > V$, $\sigma_i = 0$ for all i

If $C \leq V$, $\sigma_i = 1$ for all i

and $p_i = (v_i \cdot C) / V$

Open Question

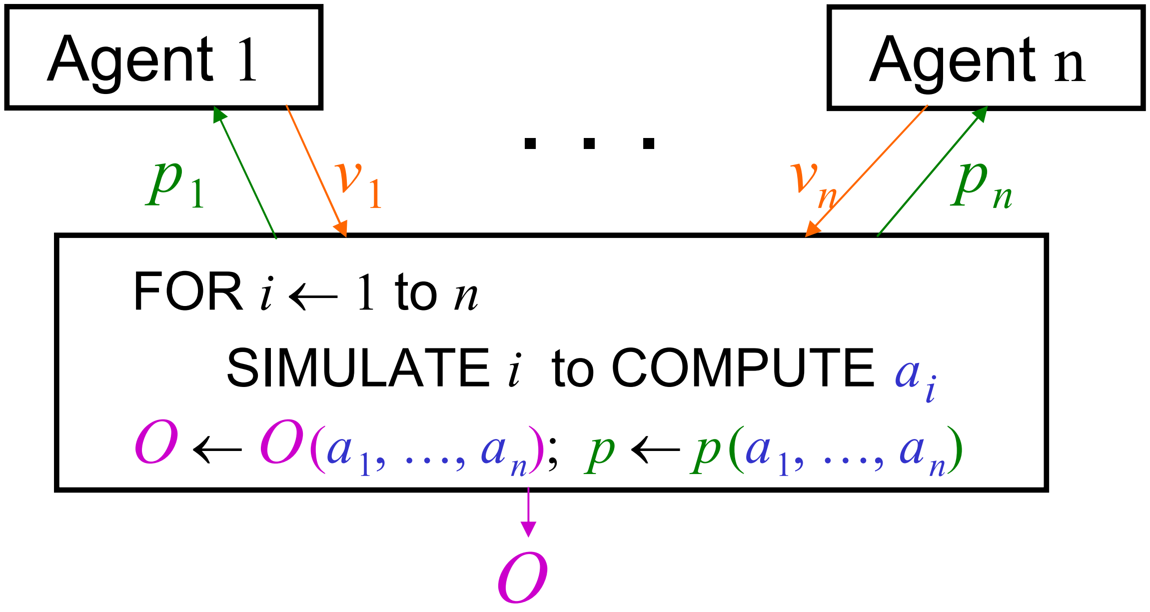
- More canonically hard problems?
- Open for *centralized* AMD as well
- Complexity theory of Internet computation
  - Formal models
  - Complexity classes
  - Reductions

Outline

- Motivation and Background
- Example: Multicast Cost Sharing
- Overview of Known Results
- Three Research Directions
  - BGP-based interdomain-routing mechanisms
  - Canonically hard DAMD problems
  - Distributed implementation challenges
- Open Questions

Revelation Principle

If there is a DS mechanism (O, p) that implements a design goal, then there is one that does so truthfully.



```
FOR $i \leftarrow 1$ to $n$
    SIMULATE $i$ to COMPUTE $a_i$
$O \leftarrow O(a_1, \ldots, a_n)$; $p \leftarrow p(a_1, \ldots, a_n)$
```

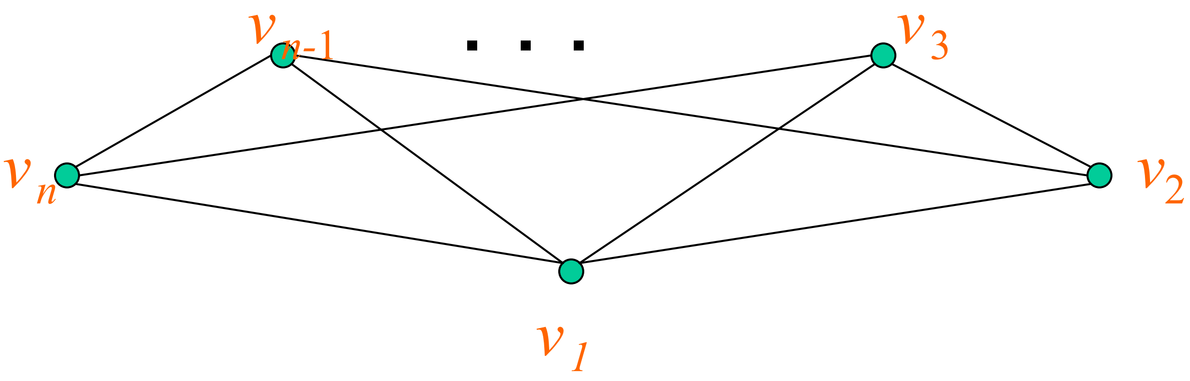
Note: Loss of privacy
Shift of computational load
Assumes centralized, obedient mechanism

Is Truthtelling Really “Dominant”?

Consider Lowest-Cost Routing:

- Mechanism is strategyproof, in the technical sense: Lying about its cost cannot improve an AS’s welfare *in this particular game*.
- But truthtelling reveals to competitors information about an AS’s internal network. This may be a disadvantage *in the long run.*
- Note that the goal of the mechanism is not acquisition of private inputs *per se* but rather evaluation of a function of those inputs.

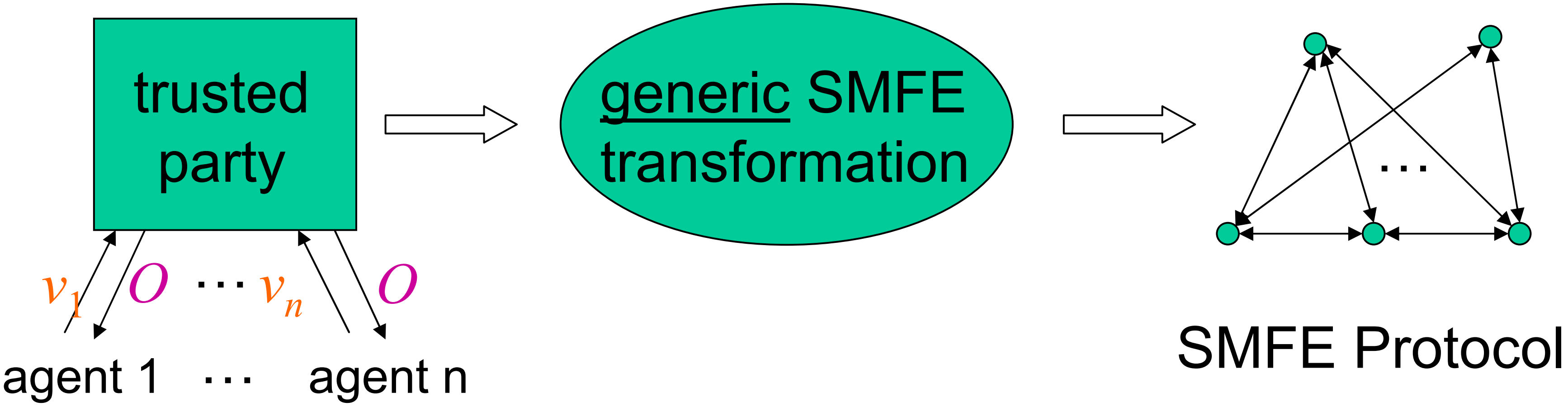
Secure, Multiparty Function Evaluation



$$O = O(v_1, \ldots, v_n)$$

- Each i learns O.
- No i can learn anything about v_j (except what he can infer from v_i and O).
- Extensive SMFE theory; see, *e.g.*, [C00, G03].

Constructive, “Compiler”-Style Results



Natural approach:

centralized mechanism ≈ trusted party
DAM ≈ SMFE protocol

Must be careful about strategic models and solution concepts.

Combining MD and SMFE

Example: Transform a centralized, strategyproof mechanism using the "secure" (against an active adversary) protocol construction in [BGW88] (with $t = 1$). Result is:

- An *input game*, with a dominant-strategy equilibrium in which every agent "shares" his true valuation.
- A *computational game*, with a Nash equilibrium in which every agent follows the protocol.
- Agent privacy!

Need specific properties of [BGW88] construction (*e.g.*, initial input commitment) as well as general definition of security.

Open Questions

- Complete understanding of what follows from known SMFE constructions
- Privacy-preserving DAMs that have good network complexity
- New solution concepts designed for Internet computation
- New kinds of mechanisms and protocols with highly transient sets of agents

Outline

- Motivation and background
- Example: Multicast cost sharing
- Overview of known results
- BGP-based interdomain-routing mechanisms
- Canonically hard DAMD problems
- Distributed implementation challenges
- Other research directions

More Problem Domains

- Caching
- Distributed Task Allocation
- Overlay Networks

★ Ad-hoc and/or Mobile Networks

- …

Ad-Hoc and/or Mobile Networks

- Nodes make same incentive-sensitive decisions as in traditional networks, *e.g.*:
  - Should I connect to the network?
  - Should I transit traffic?
  - Should I obey the protocol?
- These decisions are made more often and under faster-changing conditions than they are in traditional networks.
- Resources (*e.g.*, bandwidth and power) are scarcer than in traditional networks. Hence:
  - Global optimization is more important.
  - Selfish behavior by individual nodes is potentially more rewarding.

Approximation in DAMD

- AMD approximation is subtle. One can easily destroy strategyproofness.
- "Feasibly dominant strategies" [NR00]
- "Strategically faithful" approximation [AFK+04]
- "Tolerable manipulability" [AFK+04]
- "Approximate strategyproofness" [APTT03, GH03, KPS03, S01]

Indirect Mechanisms

Explore tradeoffs among

- agent computation
- mechanism computation
- communication
- privacy
- approximation factors

References (1)

[AFK+04] A. Archer, J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker. "Approximation and Collusion in Multicast Cost Sharing," *Games and Economic Behavior* **47** (2004), pp. 36-71. http://www.cs.yale.edu/~jf/AFKSS.pdf.

[APTT03] A. Archer, C. Papadimitriou, K. Talwar, and E. Tardos. "An Approximate Truthful Mechanism for Combinatorial Auctions with Single Parameter Agents," in *Proceedings of the 14th Symposium on Discrete Algorithms*, ACM Press / SIAM, New York / Philadelphia, 2003, pp. 205-214. http://www.orie.cornell.edu/~aarcher/Research/logcopySODAproc.ps.

[AR02] M. Adler and D. Rubenstein. "Pricing Multicast in More Practical Network Models," in *Proceedings of the 13th Symposium on Discrete Algorithms*, ACM Press / SIAM, New York / Philadelphia, 2002, pp. 981-990. http://www.cs.umass.edu/~micah/pubs/pricing.ps.

[AT02] A. Archer and É. Tardos. "Frugal Path Mechanisms," in *Proceedings of the 13th Symposium on Discrete Algorithms*, ACM Press / SIAM, New York / Philadelphia, 2002, pp. 991-999. http://www.orie.cornell.edu/~aarcher/Research/fpSODAproc.ps.

References (2)

[BGLM04] Y. Bartal, R. Gonen, and P. La Mura. "Negotiation-Range Mechanisms: Exploring the Limits of Truthful Efficient Markets," in *Proceedings of the 5th Conference on Electronic Commerce*, ACM Press, New York, 2004, pp. 1-8.

[BGW88] M. Ben-Or, S. Goldwasser, and A. Wigderson. "Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation," in *Proceedings of the 20th Symposium on Theory of Computing*, ACM Press, New York, 1988, pp. 1-10.

http://www.math.ias.edu/~avi/PUBLICATIONS/MYPAPERS/GBW88/GBW88.pdf.

[BN01] M. Babaioff and N. Nisan. "Concurrent Auctions Across the Supply Chain," in *Proceedings of the 3rd Conference on Electronic Commerce*, ACM Press, New York, 2001, pp. 1-10.

http://www.cs.huji.ac.il/~noam/chain.ps.

[BNP04] M. Babaioff, N. Nisan, and E. Pavlov. "Mechanisms for a Spatially Distributed Market," in *Proceedings of the 5th Conference on Electronic Commerce*, ACM Press, New York, 2004, pp. 9-20.

http://www.cs.huji.ac.il/~mosheb/markets-flow.ps.

References (3)

[C71] E. Clarke. “Multipart Pricing of Public Goods,” *Public Choice* **11** (1971), pp. 17-33.

[C00] R. Canetti. “Security and Composition of Multiparty Cryptographic Protocols,” *Journal of Cryptology* **13** (2000), pp. 143-202. http://www.research.ibm.com/security/sfe-def.ps.

[CFLS97] A. Condon, J. Feigenbaum, C. Lund, and P. Shor. “Random Debaters and the Hardness of Approximating Stochastic Functions,” *SIAM Journal on Computing* **26** (1997), pp. 369-400. http://www.cs.yale.edu/~jf/CFLS.ps.

[CKS81] A. Chandra, D. Kozen, and L. Stockmeyer. “Alternation,” *Journal of the Association for Computing Machinery* **28** (1981), pp, 114-133. http://doi.acm.org/10.1145/322234.322243.

[CV02] Artur Czumaj and Berthold Vöcking. “Tight Bounds for Worst-Case Equilibria,” in *Proceedings of the 13th Symposium on Discrete Algorithms*, ACM Press / SIAM, New York / Philadelphia, 2002, pp. 413-420. http://www.cis.njit.edu/~czumaj/PUBLICATIONS/SODA-2002-Proceedings-Final-Revision.pdf.

References (4)

[DGGZ03] D. Dutta, A. Goel, R. Govindan, and H. Zhang. "The Design of a Distributed Rating Scheme for Peer-to-Peer Systems," in *Proceedings of the Workshop on the Economics of Peer-to-Peer Systems*, 2003. http://cs.usc.edu/~ramesh/papers/freeride.pdf.

[DPS02] X. Deng, C. Papadimitriou, and S. Safra. "On the Complexity of Equilibria," in *Proceedings of the 34th Symposium on Theory of Computing*, ACM Press, New York, 2002, pp. 67-71. http://www.cs.berkeley.edu/~christos/equi.ps.

[FGHK02] A. Fiat, A. Goldberg, J. Hartline, and A. Karlin. "Competitive Generalized Auctions," in *Proceedings of the 34th Symposium on Theory of Computing*, ACM Press, New York, 2002, pp. 72-81. http://www.math.tau.ac.il/~fiat/stocauc.ps.

[FKSS03] J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker. "Hardness Results for Multicast Cost Sharing," *Theoretical Computer Science* **304** (2003), pp. 215-236. http://www.cs.yale.edu/~jf/FKSS2.pdf.

References (5)

[FPS01] J. Feigenbaum, C. Papadimitriou, and S. Shenker. "Sharing the Cost of Multicast Transmissions," *Journal of Computer and System Sciences* **63** (2001), pp. 21-41. http://www.cs.yale.edu/~jf/FPS.pdf.

[FPSS02] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker. "A BGP-based Mechanism for Lowest-Cost Routing," in *Proceedings of the 21st Symposium on Principles of Distributed Computing*, ACM Press, New York, 2002, pp. 173-182. http://www.cs.yale.edu/~jf/FPSS.pdf.

[FS02] J. Feigenbaum and S. Shenker. "Distributed Algorithmic Mechanism Design: Recent Results and Future Directions," in *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, ACM Press, New York, 2002, pp. 1-13. http://www.cs.yale.edu/~jf/FS.pdf.

[FSS04] J.Feigenbaum, R. Sami, and S. Shenker. "Mechanism Design for Policy Routing," to appear in *Proceedings of the 23rd Symposium on Principles of Distributed Computing,* ACM Press, New York, 2004*.*

http://www.cs.yale.edu/~jf/FSS.pdf

[G73] T. Groves. "Incentives in Teams," *Econometrica* **41** (1973), pp. 617-663.

References (6)

[G03] O. Goldreich. “Cryptography and Cryptographic Protocols,” *Distributed Computing* **16** (2003), pp. 149-159. http://www.wisdom.weizmann.ac.il/~oded/PS/foc-sur01.ps.

[GK03] A. C. Gilbert and H. Karloff. “On the Fractal Behavior of TCP,” in *Proceedings of the 35th Symposium on Theory of Computing*, ACM Press, New York, 2003, pp. 297-306. http://doi.acm.org/10.1145/780542.780588.

[GKT03] A. Gupta, A. Kumar, and M. Thorup. “Tree Based MPLS Routing,” in *Proceedings of the 15th Symposium on Parallelism in Algorithms and Architectures*, ACM Press, New York, 2003, pp. 193-199. http://doi.acm.org/10.1145/777412.777443.

[GH03] A. Goldberg and J. Hartline. “Envy-Free Auctions for Digital Goods,” in *Proceedings of the 4th ACM Conference on Electronic Commerce*, ACM Press, New York, 2003, pp. 29-35. http://www.avglab.com/andrew/pub/ec03.ps.

[GJR03] T. Griffin, A. Jaggard, and V. Ramachandran. “Design Principles of Policy Languages for Path Vector Protocols,” in *Proceedings of ACM SIGCOMM ’03,* ACM Press, New York, 2003, pp. 61-72. http://www.cs.yale.edu/~vijayr/papers/0203-griffin.pdf.

References (7)

[GMP01] A. Goel, A. Meyerson, and S. Plotkin. “Distributed Admission Control, Scheduling, and Routing with Stale Information,” in *Proceedings of the 12th Symposium on Discrete Algorithms*, ACM Press / SIAM, New York / Philadelphia, 2001, pp. 611-619. http://pollux.usc.edu/~agoel/papers/multiple.ps.

[GL79] J. Green and J.-J. Laffont. **Incentives in Public Decision Making**, North Holland, Amsterdam, 1979.

[GW99] T. Griffin and G. Wilfong. “An Analysis of BGP Convergence Properties,” in *Proceedings of SIGCOMM ’99*, ACM Press, New York, 1999, pp. 277-288. http://www.acm.org/sigcomm/sigcomm99/papers/session8-1.html.

[GY02] N. Garg and N. E. Young. “On-Line End-to-End Congestion Control,” in *Proceedings of the 43rd Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Los Alamitos, 2002, pp. 303-312. http://arxiv.org/pdf/cs.DS/0205032.

References (8)

[HS01] J. Hershberger and S. Suri. “Vickrey Prices and Shortest Paths: What is an Edge Worth?” in *Proceedings of the 42nd Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Los Alamitos, 2001, pp. 129-140. http://www.cs.yale.edu/~jf/HS01.ps.

[KP99] E. Koutsoupias and C. Papadimitriou. “Worst-Case Equilibria,” in *Proceedings of the 16th Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, vol. 1563, Springer, Berlin, 1999, pp. 404-413. http://www.cs.ucla.edu/~elias/publications/paper-kp99.pdf.

[KPS03] A. Kothari, D. Parkes, and S. Suri. “Approximately Strategyproof and Intractable Multi-Unit Auctions,” in *Proceedings of the 4th Conference on Electronic Commerce*, ACM Press, New York, 2003, pp. 166-175. http://www.cs.ucsb.edu/~suri/psdir/ec03.pdf.

[MS01] H. Moulin and S. Shenker. “Strategyproof Sharing of Submodular Costs: Budget Balance Versus Efficiency,” *Economic Theory* **18** (2001), pp. 511-533. http://www.cs.huji.il/~noam/econcs/2001/ms.ps.

[N85] J. Nagle. “On Packet Switches With Infinite Storage,” RFC 970 (1985). http://www.faqs.org/rfcs/rfc970.html.

References (9)

[NR00] N. Nisan and A. Ronen. “Computationally Feasible VCG Mechanisms,” in *Proceedings of the 2nd Conference on Electronic Commerce*, ACM Press, New York, 2000, pp. 242-252. http://www.cs.huji.ac.il/~noam/vcgbased.ps.

[NR01] N. Nisan and A. Ronen. “Algorithmic Mechanism Design,” *Games and Economic Behavior* **35** (2001), pp. 166-196. http://www.cs.huji.ac.il/~noam/selfishJ.ps.

[NWD03] T.-W. Ngan, D. Wallach, and P. Druschel. “Enforcing Fair Sharing of Peer-to-Peer Resources,” in *Proceedings of Peer-to-Peer Systems II: 2nd International Workshop on Peer-to-Peer Systems, Revised Papers*, LNCS **2735**, Springer, Heidelberg, 2003, pp. 149-159. http://www.cs.rice.edu/~druschel/publications/p2pquota.pdf.

[P85] C. Papadimitriou. “Games Against Nature,” *Journal of Computer and System Sciences* **31** (1985), pp. 288-301.

[RT02] T. Roughgarden and É. Tardos. “How Bad Is Selfish Routing?” *Journal of the Association for Computing Machinery* **49** (2002), pp. 236-259. http://www.cs.cornell.edu/timr/papers/routing.pdf.

[S01] J. Schummer. “Almost Dominant Strategy Implementation,” 2001. http://www.kellogg.nwu.edu/faculty/schummer/ftp/research/esp.ps.

References (10)

[S03] R. Sami. *Distributed Algorithmic Mechanism Design*, Ph.D. Thesis, Yale University, 2003. http://www.cs.yale.edu/~jf/Rahul-thesis.pdf.

[SP04] J. Shneidman and D. Parkes. “Specification Faithfulness in Networks with Rational Nodes,” to appear in *Proceedings of the 23rd Symposium on Principles of Distributed Computing,* ACM Press, New York, 2004. http://www.eecs.harvard.edu/econcs/pubs/pp068-shneidman.pdf.

[V61] W. Vickrey. “Counterspeculation, Auctions, and Competitive Sealed Tenders,” *Journal of Finance* **16** (1961), pp. 8-37.