Fast and Simple Approximation Algorithms for Generalized Flow

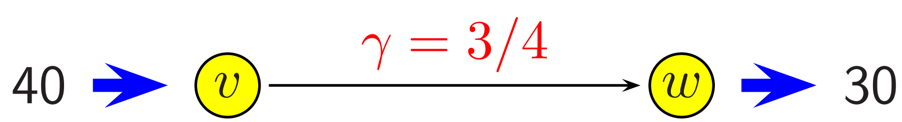
(flow with losses and gains)

Kevin Wayne

Princeton University

www.cs.princeton.edu/~wayne

(with Lisa Fleischer)



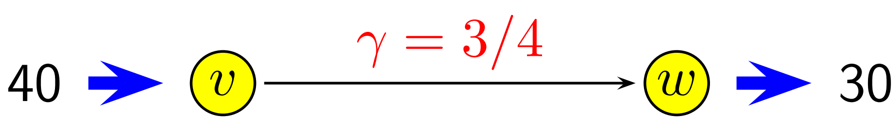
Fast and Simple Approximation Algorithms for Generalized Flow

Lisa Fleischer

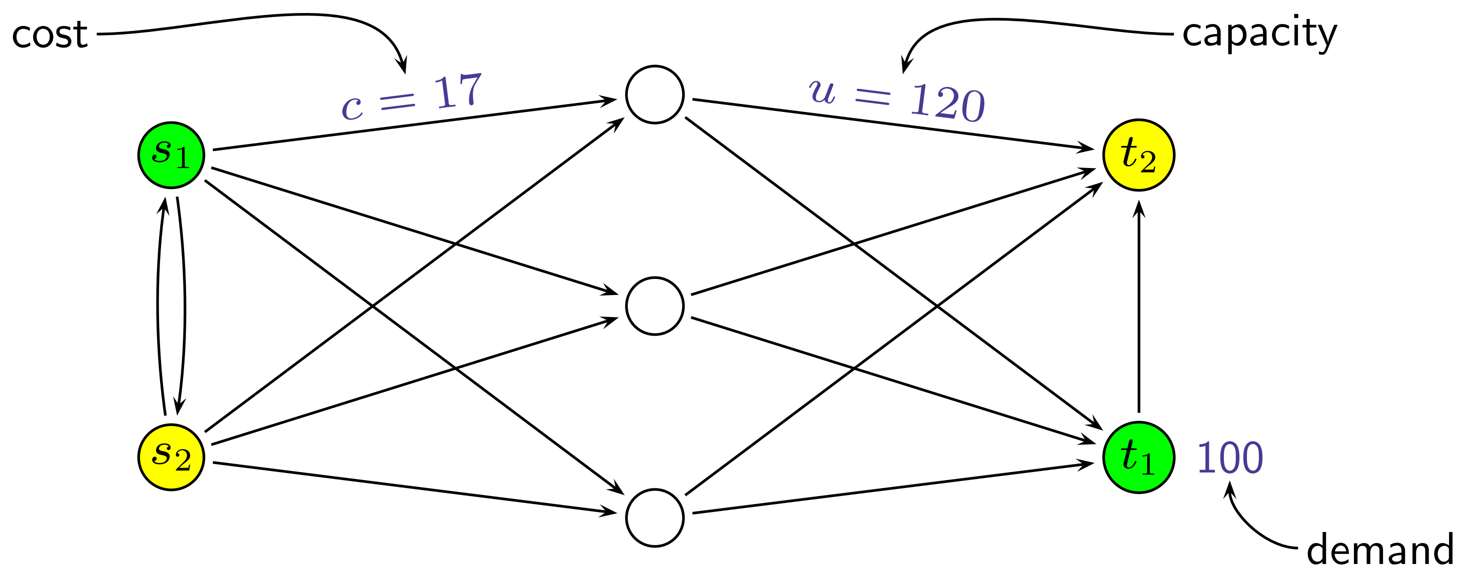
Columbia University

www.ieor.columbia.edu/~lisa

(with Kevin Wayne)



Multicommodity Flow



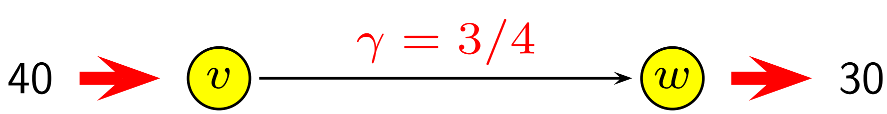
Send flow from sources to sinks as cheaply as possible

- capacity constraints
- flow conservation constraints

Generalized

$\gamma = 3/4$ gain factor

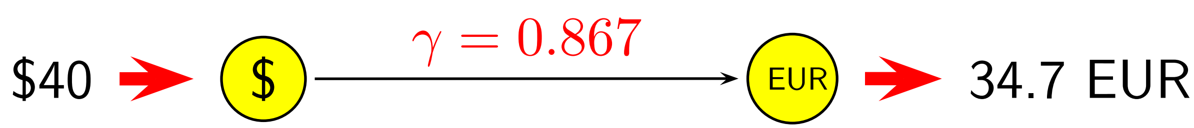
(generalized)



Applications

Commodity leaks:
leaky pipes, defects, production yields, evaporation, spoilage, theft, taxes

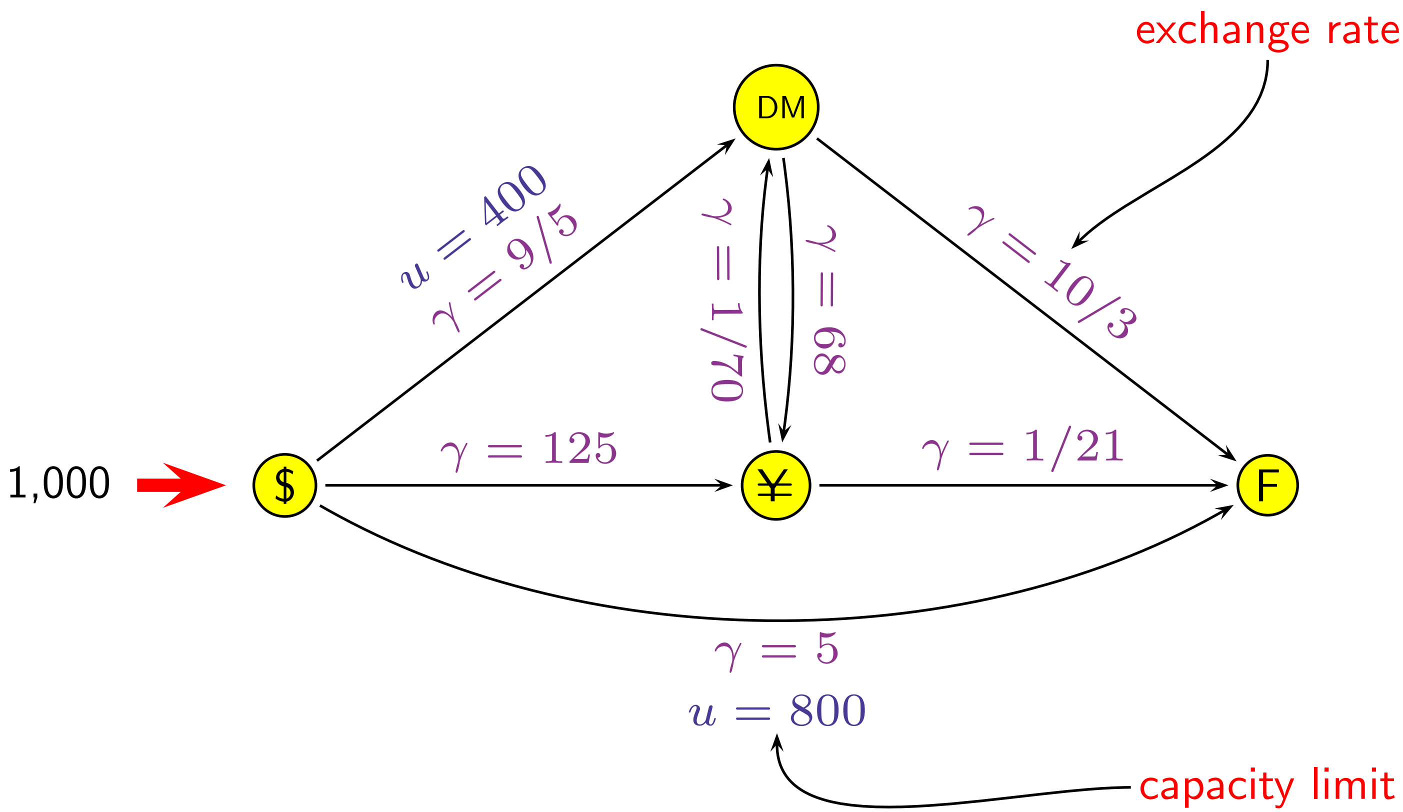
Transform one commodity into another:
currency conversion, machine loading, fuel utilization, crop management, aircraft fleet assignment



Currency Conversion

Convert $1,000 US into maximum number of French Francs

- exploit discrepancies in exchange rates
- can only convert bounded amounts of currency

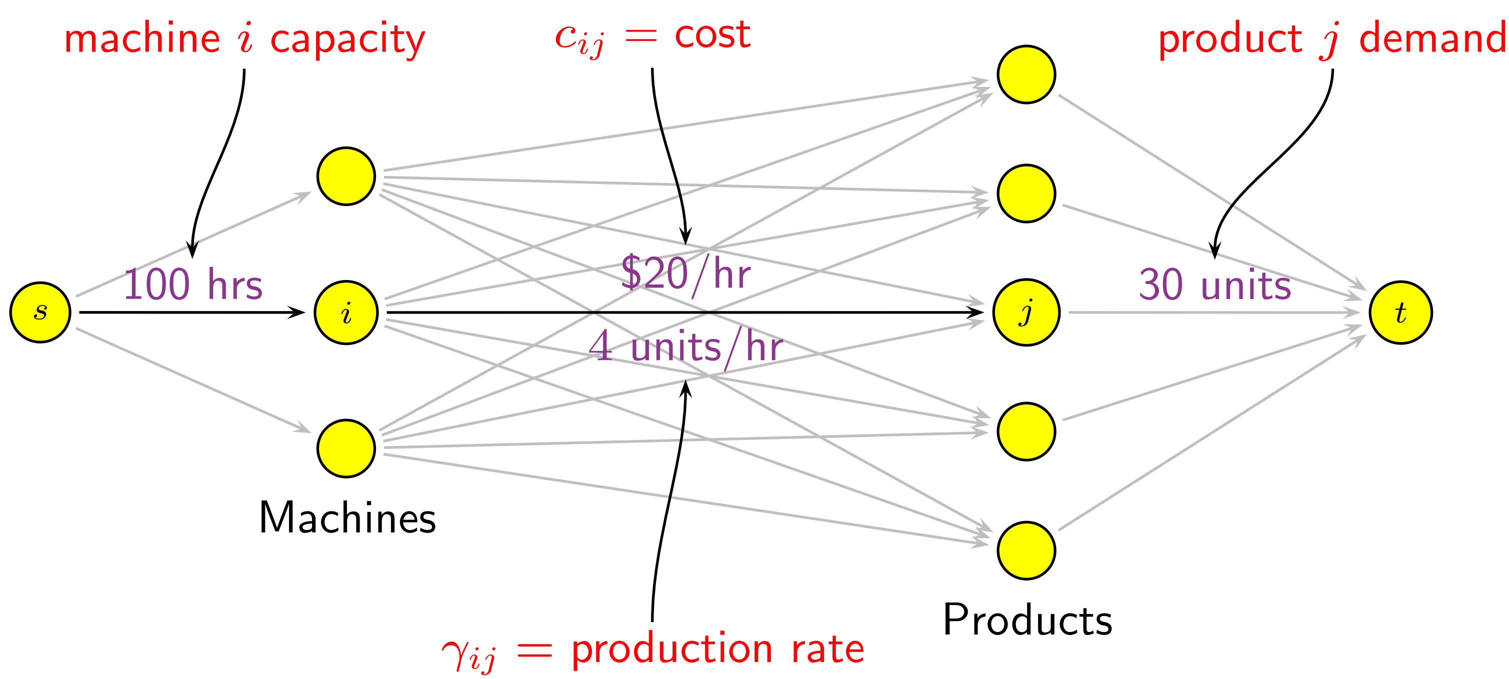


Machine Loading

Schedule p product types on q machines to minimize cost

- machines have limited capacity
- in 1 hour machine i processes γ_{ij} units of product j at cost c_{ij}
- find optimal product mix for each machine

Need to find generalized max flow of min cost:



Exact Algorithms for Generalized Flow

Generalized flow: Kantorovich '39, Dantzig '62
simplex, network simplex

Generalized max flow: Goldberg-Plotkin-Tardos '91
- first polynomial combinatorial algorithm
- other work: Cohen-Megiddo '94, Radzik '93,
  Goldfarb-Jin-Orlin '97, Tardos-Wayne '98

Generalized min cost flow: Wayne '99
first polynomial combinatorial algorithm

Generalized multi. flow: Vaidya '89, Kamath-Palmon '95
specialized interior point methods for multicommodity flow

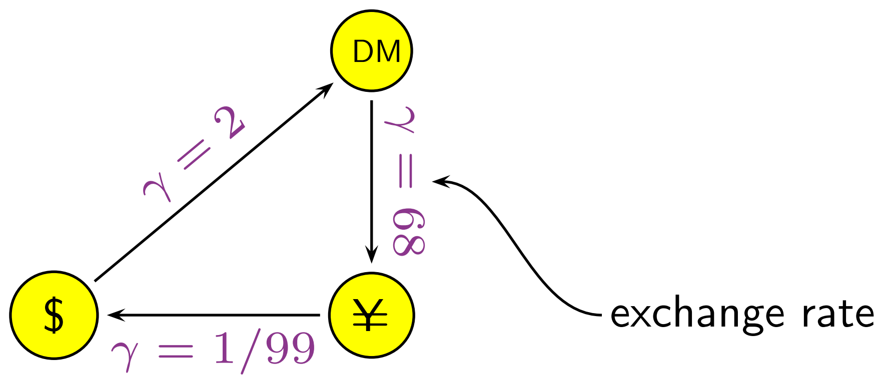
What We Do

- We give simple FPTAS for generalized max flow, multicommodity flow and versions with costs
  - find feasible flow that satisfies $\geq (1 - \epsilon)$ fraction of demand
  - running time polynomial in $1/\epsilon$ and input size
  - much faster than exact algorithms

- Our algorithms based upon Garg-Könemann '98 traditional multicommodity flow FPTAS
  - uses exponential length function: Shahrokhi-Matula '90
  - method refined and extended by: Klein, et al. '94, Leighton, et al. '95, Goldberg '92, Plotkin-Shmoys-Tardos '95, Grigoriadis-Khachiyan '96, Radzik '98, Oldham '99

A Practical Assumption: No Arbitrage

- Our algorithms work in networks with no flow-generating cycles (product of its gain factors > 1)



  - financial networks - arbitrage
  - energy networks - perpetual energy source
  - most practical applications have no flow-generating cycles

- For talk, we assume all gains $\gamma \leq 1$ (equivalent to no arbitrage condition)

Sampling of New Results

Generalized concurrent multicommodity flow: $\mathcal{O}^*(m^2 + km)$

- simplest known algorithm, appears quite practical
- matches complexity of fastest non-generalized methods
- Oldham '99: $\mathcal{O}^*(km^2n^2)$ (even works with arbitrage)

Generalized max flow $\mathcal{O}^*(m^2)$

- interesting and well-studied special case
- previous best strongly poly algorithm: $\mathcal{O}^*(m^2n)$ Radzik '93

Generalized min cost flow $\mathcal{O}^*(m^2 \log\log I)$

- $\mathcal{O}^*(m^2n^3 \log I)$ Oldham '99

k commodities, I = biggest integer, $\epsilon > 0$ constant, $\mathcal{O}^*$ hides polylog(m)

Generalized Max Flow - Path Formulation

$0 \le x(P) =$ amount of flow reaching t on s-t path P

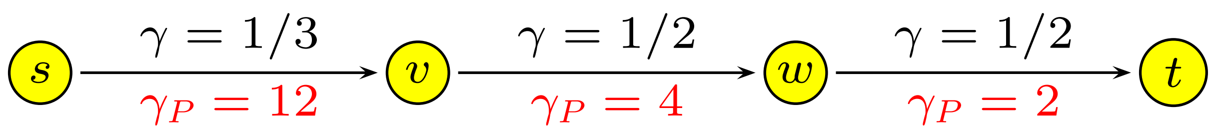
$0 \le l(e) = \underbrace{\text{length of arc } e}_{\text{marginal cost of 1 unit of capacity}}$

$$(P) \quad \max \sum_P x(P)$$
$$\sum_{P:e\in P} \gamma_P(e)\, x(P) \le u(e) \qquad \forall e$$
$$x(P) \ge 0 \qquad \forall P$$

$$(D) \quad \min \sum_e u(e)\, l(e)$$
$$l(e) \ge 0$$
$$\sum_{e\in P} \gamma_P(e) l(e) \ge 1$$

$\gamma_P(e) =$ flow sent on e to get 1 unit of flow at t on P

P :



Garg-Könemann Max Flow Algorithm

Goal: find ϵ-approximate max flow

Maintain primal infeasible flow $f(e)$, and dual infeasible length function $l(e)$ which is exponential in flow sent through arc e

Initialize $x(P) \leftarrow 0, \forall e : l(e) \leftarrow \delta(m, \epsilon)$
repeat
  $P \leftarrow$ shortest s-t path using lengths l
  $u \leftarrow$ bottleneck capacity of P (using original capacities)
  $x(P) \leftarrow x(P) + u$
  $\forall e \in P : l(e) \leftarrow l(e) \times (1 + \epsilon u/u(e))$
until $\forall P : l(P) \geq 1$ (dual feasible)
return x scaled to be primal feasible

Theorem [GK '98] $\mathcal{O}^*(\epsilon^{-2}m^2)$ FPTAS

Generalized Max Flow Algorithm

Goal: find ϵ-approximate generalized max flow

Maintain (infeasible) flow $x(P)$, length function $l(e)$ which is exponential in flow sent through arc e

initialize $x(P) \leftarrow 0$, $l(e) \leftarrow \delta(m, \epsilon)$

repeat

$P \leftarrow$ "cheapest s-t path" $\boxed{l(P) := \sum_{e \in P} \gamma_P(e) l(e)}$

cost of getting 1 unit of flow at t (no capacities)

$u \leftarrow$ most flow that can reach t on P (original capacities)

$x(P) \leftarrow x(P) + u$

$\forall e \in P : l(e) \leftarrow l(e) \times (1 + \epsilon\, u\, \gamma_P(e)/u(e))$

until $\forall P : l(P) \geq 1$ (dual feasible)

return x scaled to be feasible flow

Lemma ϵ-approximate after $\mathcal{O}^*(\epsilon^{-2} m)$ calls to subroutine

A Fast Subroutine for Gen. Shortest Path

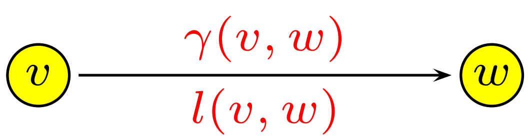
Goal: find cheapest way to deliver 1 unit of flow at t (in uncapacitated network) from s $\boxed{l(P) := \sum_{e \in P} \gamma_P(e) l(e)}$

- use greedy Dijkstra approach
- costs $l(e) \geq 0$, gains $0 < \gamma(e) \leq 1$

Update formula:

- $\pi(v)$ = cheapest way to deliver 1 unit at v from s
- getting 1 unit at v, then along (v, w) costs $\pi(v) + l(v, w)$
- but only $\gamma(v, w)$ units arrive at w

$\pi(v)$



$$\pi(w) \leq \frac{\pi(v) + l(v, w)}{\gamma(v, w)}$$

Lemma: compute π in $\mathcal{O}(m + n \log m)$ time

$\Rightarrow$ **Theorem:** $\mathcal{O}^*(\epsilon^{-2} m^2)$ FPTAS for generalized max flow

Generalized Min Cost Flow

$0 \le x(P) =$ amount of flow reaching t on s-t path P

$$
\begin{aligned}
(P) \quad & \max \sum_P x(P) \\
& \sum_{P: e \in P} \gamma_P(e)\, x(P) \le u(e) \qquad \forall e \\
& \sum_e c(e) \sum_{P: e \in P} \gamma_P(e)\, x(P) \le B
\end{aligned}
$$

$0 \le l(e) =$ length of arc e, $0 \le \phi =$ budget constraint dual variable

$$
\begin{aligned}
(D) \quad & \max \sum_e u(e) l(e) + B\phi \\
& \sum_{e \in P} \gamma_P(e)\, (l(e) + c(e)\phi) \ge 1 \quad \forall P
\end{aligned}
$$

- find cheapest path P using lengths $l(e) + \ c(e)\phi$
- send as much flow on path P without violating original capacity constraints or original budget constraint, update l and ϕ

Theorem: $\mathcal{O}^*(\epsilon^{-2} m^2)$ FPTAS for cost-bounded max flow

Theorem: $\mathcal{O}^*(\epsilon^{-2} m^2 \log\log I)$ FPTAS for min cost max flow

Closing Remarks

Generalized concurrent multicommodity flow: $\mathcal{O}^*(\epsilon^{-2}m(m+k))$

- simple and appears quite practical
- at least factor n^2 improvement
- matches complexity of fastest non-generalized methods
- with costs: $\mathcal{O}^*(\epsilon^{-2}km^2 \log\log I)$

Generalized max flow: $\mathcal{O}^*(\epsilon^{-2}m^2)$

- improves complexity even for well-studied problem
- using Tardos-Wayne gain-scaling technique, we reduce dependence on error parameter from $1/\epsilon^2$ to $\log(1/\epsilon)$

Open Problem: reduce dependency on error from $1/\epsilon^2$ to $\log(1/\epsilon)$ for version with costs or multiple commodities