Goldberg Tarjan push-relabel:

- one of many flow algorithms, v. good complexity properties
- idea:
  - use infeasible flows ("preflows")
  - keep approximate track of path length to t.
  - make flows "more feasible"

Defn's:

- work on $(V, V \times V)$
  all directed edges.
- any edge not in G gets $c=0$
-

- a <u>flow</u> is

$$f : V \times V \longrightarrow R \quad \text{st}$$

$$f(v \to w) \leq c(v \to w)$$

$$f(v \to w) = -f(w \to v)$$

$$\sum_u f(u \to v) = 0$$

for all $v \in V - \{s, t\}$

Kirchoffs laws!

- The <u>value</u> of a flow is

$$\sum_v f(v \to t)$$

- a <u>preflow</u> is $f: V \times V \to R$

  such that

$$f(v \to w) \leq c(v \to w)$$

$$f(v \to w) = -f(w \to v)$$

$$\sum_{u} f(u \to v) \geq 0$$

- <u>excess</u> $e(v) = \sum_{u} f(u \to v)$ (measures failure of Kirchoff's laws)

- we must get rid of excess
  - push toward target
  - but <u>where</u> is toward target?

- residual capacity for some preflow f is:

$$r_f(v \to w) = c(v \to w) - f(v \to w)$$

  if $r_f(v \to w) > 0$, can move flow through this edge

Label function:

$$d : V \to N_0 \cup \{\infty\}$$

non-negative natural numbers

is a valid labelling for f if

$$d(s) = |V|, \quad d(t) = 0$$

$$d(v) \leq d(w) + 1 \quad \text{for } v \to w \text{ st } r_f(v \to w) > 0$$

(approx dist to t along paths where we can push).

Algorithm :

- Initialize : w/ preflow, and valid labelling

$f(s \to v) = c(s \to v)$ for all v

$f(u \to v) = 0$ for all $u \neq s$, v.

$d(s) = |V|$, $d(t) = 0$, $d(v) = 0$, $v \in V - \{s, t\}$

- While there is an active vertex

[active is: $v \neq s, t$, $e(v) > 0$, $d(v) < \infty$]

  - choose an active vertex and execute an admissible operation for that vertex

  [i.e. try to make preflow more feasible]

Operations:

Push: (on an edge $\cancel{e}$, $v \to w$)

admissible if: v is active

and: $r_f(v \to w) > 0$

and: $d(v) = d(w) + 1$

$$\delta = \min\{e(v), r_f(v \to w)\}$$

$$f(v \to w) = f(v \to w) + \delta$$
$$f(w \to v) = f(w \to v) - \delta$$
$$r_f(v \to w) = r_f(v \to w) - \delta$$
$$r_f(w \to v) = r_f(w \to v) + \delta$$

$$e(v) = e(v) - \delta$$
$$e(w) = e(w) + \delta$$

[get rid of excess]

Relabel:

· admissible if:

v is active

and: $r_f(v \to w) > 0$

(always implies $d(v) \leq d(w)$)

$d(v) = \min \{ d(w) + 1 \}$ over edges $r_f(v \to w) > 0$

Facts.:

- If f is a preflow, d a valid labelling, v an active vertex,

  then either a push OR a relabel is admissible for v

- If f starts a preflow, v a valid labelling, they will remain so (INDUCTION)

• Let f be a preflow, and d an arbitrary valid labelling on V

Write G_f for (V, E_f)

$$E_f = \{ v \to w \in V \times V : r_f(v \to w) > 0 \}$$

(edges with residual capacity under f)

Then t is not accessible from s in G_f (i.e. the object we are manipulating is a cut)

<u>Proof:</u> assume there is a path

$$s = v_0 \to v_1 \to v_2 \to \cdots \to v_m = t$$

<u>Now</u> $d(v_i) \leq d(v_{i+1}) + 1$ (cause labelling is valid)

so $d(v_0) = d(s) \leq d(t) + m < |V|$

(because $m \leq |V| - 1$ and $d(t) = 0$)

but $d(s) = |V|$ for valid labelling,

CONTRADICTION

IF the algorithm terminates and all labels are finite, then at termination, f is a flow and is maximal

Proof: · terminates because no admissible vertex ⇒ flow

· maximal, because no augmenting path.

(Alg terminates after at most $O(|V|^2|E|)$ operations)

Proof. Consider the proof of Theorem 10.6. We do at most ϕ iterations, while each iteration takes $O(n+t)$ time, where t is the number of arcs deleted. ∎

Hence, similarly to Corollary 10.6a one has:

Corollary 10.11a. *For integer capacities, a maximum flow can be found in time* $O(n(\phi+m))$, *where* ϕ *is the maximum flow value.*

Proof. Similar to the proof of Corollary 10.6a. ∎

Therefore,

Corollary 10.11b. *For integer capacities, a maximum flow can be found in time* $O(nm \log C)$.

Proof. In the proof of Theorem 10.10, a maximum flow with respect to c' can be obtained from $2f''$ in time $O(nm)$ (by Corollary 10.11a), since the maximum flow value in the residual graph $D_{f''}$ is at most m. ∎

10.8b. Complexity survey for the maximum flow problem

Complexity survey (* indicates an asymptotically best bound in the table):

$O(n^2mC)$	Dantzig [1951a] simplex method
$O(nmC)$	Ford and Fulkerson [1955,1957b] augmenting path
$O(nm^2)$	Dinits [1970], Edmonds and Karp [1972] shortest augmenting path
$O(n^2m \log nC)$	Edmonds and Karp [1972] fattest augmenting path
$O(n^2m)$	Dinits [1970] shortest augmenting path, layered network
$O(m^2 \log C)$	Edmonds and Karp [1970,1972] capacity-scaling
$O(nm \log C)$	Dinits [1973a], Gabow [1983b,1985b] capacity-scaling
$O(n^3)$	Karzanov [1974] (preflow push); cf. Malhotra, Kumar, and Maheshwari [1978], Tarjan [1984]
$O(n^2\sqrt{m})$	Cherkasskiĭ [1977a] blocking preflow with long pushes
$O(nm \log^2 n)$	Shiloach [1978], Galil and Naamad [1979,1980]
$O(n^{5/3}m^{2/3})$	Galil [1978,1980a]

continued

	$O(nm \log n)$	Sleator [1980], Sleator and Tarjan [1981,1983a] dynamic trees
*	$O(nm \log(n^2/m))$	Goldberg and Tarjan [1986,1988a] push-relabel+dynamic trees
	$O(nm + n^2 \log C)$	Ahuja and Orlin [1989] push-relabel + excess scaling
	$O(nm + n^2\sqrt{\log C})$	Ahuja, Orlin, and Tarjan [1989] Ahuja-Orlin improved
*	$O(nm \log((n/m)\sqrt{\log C}+2))$	Ahuja, Orlin, and Tarjan [1989] Ahuja-Orlin improved + dynamic trees
*	$O(n^3/\log n)$	Cheriyan, Hagerup, and Mehlhorn [1990,1996]
	$O(n(m + n^{5/3} \log n))$	Alon [1990] (derandomization of Cheriyan and Hagerup [1989,1995])
	$O(nm + n^{2+\varepsilon})$	(for each $\varepsilon > 0$) King, Rao, and Tarjan [1992]
*	$O(nm \log_{m/n} n + n^2 \log^{2+\varepsilon} n)$	(for each $\varepsilon > 0$) Phillips and Westbrook [1993,1998]
*	$O(nm \log_{\frac{m}{n \log n}} n)$	King, Rao, and Tarjan [1994]
*	$O(m^{3/2} \log(n^2/m) \log C)$	Goldberg and Rao [1997a,1998]
*	$O(n^{2/3}m \log(n^2/m) \log C)$	Goldberg and Rao [1997a,1998]

Here $C := \|c\|_\infty$ for integer capacity function c. For a complexity survey for unit capacities, see Section 9.6a.

Research problem: Is there an $O(nm)$-time maximum flow algorithm?

For the special case of *planar* undirected graphs:

	$O(n^2 \log n)$	Itai and Shiloach [1979]
	$O(n \log^2 n)$	Reif [1983] (minimum cut), Hassin and Johnson [1985] (maximum flow)
	$O(n \log n \log^* n)$	Frederickson [1983b]
*	$O(n \log n)$	Frederickson [1987b]

For *directed* planar graphs:

	$O(n^{3/2} \log n)$	Johnson and Venkatesan [1982]
	$O(n^{4/3} \log^2 n \log C)$	Klein, Rao, Rauch, and Subramanian [1994], Henzinger, Klein, Rao, and Subramanian [1997]
*	$O(n \log n)$	Weihe [1994b,1997b]

Now, notice:

- in augmenting path, we always had a flow
    - is this a cut? not necessarily
    - eg.

$$s \xrightarrow{1\,(3)} \bullet \xrightarrow{1\,(3)} \bullet \xrightarrow{1\,(3)} d$$

- flow that isn't a cut, BECAUSE there is an augmenting path.
- c.f. discussion of aug paths.
    - a flow is ONLY a cut if there is no augmenting path.
      i.e. if its maximal.

In push-relabel, we have manipulating CUTS (by thm, above), and ONLY had a flow at optimality.

~~These two are Dual~~

Max-flow, min cut are DUAL

MAX FLOW	MIN CUT
min $-w^T f$	max $-{\lambda_i^2}^T c$
s.t. $A_R f = 0$	s.t. $-w + A_R^T \lambda_e - \lambda_i^1 + \lambda_i^2 = 0$
$f \geq 0$	$\lambda_i^1 \geq 0$
$c - f \geq 0$	$\lambda_i^2 \geq 0$

These problems are coupled by complementarity conditions

$$\left(\lambda_i^1\right)_K f_K = 0 \qquad K = 1 \cdots \# E$$

$$\left(\lambda_i^2\right)_K \cancel{f_4} \left(c_K - f_K\right) = 0 \qquad K = 1 \cdots \# E$$

(i.e. either the inequality is <u>active</u> or its Lagrange multiplier is 0)

<u>Flow's</u>
Now $\left(\lambda_i^1\right)_K = 0$ implies that flow is > 0

$\left(\lambda_i^2\right)_K = 0$ " " $< c$

- The non-zero λ_i's identify edges that could disconnect (i.e. any path ~~not~~ containing one is not augmenting)
- If the non-zero λ_is meet the <u>equality</u>, we have a <u>CUT</u>

There is a general point here

- consider $\left(f, \underbrace{\lambda_i^1, \lambda_i^2, \lambda_e}_{\text{dual vars.}}\right) = (p, d)$

f: primal vars

1) ~~at a solution~~ if p is soln to Primal, d is soln to dual

p, d are complementary

(follows from KKT).

2) if (p, d) are complementary and
p is primal feasible and
d is dual feasible ~~then~~

$\Updownarrow$

p solves primal $\quad$ d solves dual.

(This follows from KKT conditions).

<u>Hence</u> two kinds of <u>strategy</u>

primal - feasible , dual infeasible (Aug path)

primal - infeasible , dual - feasible (Push relabel