A faster and simpler strongly polynomial algorithm for generalized flow maximization

László Végh

London School of Economics

Joint work with

Neil Olver

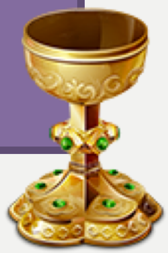
Vrije Universiteit Amsterdam and CWI

Queen Mary Algorithms Day

17 July 2018

Linear Programming
& strongly polynomial algorithms

Is there a strongly polynomial algorithm for Linear Programming?



$$\min c^\top x$$
$$Ax = b$$
$$x \geq 0$$

$$A \in \mathbb{Q}^{m \times n}$$

- Strongly polynomial
  - poly(n, m) elementary arithmetic operations and comparisons
  - PSPACE: The numbers do not get too large during the computations
- Weakly polynomial
  - Running time depends on total encoding length L
  - Ellipsoid algorithm, interior point methods

Linear Programming
& strongly polynomial algorithms

- Network flow problems:
  - Maximum flow: Edmonds-Karp-Dinitz '70-72
  - Min-cost flow: Tardos '85

- Combinatorial LP's: integer matrix A.
  - $\mathrm{poly}(n, m, \log \Delta)$ for $A \in \mathbb{Z}^{m \times n}$ with max. subdeterminant Δ: Tardos '86, Vavasis-Ye '96

LP's with at most two nonzero entries per column in the constraint matrix

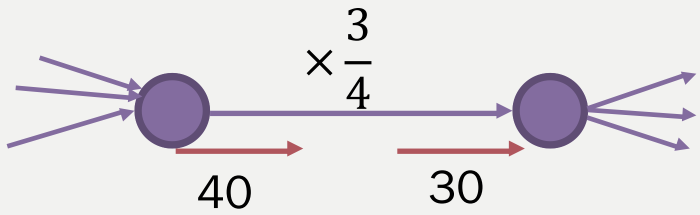
$$\min c^\top x$$
$$Ax = b$$
$$x \geq 0$$

$$A = \begin{bmatrix} * & * & 0 & 0 \\ * & 0 & * & 0 \\ 0 & * & 0 & * \\ 0 & 0 & * & 0 \end{bmatrix}$$

- Dual feasibility
  solving a system of inequalities with ≤ 2 variables per inequality: Megiddo '83
- Primal feasibility
  reduces to *max. generalized flow:* V '14, this paper
- Optimization problem
  equivalent to *min-cost generalized flow*: open
- Three nonzeros
  equivalent to general LP

Generalized flow maximization

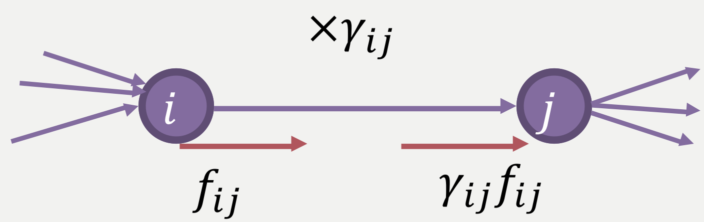
- Network flow model with gain factors



- Gain factors can model:
  - *physical changes (e.g. leakage, theft)*
  - *conversions between different types of entities (e.g. currencies, products)*
- Objective: maximize the flow amount reaching the sink t.

Generalized flow maximization

- Network flow model with gain factors



$$\begin{array}{cc} & ij \\ \begin{array}{c} i \\ \\ j \\ \\ \end{array} & \left[\begin{array}{c} -1 \\ 0 \\ \gamma_{ij} \\ 0 \end{array}\right] \end{array}$$

- Flow balance at node i

$$\nabla f_i = \sum_{j \in \delta^{in}(i)} \gamma_{ji} f_{ji} - \sum_{j \in \delta^{out}(i)} f_{ij}$$

$$\max \nabla f_t$$

s.t. $\nabla f_i \geq b_i \quad \forall i \in V - t$

$$f \geq 0$$

Generalized flow maximization

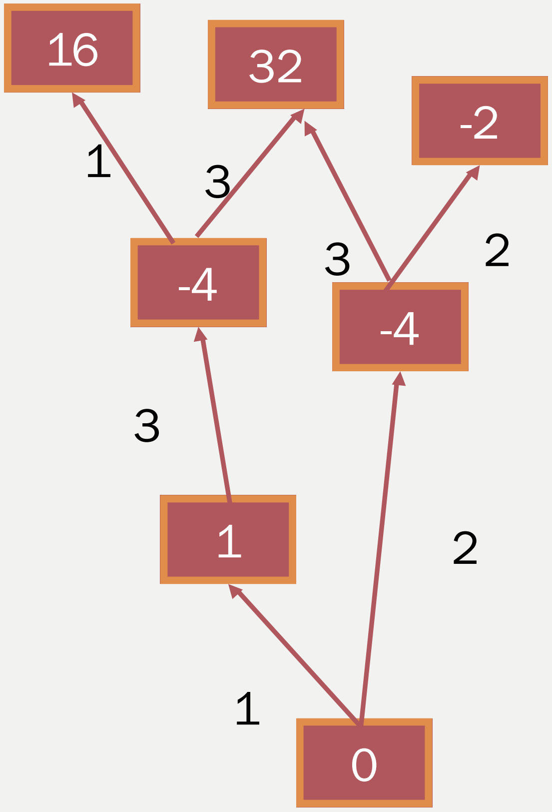
- n: #nodes, m: #arcs, B: largest integer in input data
- Kantorovich ’39, Dantzig ’62: model introduced
- Goldberg, Plotkin & Tardos ’91: first polynomial combinatorial algorithm
- Several weakly polynomial algorithms: Cohen & Megiddo ’94, Goldfarb, Jin ’96, Goldfarb, Jin & Orlin ’97, Tardos & Wayne ’98, Wayne ’02, Daitch & Spielman ’08, Restrepo & Williamson ’09, V. ’12, ...
- Best previous running times:
  - Vaidya ’89: $O(m^{1.5}n^2 \log B)$
  - Radzik ’04: $O\big((m + n\log n)mn\log B\big)$
- V ’14: $O(m^3n^2)$
- Olver-V ’17: $O\left((m + n\log n)mn\log\left(\frac{n^2}{m}\right)\right)$

Detour:
Net Present Value Problem

- Critical Path Problem with time discounts
- Introduced by Russell ’70
- Application of generalized flows
- Strongly polynomial computability becomes a crucial issue

Net Present Value Problem

- Jobs J, precedence constraints $\prec$
- Processing time $p_j \in \mathbb{N}$, profit/loss $b_j \in \mathbb{Z}$ for each $j \in J$
- Discount factor $\rho > 1$



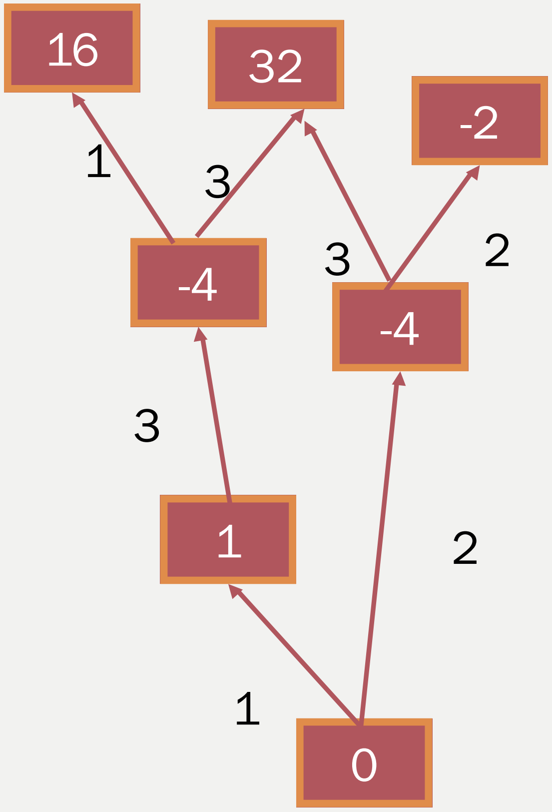
$$\max \sum_{j \in J} b_j \rho^{-c_j}$$
$$\text{s.t. } c_j \geq c_i + p_j \quad \forall i \prec j$$
$$c_t = 0, c \geq 0$$

Net Present Value Problem

$$\max \sum_{j \in J} b_j \rho^{-c_j}$$
$$\text{s.t. } c_j \geq c_i + p_j \quad \forall i \prec j$$
$$c_t = 0, c \geq 0$$

- $\mu_j := \rho^{c_j}, \gamma_{ij} := \rho^{p_j}$

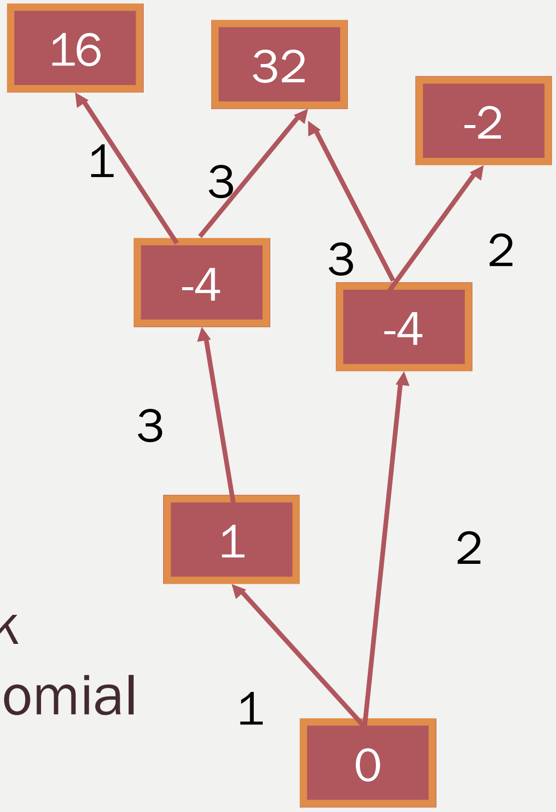
$$\max \sum_{j \in J} \frac{b_j}{\mu_j}$$
$$\text{s.t. } \mu_j \geq \gamma_{ij} \mu_i \quad \forall i \prec j$$
$$\mu_t = 1, \mu > 0$$



Net Present Value Problem

- $\mu_j := \rho^{c_j}, \gamma_{ij} := \rho^{p_j}$

$$\max \sum_{j \in J} \frac{b_j}{\mu_j}$$
$$\text{s.t.} \quad \mu_j \geq \gamma_{ij}\mu_i \quad \forall i \prec j$$
$$\mu_t = 1, \mu > 0$$



- Dual of generalized flow maximization
- Weakly polynomial algorithms cannot work
- A variant of our algorithm is strongly polynomial for this problem (at least for $\rho \in \mathbb{N}$): Correa, Olver, Schulz, V.

Generalized flow maximization

- Flow balance at node i

$$\nabla f_i = \sum_{j \in \delta^{in}(i)} \gamma_{ji} f_{ji} - \sum_{j \in \delta^{out}(i)} f_{ij}$$

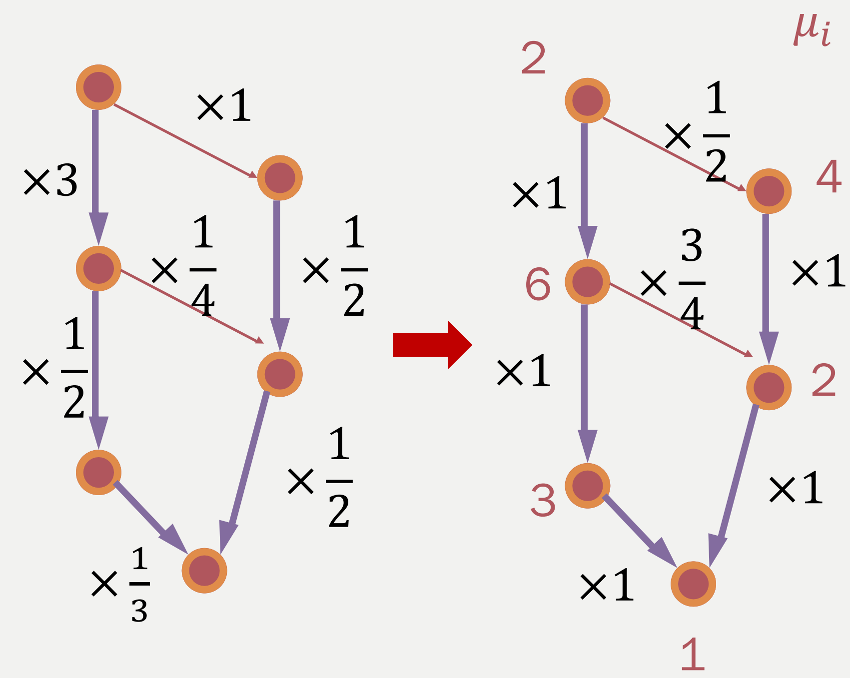
$$\begin{aligned} & \max \nabla f_t \\ \text{s.t.} \quad & \nabla f_i \geq b_i \quad \forall i \in V - t \\ & f \geq 0 \end{aligned}$$

Dual

$$\begin{aligned} & \max \mu_t \sum_{j \in J} \frac{b_j}{\mu_j} \\ \text{s.t.} \quad & \mu_j \geq \gamma_{ij} \mu_i \quad \forall ij \in E \\ & \mu > 0 \end{aligned}$$

Labellings and duality

- Change measurement unit
- $\mu : V \to \mathbb{R}_{>0}$
- $f_{ij}^{\mu} := \frac{f_{ij}}{\mu_i}$, $\quad \gamma_{ij}^{\mu} := \frac{\gamma_{ij}\mu_i}{\mu_j}$,
  $b_i^{\mu} := \frac{b_i}{\mu_i}$, $\quad \nabla f_i^{\mu} := \frac{\nabla f_i}{\mu_i}$
- $ij \in E$ is tight if $\gamma_{ij}^{\mu} = 1$
- (f, μ) is a fitting pair if
  - $\gamma_{ij}^{\mu} \leq 1 \quad \forall ij \in E$
    dual feasibility
  - $\gamma_{ij}^{\mu} = 1$, if $f_{ij} > 0$
    complementary slackness



Dual

$$\max \mu_t \sum_{j \in J} b_j^{\mu}$$

$$\text{s.t.} \quad \gamma_{ij}^{\mu} \leq 1 \quad \forall ij \in E$$

$$\mu > 0$$

Main progress: abundant arcs

- Main goal: find a dual optimal solution μ^*.
- Primal optimal solution: max normal flow computation on tight arcs w.r.t. μ^*.
- Main progress:
  - *find an abundant arc a that must be tight for every optimal dual solution*
  - *contract a and recurse.*
- Standard technique for minimum cost flows: Orlin '93
  - *also used in V. '14*

Main progress: abundant arcs

- Feasible flow: $\nabla x_i \geq b_i \quad \forall i \in V - t$
- Total relabeled excess

$$Ex(x,\mu) := \sum_{i \in V-t} (\nabla x_i^\mu - b_i^\mu)$$

LEMMA: If x is a feasible flow and (x,μ) is a fitting pair, and

$$x_a^\mu > Ex(x,\mu),$$

Then a is an abundant arc.

- Proof: standard flow decomposition technique.

Main progress: contractible arcs

Assume the following hold for flow f, and labelling μ:

- (f, μ) is a fitting pair (complementary slackness)
- $\nabla f_i^\mu \geq b_i^\mu \quad \forall i \in V - t$ (feasibility)
- $\nabla f_i^\mu \leq b_i^\mu + 2$ (small excess)

Then

- every arc a with $f_a^\mu > 2n \geq Ex(x, \mu)$ is contractible;
- if $\left|b_i^\mu\right| > 2n^2$, then there exists a contractible arc incident to node i.

Main progress: contractible arcs

Assume the following hold for flow f, and labelling μ:

- (f, μ) is a fitting pair (complementary slackness)
- $\nabla f_i^\mu \geq b_i^\mu \quad \forall i \in V - t \quad$ (feasibility)
- $\nabla f_i^\mu \leq b_i^\mu + 2$ (small excess)

- Problem: difficult to have both fitting pairs and feasibility in an augmenting path framework
- Many previous algorithms relax the fitting pair notion
- New idea: relax feasibility

Algorithmic setup

Maintain a flow f, and labelling μ:

- (f, μ) is a fitting pair (complementary slackness)
- ~~$\nabla f_i^\mu \geq b_i^\mu \ \forall i \in V - t$ (feasibility)~~
- $\nabla f_i^\mu \leq b_i^\mu + 2$ (small excess)
- μ is a safe labelling: there exists a primal feasible solution on tight arcs for μ.
  Can be characterized by a simple cut condition

LEMMA: If (f, μ) is a fitting pair, and μ is <u>safe,</u> then there exists a flow x such that (x, μ) is a fitting pair and

$$b_i \leq \nabla x_i \leq \max(b_i, \nabla f_i) \ \forall i \in V - t$$

- f^μ is integral and small ($f_a^\mu \leq 2n$ for all a) 🙂

Work with two flows instead of one!

LEMMA: If (f, μ) is a fitting pair, and μ is **safe**, then there exists a flow x such that (x, μ) is a fitting pair and

$$b_i \leq \nabla x_i \quad \leq \max(b_i, \nabla f_i) \;\; \forall i \in V - t$$

- If $\left|b_i^\mu\right| > 2n^2$, then compute such an x by a flow feasibility algorithm.

This satisfies:

- (x, μ) is a fitting pair (complementary slackness)
- $\nabla x_i^\mu \geq b_i^\mu \quad \forall i \in V - t$ (feasibility)
- $\mathrm{Ex}(\mathrm{x}, \mu) \leq 2n$ (small excess)
- There is an arc a with $x_a^\mu > 2n$ and this is contractible.

Main step

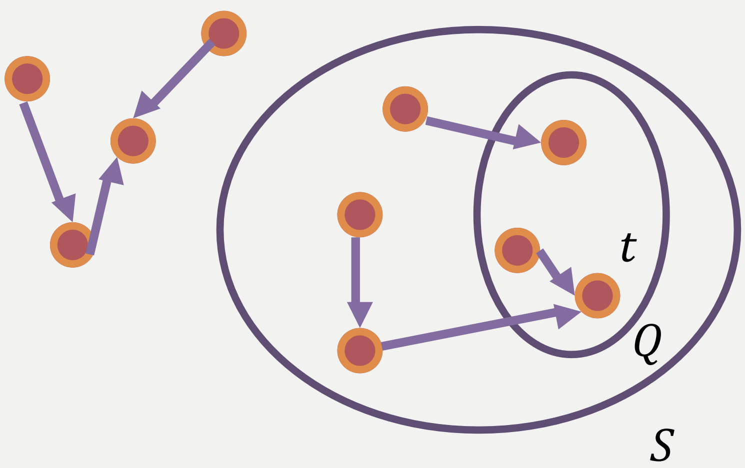
multiplicative Dijkstra-type update

- $Q \coloneqq \{t\} \cup \{j \in V : \nabla f_j^\mu < b_j^\mu\}$ – "sink set"
- Try send 1 unit of relabelled flow from a node j with $\nabla f_j^\mu \geq b_j^\mu + 1$ to a node in Q on a tight path.
- S: set of nodes that can reach Q on a tight path

Update for certain $\alpha > 1$

$$\mu_i := \begin{cases} \frac{\mu_i}{\alpha}, & if\ i \in S \\ \mu_i, & if\ i \in V \setminus S \end{cases}$$

$$f_a := \begin{cases} \frac{f_a}{\alpha}, & if\ a \in E[S] \\ f_a, & otherwise \end{cases}$$

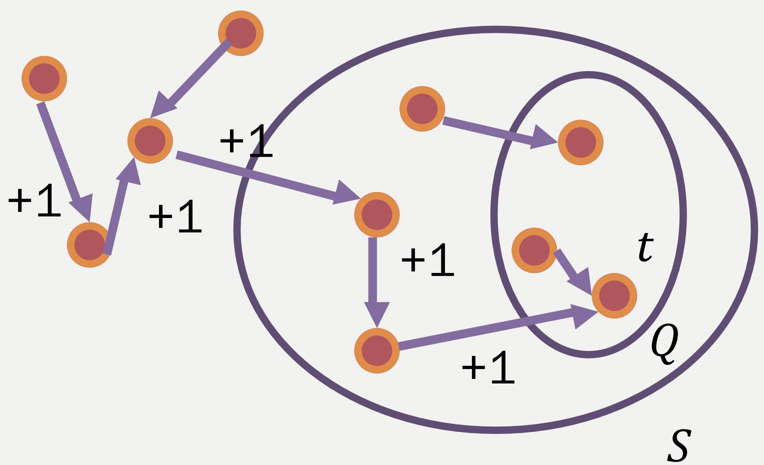


Main step
multiplicative Dijkstra-type update

Update for certain $\alpha > 1$

$$\mu_i := \begin{cases} \dfrac{\mu_i}{\alpha}, & if\ i \in S \\ \mu_i, & if\ i \in V \setminus S \end{cases}$$

$$f_a := \begin{cases} \dfrac{f_a}{\alpha}, & if\ a \in E[S] \\ f_a, & otherwise \end{cases}$$



- The relabeled flow f^μ does not change.
- Arcs entering S may become tight.
- Safety of labelling μ is maintained.

Analysis key idea

Update for certain $\alpha > 1$

$$\mu_i := \begin{cases} \frac{\mu_i}{\alpha}, & if\ i \in S \\ \mu_i, & if\ i \in V \setminus S \end{cases}$$

$$f_a := \begin{cases} \frac{f_a}{\alpha}, & if\ a \in E[S] \\ f_a, & otherwise \end{cases}$$

- How does the excess of a node change?

$$\nabla f_i^\mu - b_i^\mu$$

- Can increase only if $i \in S$ and $b_i < 0$.

$$\Phi := \sum_{i:b_i<0} \nabla f_i^\mu - b_i^\mu \text{ and } \Psi := \sum_{i:b_i<0} |b_i^\mu|$$

- The two potentials increase together at label updates.
- Path augmentations almost always decrease Φ.

Comparison with V '14

V. '14	Current paper
$O(m^3n^2)$	$O\left((m+n\log n)mn\log\left(\frac{n^2}{m}\right)\right)$
Main progress by contracting arcs. Flow sent on tight paths.	
Δ-relaxation of (f,μ) fitting Feasible flow f Non-integral f^{μ}	(f,μ)-fitting pair Infeasible flow f, safe labelling μ Integral f^{μ}
Multiplicative potential argument	Additive potential argument
Additional cleanup step	-
Complicated numerical rounding	Simple numerical rounding

Conclusion

- Amortizing here and there, we get

$$O\left((m + n\log n)mn\log\left(\frac{n^2}{m}\right)\right)$$

- Also works for the Net Present Value problem for integer discount factor.
- Open questions:
  - *Net Present Value problem with rational discount factors.*
  - *Strongly polynomial algorithm for minimum cost generalized flows.*
  - *Strongly polynomial algorithm for LP.*

Thank you!