Ahuja Section 10.2 Capacity Scaling

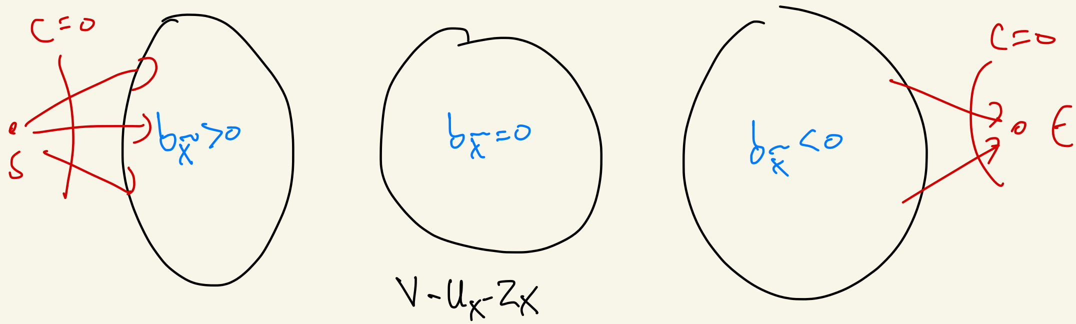
Note that we modify the algorithm to avoid assumption 9.4 in Ahuja!

- Refinement (improved version) of the shortest (min cost) augmenting path method.
- Reduces the number of iterations from $O(nB)$ to $O(m \log U)$ when

  $B = \max\{|b(v)| \mid v \in V\}$ and

  $U = \max\{u_{ij} \mid ij \in A\}$
- Recall that we assume that all arc costs c_{ij} are non negative in $N = (V, A, \ell \equiv 0, u, b, c)$
- Denote by $S(n, m, C)$ the time to solve a shortest path problem in a digraph on n vertices, m arcs and $C = \max\{c_{ij} \mid ij \in A\}$

$U_x = \{v \mid b_x(v) < b(v)\}$ $\qquad$ $Z_x = \{v \mid b_x(v) > b(v)\}$



$$N = (V, A, \ell \equiv 0, u, b, c)$$

suppose x' is feasible in N and x is not (yet) feasible

then $x' = x \oplus \tilde{x}$ when $\tilde{x} \in N(x)$

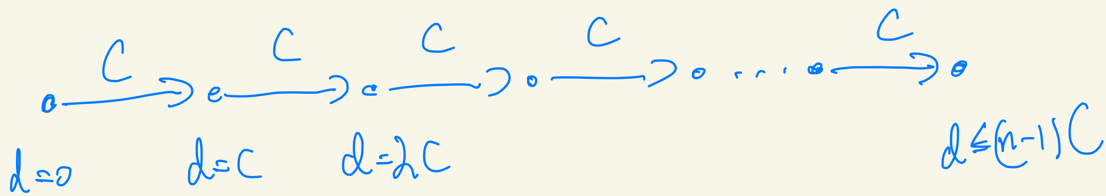
and $b_{x'} = b_x + b_{\tilde{x}}$

$B = \max\{|b(v)| \mid v \in V\}$

$C = \max\{|c_{ij}| \mid ij \in A\}$

$$O(n B\, S(n, m, nC))$$

- Recall that when we work with reduced costs, as in Ahuja, then they may be as large as nC so the time to solve one shortest path calculation in the residual network is $O(S(n,m,nC))$.



$$\pi \leftarrow \pi - d$$

$$c_{ij}^{\pi} = c_{ij} - \pi_i + \pi_j$$

Scaling idea (similar to cap. scaling for max flow):

$U = \max\{u_{ij} \mid ij \in A\}$

Let $\Delta_0 = 2^k$ $\quad k = \lfloor \log_2 U \rfloor$

Consider phases $q = 0, 1, 2, \dots k$

where $\Delta_q = \dfrac{\Delta_0}{2^q}$ $\quad q = 0, 1, 2 \dots k$

and we only use arcs of residual cap at least Δ_q in phase q

Note that in this algorithm, when we augment along a path P in phase q we augment by exactly Δ_q units even if $\delta(P) > \Delta_q$

- In phase q we consider the subnetwork $N(x, \Delta_q)$ of $N(x)$ which consists of those arcs which have residual capacity at least Δ_q
- Note that $\Delta_k = 1$ so $N(x, \Delta_k) = N(x)$
- Recall from BJG section 3.10.2 that $U_x = \{v \mid b_x(v) < b(v)\}$ and $Z_x = \{v \mid b_x(v) > b(v)\}$ and that $U_x = \emptyset \Leftrightarrow Z_x = \emptyset \Leftrightarrow x$ is feasible
- Let $E = \sum_{v \in U_x} (b(v) - b_x(v))$ E is the total excess

  If $E = 0$ the current x is optimal and feasible
- We start with $x \equiv 0$ and $\pi \equiv 0$ as there is no negative cost arc in N, the same holds for $N(x) = N$ wrt c^π
- We maintain a potential π and modify the current flow x such that we obtain $c_{ij}^\pi \geq 0$ for every arc in $N(x, \Delta_q)$ $q = 0, 1, \ldots, k$

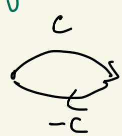
- Note that if $c^{\pi}_{ij} < 0$ when we enter phase q then

(*) $\Delta_q \le r_{ij} < 2\Delta_q = \Delta_{q-1}$

as x, π was an optimal pair for $N(x, 2\Delta_q)$ when phase $q-1$ finished.

- when we enter phase q for >0 we make sure that every arc of $N(x, \Delta_q)$ has $c^{\pi}_{ij} \ge 0$ by saturating those arcs ij for which we have $c^{\pi}_{ij} < 0$ and $r_{ij} \ge \Delta_q$. That is, we change x s.t. ij is no longer an arc of $N(x)$.

- By (*) saturating all arcs ij with $c^{\pi}_{ij} < 0$ will change E by at most $2m\Delta_q$



- Assume we have changed x s.t. $c^{\pi}_{ij} \ge 0$ for all $ij \in N(x, \Delta_q)$ and define $S(\Delta_q)$ and $T(\Delta_q)$ as follows

$$S(\Delta_q) = \{ v \mid b_x(v) + \Delta_q \le b(v) \}$$

$$T(\Delta_q) = \{ v \mid b_x(v) - \Delta_q \ge b(v) \}$$

Algorithm idea:

- start with $x \equiv 0$, $\pi \equiv 0$ and $\Delta_0 = 2^k$
- In phase q we first saturate some arcs s.t. the new x satisfies $c_{ij}^{\pi} \ge 0$ for all arcs ij in $N(x, \Delta_q)$

  Now augment along $(S(\Delta_q), T(\Delta_q))$-paths as long as such a path exists while updating x, π and $S(\Delta_q)$, $T(\Delta_q)$.
- when no more $(S(\Delta_q), T(\Delta_q))$-paths go to phase $q+1$ if $q < k$ or stop if $q = k$.

begin

- Initialize $x \equiv 0$, $\pi \equiv 0$ $\Delta = 2^k$, when $k = \lfloor \log_2 U \rfloor$
- while $\Delta \geq 1$ do
  - $\forall ij$ in $N(x)$ do
    - if $r_{ij} \geq \Delta$ and $c^{\pi}_{ij} < 0$ then
      - update x by sending r_{ij} units along ij in $N(x)$

  Let $S(\Delta) = \{v \mid b_x(v) + \Delta \leq b(v)\}$ and $T(\Delta) = \{v \mid b_x(v) - \Delta \geq b(v)\}$

  while there exists an $(S(\Delta), T(\Delta))$-path in $N(x, \Delta)$ do

  while then exists an $s \in S(\Delta)$ and $t \in T(\Delta)$ s.t. $N(x, \Delta)$ has an (s,t)-path
  - select $s \in S(\Delta)$ and $t \in T(\Delta)$ s.t. $N(x,\Delta)$ has an (s,t)-path
  - calculate shortest path distances $d(\cdot)$ from s in $N(x, \Delta)$ with respect to the reduced costs c^{π}_{ij} and let P be a shortest (s,t)-path
  - update $\pi \leftarrow \pi - d$
  - Augment by Δ units along P
  - update x, $S(\Delta)$, $T(\Delta)$ and $N(x, \Delta)$

  end

  $\Delta \leftarrow \Delta / 2$

  end

end

Theorem Suppose $N = (V, A, \ell \equiv 0, u, b, c)$ has a feasible flow. Then the scaling algorithm will find an optimal (min cost) feasible flow in N in time

$$O(m \log U \, S(n, m, nC))$$

<u>Theorem</u> suppose $N = (V, A, l \equiv 0, u, b, c)$ has a feasible flow. Then the scaling algorithm will find an optimal (min cost) feasible flow in N in time

$$O(m \log U \, S(n, m, nC))$$

<u>P:</u> Recall that if N has a feasible flow x' then we can find a flow $\tilde{x}$ in $N(x,1) = N(x)$ s.t $x' = x \oplus \tilde{x}$

when x is the flow when we enter phase k and have saturated arcs of $N(x,1)$ with $c^{\pi}_{ij} < 0$

Hence, since $N(x,1) = N(x)$ the algorithm will terminate with a feasible flow x^*

The flow x^* is optimal because

$$c^{\pi}_{ij} \geq 0 \quad \forall \text{ arcs } ij \text{ in } N(x^*)$$

when π is the final potential

Hence we just need to prove the complexity bound $O(m \log U \, S(n, m, nC))$

There are $k+1 = O(\log U)$ phases and each shortest (s,t)-path can be found in time $O(S(n,m,nC))$ so it is enough to prove that there are $O(n+m)$ augmentations in each phase.

Recall that $U_X = \{v \mid b_X(v) < b(v)\}$, $Z_X = \{v \mid b_X(v) > b(v)\}$

Consider a flow decomposition of a feasible flow x' in N into at most $n+m$ paths $P_1, P_2, \dots P_r$ $r \leq n+m$ and some cycles.

each of them have capacity at most $U \leq 2\Delta_0$

This implies that $E \leq 2(n+m)\Delta_0$ when
$E = \sum_{v \in U_X} (b(v) - b_X(v))$ is the total excess

- This implies that in phase 0 we have at most $2(n+m)$ augmentations as each decreases E by exactly Δ_0 units

- Recall that we leave a phase q when there is no path from $S(\Delta_q)$ to $T(\Delta_q)$
- Let us bound E when we enter phase $q+1$:
  Let x be our current flow, let x' be a feasible flow in N and let $\tilde{x} \in N(x)$ satisfy that $x' = x \oplus \tilde{x}$
- By flow decomposition, $\tilde{x}$ decomposes into at most $n+m$ paths and cycles and each path has flow value less than Δ_q as there is no $(S(\Delta_q), T(\Delta_q))$-path
- As path in the flow decomposition starts in U_X and ends in Z_X this implies that $E \leq (n+m)\Delta_q$

Thus when we enter phase $q+1$ we have $\Delta_{q+1} = \frac{\Delta_q}{2}$

so $E \leq 2(n+m)\Delta_{q+1}$ holds

We start phase $q+1$ by saturating arcs with $r_{ij} \geq \Delta_{q+1}$ and $c_{ij}^{\pi} < 0$

Recall that such arcs have $2\Delta_{q+1} > r_{ij} \geq \Delta_{q+1}$

since every arc of $N(x, \Delta_q)$ has $c_{ij}^{\pi} \geq 0$.

Thus saturating all such arcs changes E by at most $2\Delta_{q+1} \cdot m$

So when the while loop starts we have

$$E \leq 2(n+m)\Delta_{q+1} + 2m\Delta_{q+1}$$
$$\leq 4(n+m)\Delta_{q+1}$$

This implies that there are at most $4(n+m)$ augmentations in phase $q+1$

This argument holds for all phases $1, 2 \ldots k$, showing that each phase has $O(n+m)$ augmentations so the proof is complete $\square$.