

The Push-Relabel Algorithm

CSCI 4113 / 6101



Augmenting Paths vs Push-Relabel

Augmenting paths algorithms

- Maintain a flow
- Augment until it is optimal

Augmenting Paths vs Push-Relabel

Augmenting paths algorithms

- Maintain a flow
- Augment until it is optimal

Push-relabel algorithms

- Maintain optimal preflow
- Reduce infeasibility until feasible

Preflow:

- Satisfies capacity constraints
- May violate flow conservation

The Initial Preflow



$$f(s, v) = c(s, v) \quad \forall v \in V$$

$$f(u, v) = 0 \quad \forall (u, v) \in E, u \neq s$$

Towards Feasibility

$$\text{Excess } e(u) = \sum_{v \in V} (f(v, u) - f(u, v)) \quad (\text{Net in-flow of } u)$$

$(u \notin \{s, t\})$

Towards Feasibility

$$\text{Excess } e(u) = \sum_{v \in V} (f(v, u) - f(u, v)) \quad (\text{Net in-flow of } u)$$

$(u \notin \{s, t\})$

Feasible flow: All vertices have excess 0

Towards Feasibility

$$\text{Excess } e(u) = \sum_{v \in V} (f(v, u) - f(u, v)) \quad (\text{Net in-flow of } u) \\ (u \notin \{s, t\})$$

Feasible flow: All vertices have excess 0

Improve feasibility: Push excess to out-neighbours in G_f without exceeding residual edge capacities.

Towards Feasibility

$$\text{Excess } e(u) = \sum_{v \in V} (f(v, u) - f(u, v)) \quad (\text{Net in-flow of } u)$$

$(u \notin \{s, t\})$

Feasible flow: All vertices have excess 0

Improve feasibility: Push excess to out-neighbours in G_f without exceeding residual edge capacities.

$\Rightarrow f$ remains a preflow

Towards Feasibility

$$\text{Excess } e(u) = \sum_{v \in V} (f(v, u) - f(u, v)) \quad (\text{Net in-flow of } u) \\ (u \notin \{s, t\})$$

Feasible flow: All vertices have excess 0

Improve feasibility: Push excess to out-neighbours in G_f without exceeding residual edge capacities.

$\Rightarrow f$ remains a preflow

- If the out-neighbour is s or t , the total excess is reduced.

Node Heights

Label or height $h(v)$ of every node $v \in V$ used to guide excess elimination.

Node Heights

Label or height $h(v)$ of every node $v \in V$ used to guide excess elimination.

Valid height function:

- $h(s) = n$
- $h(t) = 0$
- $\forall (u, v) \in G_f : h(u) \leq h(v) + 1$

Node Heights

Label or height $h(v)$ of every node $v \in V$ used to guide excess elimination.

Valid height function:

- $h(s) = n$
- $h(t) = 0$
- $\forall (u, v) \in E_f : h(u) \leq h(v) + 1$

Intuition:

- Water flows downhill
- Too steep is dangerous

Basic Operations

$u \in V$ with $e(u) > 0$

Basic Operations

$u \in V$ with $e(u) > 0$

Push: If $\exists (u, v) \in G_f$ with
 $h(u) = h(v) + 1$

Basic Operations

$u \in V$ with $e(u) > 0$

Push: If $\exists (u, v) \in G_f$ with
 $h(u) = h(v) + 1$

$\Rightarrow$ Move $\delta = \min(e(u), c_f(u, v))$
flow from u to v

Basic Operations

$u \in V$ with $e(u) > 0$

Push: If $\exists (u, v) \in G_f$ with
 $h(u) = h(v) + 1$

$\Rightarrow$ Move $\delta = \min(e(u), c_f(u, v))$
flow from u to v

If $(u, v) \in G$: $f(u, v) = f(u, v) + \delta$

If $(v, u) \in G$: $f(v, u) = f(v, u) - \delta$

Basic Operations

$u \in V$ with $e(u) > 0$

Push: If $\exists (u, v) \in G_f$ with
 $h(u) = h(v) + 1$

Relabel: If $h(u) \leq h(v) \quad \forall (u, v) \in G_f$

$\Rightarrow$ Move $\delta = \min(e(u), c_f(u, v))$
flow from u to v

If $(u, v) \in G$: $f(u, v) = f(u, v) + \delta$

If $(v, u) \in G$: $f(v, u) = f(v, u) - \delta$

Basic Operations

$u \in V$ with $e(u) > 0$

Push: If $\exists (u, v) \in G_f$ with
 $h(u) = h(v) + 1$

$\Rightarrow$ Move $\delta = \min(e(u), c_f(u, v))$
flow from u to v

If $(u, v) \in G$: $f(u, v) = f(u, v) + \delta$

If $(v, u) \in G$: $f(v, u) = f(v, u) - \delta$

Relabel: If $h(u) \leq h(v) \ \forall (u, v) \in G_f$

$\Rightarrow$ Update $h(u) = 1 + \min_{(u, v) \in G_f} h(v)$

The Complete Push-Relabel Algorithm

Initialization:

$$f(u,v) = \begin{cases} c(u,v) & \text{if } u=s \\ 0 & \text{if } u \neq s \end{cases}$$

$$h(u) = \begin{cases} n & \text{if } u=s \\ 0 & \text{if } u \neq s \end{cases}$$

Turn f into a feasible flow:

while $\exists u \in V : e(u) > 0$ do

if $\exists (u,v) \in G_f : h(u) = h(v) + 1$

then

Push(u,v)

else

Relabel(u)

Correctness Proof of Push-Relabel Algorithm

The algorithm
maintains a valid
height function

Push maintains a
valid height function

Relabel(u) increases $h(u)$

Relabel maintains a
valid height function

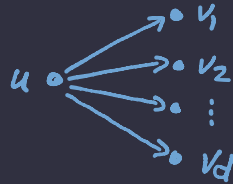
While f is not a flow,
Push or Relabel is applicable

At no time is there an
 st -path in G_f

When the algorithm terminates, f is a maximum st -flow.

Relabel(u) Increases $h(u)$

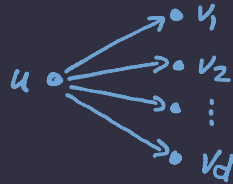
$v_1, \dots, v_d = u$'s out-neighbours in G_f



After Relabel(u), $h(u) = 1 + \min_{1 \leq i \leq d} h(v_i)$

Relabel(u) Increases $h(u)$

$v_1, \dots, v_d = u$'s out-neighbours in G_f



After Relabel(u), $h(u) = 1 + \min_{1 \leq i \leq d} h(v_i)$

Before Relabel(u), $h(u) \leq \min_{1 \leq i \leq d} h(v_i)$:

Otherwise, $\exists 1 \leq i \leq d: h(u) = h(v_i) + 1$ because $h(u) \leq h(v_i) + 1 \forall 1 \leq i \leq d$.

$\Rightarrow$ Would have called Push(u, v_i) instead.

Correctness Proof of Push-Relabel Algorithm

The algorithm
maintains a valid
height function

Push maintains a
valid height function

Relabel(u) increases $h(u)$

Relabel maintains a
valid height function

While f is not a flow,
Push or Relabel is applicable

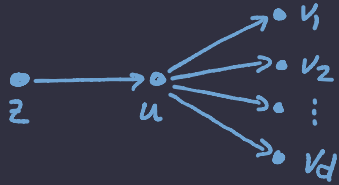
At no time is there an
 st -path in G_f

When the algorithm terminates, f is a maximum st -flow.

Relabelling Maintains Valid Height Function

$v_1, \dots, v_d = u$'s out-neighbours in G_f

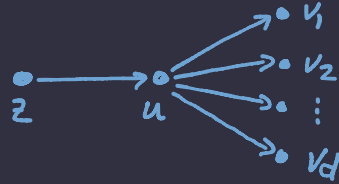
$z =$ in-neighbour of u in G_f



Relabelling Maintains Valid Height Function

$v_1, \dots, v_d = u$'s out-neighbours in G_f

$z =$ in-neighbour of u in G_f

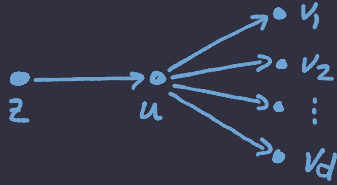


$$h(u) = 1 + \min_{1 \leq i \leq d} h(v_i) \Rightarrow h(u) \leq h(v_i) + 1 \quad \forall 1 \leq i \leq d.$$

Relabelling Maintains Valid Height Function

$v_1, \dots, v_d = u$'s out-neighbours in G_f

$z =$ in-neighbour of u in G_f



$$h(u) = 1 + \min_{1 \leq i \leq d} h(v_i) \Rightarrow h(u) \leq h(v_i) + 1 \quad \forall 1 \leq i \leq d.$$

- Before $\text{Relabel}(u)$, $h(z) \leq h(u) + 1$

- $\text{Relabel}(u)$ increases $h(u)$

$\Rightarrow h(z) \leq h(u) + 1$ after $\text{Relabel}(u)$

Correctness Proof of Push-Relabel Algorithm

The algorithm
maintains a valid
height function

Push maintains a
valid height function

Relabel(u) increases $h(u)$

Relabel maintains a
valid height function

While f is not a flow,
Push or Relabel is applicable

At no time is there an
 st -path in G_f

When the algorithm terminates, f is a maximum st -flow.

There Is No st-Path in G_f

Assume such a path exists:



There Is No st-Path in G_f

Assume such a path exists:



- $k \leq n-1$

There Is No st-Path in G_f

Assume such a path exists:



- $k \leq n-1$
- $h(x_{i-1}) \leq h(x_i) + 1 \quad \forall 1 \leq i \leq k$

There Is No st-Path in G_f

Assume such a path exists:



- $k \leq n-1$
- $h(x_{i-1}) \leq h(x_i) + 1 \quad \forall 1 \leq i \leq k$

$$\Rightarrow h(s) \leq h(t) + n - 1$$

There Is No st-Path in G_f

Assume such a path exists:



- $k \leq n-1$
- $h(x_{i-1}) \leq h(x_i) + 1 \quad \forall 1 \leq i \leq k$

$$\Rightarrow h(s) \leq h(t) + n - 1$$

$$h(s) = n \quad h(t) = 0 \quad \Downarrow$$

Running Time of Push-Relabel Algorithm

- Maintain adjacency list representation of G_f
- Adjacency lists split into "pushable" / "non-pushable" edges
- Maintain list of excess vertices

Running Time of Push-Relabel Algorithm

- Maintain adjacency list representation of G_f
- Adjacency lists split into "pushable" / "non-pushable" edges
- Maintain list of excess vertices

Operation

Choose excess vertex

Decide whether to Push or Relabel

Push

Relabel

Running Time of Push-Relabel Algorithm

- Maintain adjacency list representation of G_f
- Adjacency lists split into "pushable" / "non-pushable" edges
- Maintain list of excess vertices

Operation

Cost

Choose excess vertex

$O(1)$

Decide whether to Push or Relabel

$O(1)$

Push

$O(1)$

Relabel

$O(\deg(u))$

Running Time of Push-Relabel Algorithm

- Maintain adjacency list representation of G_f
- Adjacency lists split into "pushable" / "non-pushable" edges
- Maintain list of excess vertices

<u>Operation</u>	<u>Cost</u>	<u>Count</u>
Choose excess vertex	$O(1)$	
Decide whether to Push or Relabel	$O(1)$	
Push	$O(1)$	$O(n^2m)$
Relabel	$O(\deg(u))$	$O(n^2)$

Running Time of Push-Relabel Algorithm

- Maintain adjacency list representation of G_f
- Adjacency lists split into "pushable" / "non-pushable" edges
- Maintain list of excess vertices

<u>Operation</u>	<u>Cost</u>	<u>Count</u>
Choose excess vertex	$O(1)$	$O(n^2m)$
Decide whether to Push or Relabel	$O(1)$	$O(n^2m)$
Push	$O(1)$	$O(n^2m)$
Relabel	$O(\deg(u))$	$O(n^2)$

Running Time of Push-Relabel Algorithm

- Maintain adjacency list representation of G_f
- Adjacency lists split into "pushable" / "non-pushable" edges
- Maintain list of excess vertices

<u>Operation</u>	<u>Cost</u>	<u>Count</u>	<u>Total cost</u>
Choose excess vertex	$O(1)$	$O(n^2m)$	$O(n^2m)$
Decide whether to Push or Relabel	$O(1)$	$O(n^2m)$	$O(n^2m)$
Push	$O(1)$	$O(n^2m)$	$O(n^2m)$
Relabel	$O(\deg(u))$	$O(n^2)$	$O(nm)$

Running Time of Push-Relabel Algorithm

- Maintain adjacency list representation of G_f
- Adjacency lists split into "pushable" / "non-pushable" edges
- Maintain list of excess vertices

<u>Operation</u>	<u>Cost</u>	<u>Count</u>	<u>Total cost</u>
Choose excess vertex	$O(1)$	$O(n^2m)$	$O(n^2m)$
Decide whether to Push or Relabel	$O(1)$	$O(n^2m)$	$O(n^2m)$
Push	Faster variants reduce $O(1)$	$O(n^2m)$	$O(n^2m)$
Relabel	this to $O(n^3)$ or $O(n^2\sqrt{m})$ $O(\deg(u))$	$O(n^2)$	$O(nm)$

Counting Relabel Operations

Claim: For every excess vertex u , there exists a path from u to s in G_f .

Counting Relabel Operations

Claim: For every excess vertex u , there exists a path from u to s in G_f .

$$\Rightarrow h(u) \leq h(s) + n - 1 = 2n - 1$$

Counting Relabel Operations

Claim: For every excess vertex u , there exists a path from u to s in G_f .

$$\Rightarrow h(u) \leq h(s) + n - 1 = 2n - 1$$

- Initially, $h(u) = 0$
- $\text{Relabel}(u)$ increases $h(u)$

Counting Relabel Operations

Claim: For every excess vertex u , there exists a path from u to s in G_f .

$$\Rightarrow h(u) \leq h(s) + n - 1 = 2n - 1$$

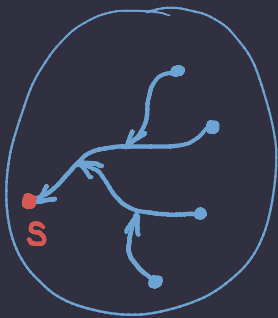
- Initially, $h(u) = 0$
- $\text{Relabel}(u)$ increases $h(u)$

$\Rightarrow$ Every vertex is relabelled at most $2n - 1$ times.

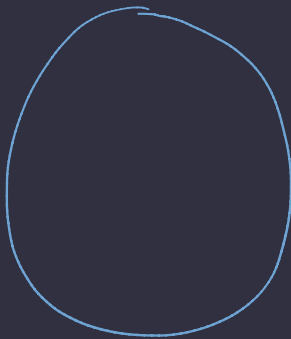
Counting Relabel Operations

Claim: For every excess vertex u , there exists a path from u to s in G_f .

Proof: $C =$ all vertices that can reach s in G_f



C



$V \setminus C$

Assume $\exists u \in V \setminus C : e(u) > 0$

Assume $\exists u \in V \setminus C : e(u) > 0$

$$0 < \sum_{u \in V \setminus C} e(u) = \sum_{u \in V \setminus C} \sum_{v \in V} (f(v, u) - f(u, v))$$

Assume $\exists u \in V \setminus C : e(u) > 0$

$$\begin{aligned} 0 < \sum_{u \in V \setminus C} e(u) &= \sum_{u \in V \setminus C} \sum_{v \in V} (f(v, u) - f(u, v)) \\ &= \sum_{u \in V \setminus C} \sum_{v \in C} (f(v, u) - f(u, v)) + \sum_{u \in V \setminus C} \sum_{v \in V \setminus C} (f(v, u) - f(u, v)) \end{aligned}$$

Assume $\exists u \in V \setminus C : e(u) > 0$

$$\begin{aligned} 0 < \sum_{u \in V \setminus C} e(u) &= \sum_{u \in V \setminus C} \sum_{v \in V} (f(v, u) - f(u, v)) \\ &= \sum_{u \in V \setminus C} \sum_{v \in C} (f(v, u) - f(u, v)) + \sum_{u \in V \setminus C} \sum_{v \in V \setminus C} (f(v, u) - f(u, v)) \\ &\leq \sum_{v \in V} \sum_{u \in V \setminus C} f(v, u) \end{aligned}$$

Assume $\exists u \in V \setminus C : e(u) > 0$

$$\begin{aligned} 0 < \sum_{u \in V \setminus C} e(u) &= \sum_{u \in V \setminus C} \sum_{v \in V} (f(v, u) - f(u, v)) \\ &= \sum_{u \in V \setminus C} \sum_{v \in C} (f(v, u) - f(u, v)) + \sum_{u \in V \setminus C} \sum_{v \in V \setminus C} (f(v, u) - f(u, v)) \\ &\leq \sum_{v \in V} \sum_{u \in V \setminus C} f(v, u) \end{aligned}$$

$\Rightarrow \exists v \in C, u \notin C : f(v, u) > 0$

Assume $\exists u \in V \setminus C : e(u) > 0$

$$\begin{aligned} 0 < \sum_{u \in V \setminus C} e(u) &= \sum_{u \in V \setminus C} \sum_{v \in V} (f(v, u) - f(u, v)) \\ &= \sum_{u \in V \setminus C} \sum_{v \in C} (f(v, u) - f(u, v)) + \sum_{u \in V \setminus C} \sum_{v \in V \setminus C} (f(v, u) - f(u, v)) \\ &\leq \sum_{v \in V} \sum_{u \in V \setminus C} f(v, u) \end{aligned}$$

$\Rightarrow \exists v \in C, u \notin C : f(v, u) > 0$

$\Rightarrow (u, f) \in G_f$

Assume $\exists u \in V \setminus C : e(u) > 0$

$$\begin{aligned} 0 < \sum_{u \in V \setminus C} e(u) &= \sum_{u \in V \setminus C} \sum_{v \in V} (f(v, u) - f(u, v)) \\ &= \sum_{u \in V \setminus C} \sum_{v \in C} (f(v, u) - f(u, v)) + \sum_{u \in V \setminus C} \sum_{v \in V \setminus C} (f(v, u) - f(u, v)) \\ &\leq \sum_{v \in V} \sum_{u \in V \setminus C} f(v, u) \end{aligned}$$

$\Rightarrow \exists v \in C, u \notin C : f(v, u) > 0$

$\Rightarrow (u, f) \in G_f$

$\Rightarrow u \in C \quad \downarrow$

Counting Push Operations

Two types of Push operations:

Saturating push = $\text{Push}(u, v)$ that moves $c_f(u, v)$ flow from u to v

Non-saturating push = $\text{Push}(u, v)$ that moves $< c_f(u, v)$ flow from u to v

Counting Push Operations

Two types of Push operations:

Saturating push = Push(u, v) that moves $c_f(u, v)$ flow from u to v

$\Rightarrow (u, v)$ disappears from G_f

$\Rightarrow$ Need Push(v, u) before next Push(u, v).

Non-saturating push = Push(u, v) that moves $< c_f(u, v)$ flow from u to v

Counting Push Operations

Two types of Push operations:

Saturating push = $\text{Push}(u, v)$ that moves $c_f(u, v)$ flow from u to v

$\Rightarrow (u, v)$ disappears from G_f

$\Rightarrow$ Need $\text{Push}(v, u)$ before next $\text{Push}(u, v)$.

Non-saturating push = $\text{Push}(u, v)$ that moves $< c_f(u, v)$ flow from u to v

$\Rightarrow e(u) = 0$ afterwards

But can be done multiple times without intervening $\text{Push}(v, u)$.

Counting Push Operations

Two types of Push operations:

Saturating push = $\text{Push}(u, v)$ that moves $c_f(u, v)$ flow from u to v

$\Rightarrow (u, v)$ disappears from G_f

$\Rightarrow$ Need $\text{Push}(v, u)$ before next $\text{Push}(u, v)$. $O(nm)$

Non-saturating push = $\text{Push}(u, v)$ that moves $< c_f(u, v)$ flow from u to v

$\Rightarrow e(u) = 0$ afterwards

But can be done multiple times without intervening $\text{Push}(v, u)$. $O(n^2m)$

Counting Saturating Push Operations

Claim: Between two saturating $\text{Push}(u, v)$ operations $h(u)$ increases.

Counting Saturating Push Operations

Claim: Between two saturating $\text{Push}(u, v)$ operations $h(u)$ increases.

$\Rightarrow$ At most $2n-1$ saturating Push operations per edges

$\Rightarrow O(nm)$ saturating Push operations in total

Counting Saturating Push Operations

Claim: Between two saturating $\text{Push}(u, v)$ operations $h(u)$ increases.

Proof: $\text{Push}(u, v)$



$\text{Push}(u, v)$



Counting Saturating Push Operations

Claim: Between two saturating $\text{Push}(u, v)$ operations $h(u)$ increases.

Proof: $\text{Push}(u, v)$



$\text{Push}(v, u)$ to desaturate (u, v)



$\text{Push}(u, v)$



Counting Saturating Push Operations

Claim: Between two saturating $\text{Push}(u, v)$ operations $h(u)$ increases.

Proof: $\text{Push}(u, v)$



$\text{Push}(v, u)$ to desaturate (u, v)



$\text{Push}(u, v)$



Counting Saturating Push Operations

Claim: Between two saturating $\text{Push}(u, v)$ operations $h(u)$ increases.

Proof: $\text{Push}(u, v)$



$$h_1(u)$$

$\text{Push}(v, u)$ to desaturate (u, v)



$$h_2(v) = h_2(u) + 1$$

$$\Rightarrow h_1(u) + 1$$

$\text{Push}(u, v)$



Counting Saturating Push Operations

Claim: Between two saturating $\text{Push}(u, v)$ operations $h(u)$ increases.

Proof: $\text{Push}(u, v)$



$$h_1(u)$$

$\text{Push}(v, u)$ to desaturate (u, v)



$$h_2(v) = h_2(u) + 1$$

$$\Rightarrow h_1(u) + 1$$

$\text{Push}(u, v)$



$$h_3(u) = h_3(v) + 1$$

$$\Rightarrow h_2(v) + 1$$

$$\Rightarrow h_1(u) + 2$$

Counting Non-Saturating Push Operations

Potential function $\Phi = \sum_{e(v) > 0} h(v)$ (Total height of all excess vertices)

Counting Non-Saturating Push Operations

Potential function $\Phi = \sum_{e(v) > 0} h(v)$ (Total height of all excess vertices)

Initially, $\Phi = 0$.

Counting Non-Saturating Push Operations

Potential function $\Phi = \sum_{e(v) > 0} h(v)$ (Total height of all excess vertices)

Initially, $\Phi = 0$.

At all times, $\Phi \geq 0$.

Counting Non-Saturating Push Operations

Potential function $\Phi = \sum_{e(v) > 0} h(v)$ (Total height of all excess vertices)

Initially, $\Phi = 0$.

At all times, $\Phi \geq 0$.

Claims:

- (i) Relabel and saturating Push operations increase Φ by $O(n^2m)$ in total.
- (ii) Every non-saturating Push operation decreases Φ by at least 1.

Counting Non-Saturating Push Operations

Potential function $\Phi = \sum_{e(v) > 0} h(v)$ (Total height of all excess vertices)

Initially, $\Phi = 0$.

At all times, $\Phi \geq 0$.

Claims:

- (i) Relabel and saturating Push operations increase Φ by $O(n^2m)$ in total.
- (ii) Every non-saturating Push operation decreases Φ by at least 1.

$\Rightarrow O(n^2m)$ non-saturating Push operations.

Potential Increase Due to Relabelling

Relabel (u): $h_1(u)$ = height of u before Relabel

$h_2(u)$ = height of u after Relabel

Φ increases by $h_2(u) - h_1(u)$

Potential Increase Due to Relabelling

Relabel (u): $h_1(u)$ = height of u before Relabel

$h_2(u)$ = height of u after Relabel

Φ increases by $h_2(u) - h_1(u)$

$\Rightarrow$ Total increase of Φ from relabelling $u \leq 2n - 1$

Potential Increase Due to Relabelling

Relabel(u): $h_1(u)$ = height of u before Relabel

$h_2(u)$ = height of u after Relabel

Φ increases by $h_2(u) - h_1(u)$

$\Rightarrow$ Total increase of Φ from relabelling $u \leq 2n - 1$

$\Rightarrow$ Total increase of Φ from relabelling $< 2n^2$

Potential Increase Due to Saturating Push Operations

$\text{Push}(u, v)$: Worst case:

- $e(u) > 0$ after Push
- $e(v) = 0$ before Push

$\Rightarrow \Phi$ increases by $h(v) \leq 2n - 1$

Potential Increase Due to Saturating Push Operations

Push(u, v): Worst case:

- $e(u) > 0$ after Push
- $e(v) = 0$ before Push

$\Rightarrow \Phi$ increases by $h(v) \leq 2n-1$

$O(nm)$ saturating Push operations

$\Rightarrow$ Total increase of Φ from saturating Push operations = $O(n^2m)$

Potential Decrease Due to Non-Saturating Push Operation

Push(u, v): Worst case:

- $e(v) = 0$ before Push
- $e(u) = 0$ after Push

$\Rightarrow \Phi$ drops by at least $h(u) - h(v)$

Potential Decrease Due to Non-Saturating Push Operation

Push(u, v): Worst case:

- $e(v) = 0$ before Push
- $e(u) = 0$ after Push

$\Rightarrow \Phi$ drops by at least $h(u) - h(v)$

$h(u) = h(v) + 1$ (because we push)

Potential Decrease Due to Non-Saturating Push Operation

Push(u, v): Worst case:

- $e(v) = 0$ before Push
- $e(u) = 0$ after Push

$\Rightarrow \Phi$ drops by at least $h(u) - h(v)$

$h(u) = h(v) + 1$ (because we push)

$\Rightarrow \Phi$ drops by at least 1.