Preflow Push Algorithms for Maximum Flow

R. Chandrasekaran

Abstract

0.1 Maximum Flow Problem:[CLRS]

INPUT: A directed simple graph $G = [V; E]$ with positive edge capacities $c(e); e \in E$ and two distinguished nodes $s, t \in V$.

Definition 1 *A feasible flow is a function $f : E \to R_+$ that satisfies:*

$$\sum_{\substack{v \in V \\ (u,v) \in E}} f(u,v) - \sum_{\substack{v \in V \\ (v,u) \in E}} f(v,u) = \begin{cases} F & u = s \\ 0 & u \neq s, t \\ -F & u = t \end{cases}$$

$$0 \leq f(u,v) \leq c(u,v) \qquad \forall (u,v) \in E$$

We want a feasible flow that maximizes F.

Definition 2 *A* ***preflow*** *is a function $f : E \to R_+$ that satisfies:*

$$e^f(u) = \sum_{\substack{v \in V \\ (v,u) \in E}} f(v,u) - \sum_{\substack{v \in V \\ (u,v) \in E}} f(u,v) \geq 0 \qquad \forall u \neq s$$

$$0 \leq f(u,v) \leq c(u,v) \qquad \forall (u,v) \in E$$

where $e^f(u)$ is called the ***excess*** *at u with respect to preflow f. This algorithm maintains a preflow at all times and ends with a feasible flow that maximizes F.*

Definition 3 *A node u for which $e^f(u) > 0$ is said to be* ***overflowing*** *with respect to f.*

Definition 4 ***Residual Capacity:*** *$c^f(u,v) = c(u,v) - f(u,v) \geq 0 \forall u \in V, v \in V$.*

Definition 5 ***Residual Graph:*** *$G^f = [V, E^f]; E^f = \{(u,v) : u \in V, v \in V, c^f(u,v) > 0\}$*

Definition 6 *A function $h^f : V \to N$ that satisfies*

$$
\begin{aligned}
h^f(s) &= |V| \\
h^f(t) &= 0 \\
h^f(u) &\leq h^f(v) + 1 \qquad \forall (u,v) \in E^f
\end{aligned}
$$

*is called a **height function** relative to f.*

Definition 7 *Given a preflow f and a height function h^f, (u,v) is called an **admissible edge** if $(u,v) \in E^f$ [i.e. $c^f(u,v) > 0$], u is overflowing [i.e $e^f(u) > 0$], and $h^f(u) = h^f(v) + 1$.*

Lemma 1 *Let $G = [V; E]$ be a flow network. Let f be a preflow in G and let h^f be a height function relative to f. Then $[h^f(u) > h^f(v)+1] \Rightarrow (u,v) \notin E^f$.*

PUSH(u,v):

1. // **Applies when**: $e^f(u) > 0; c^f(u,v) > 0; h^f(u) = h^f(v) + 1$. This means that (u,v) is an admissible edge.
2. //**Action**: Push $\Delta^f(u,v) = \min[e^f(u), c^f(u,v)]$ units of flow from u to v along edge $(u,v) \in E^f$.
3. $\Delta^f(u,v) = \min[e^f(u), c^f(u,v)]$
4. **if** $(u,v) \in E^f$
5. $\quad f(u,v) \leftarrow f(u,v) + \Delta^f(u,v)$
6. **else** $f(v,u) \leftarrow f(v,u) - \Delta^f(u,v)$
7. $e^f(u) \leftarrow e^f(u) - \Delta^f(u,v)$
8. $e^f(v) \leftarrow e^f(v) + \Delta^f(u,v)$

Definition 8 *If $\Delta^f(u,v) = c^f(u,v)$, this PUSH operation is called a **saturating push**; else, it is a **nonsaturating push**.*

Lemma 2 *If PUSH(u,v) is a nonsaturating push, then u is no longer overflowing after this push.*

Now we describe the other operation in this algorithm.
RELABEL(u):

1. //**Applies when:** u is overflowing and $h^f(u) \leq h^f(v) \forall (u,v) \in E^f$
2. //**Action:** Increase the height of u
3. $h^f(u) \leftarrow 1 + \min[h^f(v) : v$ such that $(u,v) \in E^f]$

Remark 1 *If u is overflowing, then $\{v : (u,v) \in E^f\} \neq \phi$; actually there is a directed path from u to s in G^f.*

Lemma 3 *Let u be an overflowing vertex; i.e. $e^f(u) > 0$. Then, either there exists an applicable PUSH(u,v) for some $v \in V$ or RELABEL(u) is applicable.*
Proof. *$[(u,v) \in E^f] \Rightarrow h^f(u) \leq h^f(v) + 1$ since h^f is a height function. If PUSH does not apply to u, then $h^f(u) \leq h^f(v)$ for all $(u,v) \in E^f$ and hence RELABEL applies to u.* ■

0.2 The Main Algorithm:

GENERIC-PUSH-RELABEL(G, s, t, c)

1. INITIALIZE-PREFLOW(G, s, t, c)
2. **while** there exists applicable push or relabel operation,
3. select an applicable push or relabel operation and perform it.

INITIALIZE-PREFLOW(G, s, t, c)

1. **for** each vertex $v \in V$
2. $h^f(v) \leftarrow 0$

3. $e^f(v) \leftarrow 0$

4. **for** each edge $(u, v) \in E$

5. $f(u, v) \leftarrow 0$

6. $h^f(s) \leftarrow |V|$

7. **for** each vertex $v \in adj[s]$

8. $f(s, v) \leftarrow c(s, v)$

9. $e^f(v) \leftarrow c(s, v)$

10. $e^f(s) \leftarrow e^f(s) - c(s, v)$

1 Analysis of the Algorithm:

Lemma 4 *$h^f(u)$ is nondecreasing as the algorithm GENERIC-PUSH-RELABEL(G, s, t, c) evolves for all $u \in V$.*
Proof. *$h^f(u)$ changes during the course of the algorithm only when we relabel u. This happens when $h^f(u) \leq h^f(v)$ for all $(u, v) \in E^f$ and the new value of $h^f(u) = 1 + \min_{v:(u,v)\in E^f}[h^f(v)]$. Hence the lemma.* ∎

Lemma 5 *GENERIC-PUSH-RELABEL algorithm maintains the attribute h as a height function at all times.*
Proof. *$h^f(s) = |V|; h^f(t) = 0$ at all times during the evolution of this algorithm. $h^f(u)$ changes is when u is relabeled. The manner in which it is relabeled preserves the relation $h^f(u) \leq h^f(v) + 1 \forall (u, v) \in E^f$. PUSH$(u, v)$ may add edge (v, u) to E^f or remove (u, v) from E^f; in the former case, we have $h^f(v) = h^f(u) - 1 \leq h^f(u) + 1$. Hence the lemma follows.* ∎

Lemma 6 *Let f be a preflow and let h^f be a height function with respect to f in a flow network $G = [V, E]$. Then, there is no path in G^f from s to t.*
Proof. *Suppose there is a (simple) path $p = (s = v_0, v_1, ..., v_k = t)$ in G^f; $k \leq |V| - 1$. Since h^f is a height function, it follows that*

$$h^f(v_i) \leq h^f(v_{i+1}) + 1 \qquad \textit{for } i = 0, 1, 2, ...k - 1$$

Hence, it follows that

$$h^f(s) \leq h^f(t) + k$$

Since $h^f(t) = 0$, this implies that $h^f(s) \leq k \leq |V| - 1$; but this is a contradiction to the fact that $h^f(s) = |V|$. Hence the result follows. ■

Theorem 1 *If GENERIC-PUSH-RELABEL(G, s, t, c) algorithm terminates, then the preflow f it computes is a maximum flow from s to t in (G, c).*
Proof. *If f is a preflow at some step and we perform a PUSH operation, then resulting f is also a preflow. INITIALIZE yields a preflow. Hence, at all times, the algorithm maintains a preflow by induction. At termination, $e^f(u) = 0$ for all $u \in V - \{s\}$. Hence, f at termination is a flow in (G, c). Moreover, at termination, there is no path in G^f and hence f is a maximum flow.* ■

1.1 Complexity Analysis of the Algorithm:

Lemma 7 *Let f be a preflow in (G, s, t, c) where $G = [V, E]$ is a flow network. Then, for any overflowing vertex x, there is a simple path from x to s in G^f.*
Proof. *By induction. The only nodes that are overflowing after INITIALIZE belong to the set $\{v : c(s, v) > 0\}$. For each of these, in G^f just after INITIALIZE, there is an edge of the form (v, s). Suppose at some step of the algorithm, the result holds. Since RELABEL does not affect this, suppose we now perform a PUSH(u, v). Just before this operation, vertex u was overflowing and hence by induction hypothesis, there is a simple path in G^f from u to x in the preceding residual graph G^f. This PUSH does not affect that path. After this PUSH there is the edge (v, u) in the new residual graph and hence a path from v to s. This takes care of the possibility that vertex v is overflowing after the PUSH.* ■

Lemma 8 *At any time during GENERIC-PUSH-RELABEL(G, s, t, c), $h^f(u) \leq 2|V| - 1$ for all $u \in V$.*
Proof. *The statement is clearly true for $u = s, t$. For any other vertex u, $h^f(u)$ is initially equal to $0 \leq 2|V| - 1$. We will show that after each RELABEL(u), the result holds. When u is relabeled, it is overflowing (both before and after RELABEL) and hence there is a simple path $p = (u = v_0, v_1, ..., v_k = s)$ from u to s in G^f with $k \leq |V| - 1$. Moreover, $(v_i, v_{i+1}) \in E^f$ for $0 \leq i \leq k - 1$. Hence, $h^f(v_i) \leq h^f(v_{i+1}) + 1$ for $0 \leq i \leq k - 1$. Hence $h^f(u) \leq h^f(s) + |V| - 1 \leq 2|V| - 1$.* ■

There are three types of operations in this algorithm: RELABEL, saturating PUSH, and nonsaturating PUSH. In the analysis phase,we separately bound the number of each of these types of operations to get an overall complexity of $O(V^2E)$ for the generic algorithm which does not specify the order in which these are to be carried out. By doing them in special ways, we can save further to reduce the complexity to $O(V^3)$.

1.1.1 Bound on number of RELABEL Operations:

Since $h^f(u) \leq 2|V| - 1$ for all $u \in V - \{s,t\}$, and $h^f(s), h^f(t)$ do not change, each RELABEL(u) increases $h^f(u)$ by at least 1, the number of RELABEL operations is bounded by $(2|V| - 1)(|V| - 2) = O(|V|^2)$.

1.1.2 Bound on number of Saturating PUSH Operations:

For any two vertices u and v, let us count the number of saturating pushes for both (u,v) and (v,u) together. If there are such pushes, then either (u,v) or (v,u) or both belong to E. Suppose a saturating push from u to v has occurred. $h^f(v) = h^f(u) - 1$. For another saturating push from u to v to occur later, the algorithm must first push flow from v to u which cannot happen until $h^{f'}(v) = h^{f'}(u) + 1$. Since $h^f(u)$ never decreases for any vertex u, in order for this to happen $h^f(v)$ needs to increase at least by 2. Likewise $h^f(u)$ must also increases by 2 between successive saturating pushes from v to u. Height starts at 0 and is always $\leq 2|V| - 1$. Hence, the number of times this happens (the height increases by 2) is no more than $|V|$. Hence the number of saturating pushes using the vertices u and v is no more than $2|V|$ and therefore the total number of these is no more than $2|V||E|$.

1.1.3 Bound on number of Nonsaturating PUSH Operations:

Let

$$\Phi = \sum_{v:e^f(v)>0} h^f(v)$$

be a potential function. Initially, $\Phi = 0$, and this may change after each RELABEL, saturating PUSH, and nonsaturating PUSH operations. Since we have bounds on the first two operations, we can bound their contributions to Φ. Then we show that nonsaturating PUSH operations reduce Φ by at least 1. This will then give us an upper bound on nonsaturating PUSH operations.

Φ increases only when we RELABEL or use a saturating PUSH operation. RELABEL(u) cannot increase its height by more than its maximum height which is $2|V| - 1$. A Saturating PUSH from u to v renders $e^f(u) = 0$ after the operation but can make $e^f(v)$ positive and so increase Φ by $h^f(v)$;but in order for PUSH(u, v) to apply, we have $h^f(u) = h^f(v) + 1$ and hence the decrease in Φ cannot be less than 1. Thus, during the course of the algorithm, Φ can increase by no more than $(2|V| - 1)^2(|V| - 2) + 2|V||E|(2|V| - 1) \leq 4|V|^2(|V|+|E|)$. Since $\Phi \geq 0$, the number of nonsaturating PUSH operations is bounded by $4|V|^2(|V| + |E|)$. RELABEL takes $O(|V|)$; PUSH takes $O(1)$ time; Selecting the operation that applies takes $O(1)$ time with appropriate data structure. [Show this]. Hence the entire algorithm takes $O|V|^2|E|)$ time. To reduce this to $O(|V|^3)$ use RELABEL-TO-THE-FRONT approach.