

**FINDING MAXIMUM FLOWS IN NETWORKS
WITH NONZERO LOWER BOUNDS
USING PREFLOW METHODS**

Tongnyoung Yi

and

Katta G. Murty

Department of Industrial and Operations Engineering

The University of Michigan

Ann Arbor, MI 48109-2117

Technical Report 91-21

July 1991

Finding Maximum Flows in Networks with Nonzero Lower Bounds using Preflow Methods

Tongnyoul Yi and Katta G. Murty

Department of Industrial and Operations Engineering
University of Michigan
Ann Arbor, Michigan 48109-2117, U.S.A.

July 1991

Abstract

There are methods such as preflow methods(for example, the preflow push algorithm by Goldberg and Tarjan[2, 3]) which can be used directly to find maximum flows only in networks with zero lower bounds. We describe a simple two phase approach for finding a maximum flow in a network with nonzero lower bounds, using such an algorithm.

Let $G = (\mathcal{N}, \mathcal{A}, \vec{\ell}, \vec{k}, \check{s}, \check{t})$ denote a pure directed single commodity flow network with \mathcal{N} = set of nodes; \mathcal{A} = set of arcs; $\vec{\ell} = (\vec{\ell}_{ij})$, $\vec{k} = (\vec{k}_{ij})$, the vectors of lower and upper bounds (upper bounds are also called capacities) for arc flows in G satisfying $\vec{\ell} \leq \vec{k}$, $\vec{\ell} \neq 0$; \check{s}, \check{t} , the source and sink nodes in G . A *feasible flow vector* $f = (f_{ij})$ is a vector of flow f_{ij} defined over arcs $(i, j) \in \mathcal{A}$ satisfying

$$\vec{\ell} \leq f \leq \vec{k} \quad (1)$$

$$f(i, \mathcal{N}) - f(\mathcal{N}, i) = \begin{cases} 0, & \text{if } i \neq \check{s}, \check{t} \\ v, & \text{if } i = \check{s} \\ -v, & \text{if } i = \check{t} \end{cases} \quad (2)$$

where for any $i \in \mathcal{N}$, $f(i, \mathcal{N}) =$ sum of f_{ij} over j such that $(i, j) \in \mathcal{A}$ and $f(\mathcal{N}, i) =$ sum of f_{ji} over j such that $(j, i) \in \mathcal{A}$.

The maximum flow problem in G is to find a feasible flow vector with the maximum possible value. The conditions (1) are the bound constraints. (2) are known as the *flow conservation equations*.

If there are two or more parallel arcs joining the same pair of nodes in G , they can all be combined into a single arc with the same orientation, with bounds obtained by summing up the bounds on the combined arcs. So, without any loss of generality, we assume that there are no parallel arcs in G .

There is one significant difference between networks with zero lower bounds, and those with nonzero lower bounds. If the lower bound vector is zero, there is always a feasible flow vector, since the vector 0 is itself feasible. If the lower bound vector is nonzero, it is possible that there is no feasible flow vector.

There are a class of methods for the maximum flow problem in networks with zero lower bounds based on preflow. A *preflow* is a vector defined on \mathcal{A} satisfying (1); and (2) as \leq inequalities for $i \neq \check{s}$ or \check{t} , instead of equalities as required. An example of this type of method is the recently developed preflow push algorithm due to Goldberg and Tarjan [2, 3]. These methods obtain a sequence of preflows and terminate only when the preflow becomes a feasible flow vector, and at that time it will be a maximum flow. That is why they do not detect infeasibility directly, and are thus applied only for solving the maximum flow problem in networks with zero lower bounds.

In this note we develop a simple two phase approach for tackling the maximum flow problem in a network with nonzero lower bounds, using such an algorithm.

Given a feasible flow vector $f = (f_{ij})$ in our network G , a path \mathcal{P} from \check{s} to \check{t} in G is said to be a *flow augmenting path* (FAP) with respect to f if

$$f_{ij} < \vec{k}_{ij} \text{ for all forward arcs } (i, j) \text{ on } \mathcal{P}$$

$$f_{ij} > \vec{k}_{ij} \text{ for all reverse arcs } (i, j) \text{ on } \mathcal{P}$$

The classical result of Ford and Fulkerson[1] states that a feasible flow vector in a network is a maximum value flow if and only if there exists no FAP from the source to the sink with respect to it. This result is the one most commonly used to check whether a given feasible flow vector is a maximum value one, and is the basis for several augmenting path methods for the maximum flow problem.

Given a feasible flow vector $f = (f_{ij})$ in G , construct a set of arcs denoted by $\mathcal{A}(f)$ by the following rules:

- (i) for each $(i, j) \in \mathcal{A}$ satisfying $f_{ij} < \check{k}_{ij}$, include the arc (i, j) in $\mathcal{A}(f)$ with a + label, lower bound 0, and capacity(or upper bound) $\check{\kappa}_{ij} = \check{k}_{ij} - f_{ij}$. This + labeled arc (i, j) in $G(f)$ is said to correspond to the arc (i, j) in G .
- (ii) for each $(i, j) \in \mathcal{A}$ satisfying $f_{ij} > \check{l}_{ij}$, include the arc (j, i) in $\mathcal{A}(f)$ with a - label, lower bound 0, and capacity $\check{\kappa}_{ji} = f_{ij} - \check{l}_{ij}$. This - labeled arc (j, i) in $G(f)$ is said to correspond to the arc (i, j) in G .

$\mathcal{A}(f)$ is called the set of residual arcs, and the network $G(f) = (\mathcal{N}, \mathcal{A}(f), 0, \check{\kappa} = (\check{\kappa}_{ij}), \check{s}, \check{t})$ is known as the residual network with respect to f . Notice that $|\mathcal{A}(f)| \leq 2|\mathcal{A}|$, i.e., the number of arcs in the residual network is at most twice the number in the original network.

Let *Algorithm 1* refer to any algorithm such as preflow push algorithm which can process directly only maximum flow problems in networks with zero lower bounds. The two phase procedure for processing the maximum flow problem in our network G with lower bound vector $\check{l} \neq 0$ by *Algorithm 1* is the following:

Phase I Transform the problem of finding a feasible flow vector in G into that of finding a maximum flow in an augmented network G^* in which lower bounds for all arc flows are zero, by well known classical techniques(See Ford and Fulkerson[1] or Murty[4, section 2.6]). Solve the maximum flow problem in G^* by *Algorithm 1* directly. From this either conclude that there is no feasible flow vector in G and terminate, or obtain a feasible flow vector in it. Suppose a feasible flow vector $\bar{f} = (\bar{f}_{ij})$ of value \bar{v} has been found in G . Go to phase II.

Phase II Construct the residual network $G(\bar{f})$ as described above. Since the lower bound vector in $G(\bar{f})$ is 0 by definition, a maximum flow in $G(\bar{f})$ from \check{s} to \check{t} can be found by *Algorithm 1* directly, which do. To avoid confusion with flow vector in the original network G , we denote the maximum flow obtained in $G(\bar{f})$ by $\bar{g} = (\bar{g}_{pq})$, and its value by ω . Lower bounds in the residual network $G(\bar{f})$ are 0, and for a pair of nodes p, q , if there are arcs (p, q) and (q, p) both with positive flows in \bar{g} , then the flows on them can be canceled(i.e., replace the larger of $\bar{g}_{pq}, \bar{g}_{qp}$ by their difference and the smaller by 0) and at least one of these flows converted to 0, leading to

another feasible flow vector in $G(\bar{f})$ of the same maximum value. We assume that this is done. So, without loss of generality, we assume that for any pair of nodes, the vector \bar{g} has positive flows in at most one of the two orientations for arcs joining them.

Define the flow vector $\hat{f} = (\hat{f}_{ij})$ in G , where for $(i, j) \in \mathcal{A}$

$$\hat{f}_{ij} = \begin{cases} \bar{f}_{ij} + \bar{g}_{ij}, & \text{if there is a } + \text{ labeled corresponding arc } (i, j) \text{ in } G(\bar{f}) \text{ with } \bar{g}_{ij} > 0 \\ \bar{f}_{ij} - \bar{g}_{ji}, & \text{if there is a } - \text{ labeled corresponding arc } (j, i) \text{ in } G(\bar{f}) \text{ with } \bar{g}_{ji} > 0 \\ \bar{f}_{ij}, & \text{otherwise} \end{cases}$$

Then \hat{f} is a maximum value feasible flow vector in G and its value is $\bar{v} + \omega$. Terminate.

THEOREM *The flow vector \hat{f} obtained in Phase II of the above procedure is a maximum value feasible flow vector in G .*

PROOF Since \bar{f} is a feasible flow vector in G of value \bar{v} ; and \bar{g} is a feasible flow vector of value ω in $G(\bar{f})$, the fact that \hat{f} is a feasible flow vector of value $\bar{v} + \omega$ follows from the flow conservation equations satisfied by \bar{f} and \bar{g} in the respective networks G and $G(\bar{f})$, and the definition of upper bounds in $G(\bar{f})$.

Now, to show that \bar{f} is a maximum flow in G , suppose not. Then there must exist an FAP from \check{s} to \check{t} with respect to \hat{f} in G . Suppose it is \mathcal{P} . We will now show that using \mathcal{P} we can construct an FAP from \check{s} to \check{t} \mathcal{P}_1 , in $G(\bar{f})$ with respect to \bar{g} .

1. If (i, j) is a forward arc on \mathcal{P} with $\hat{f}_{ij} = \bar{f}_{ij}$, we have $\bar{f}_{ij} < \check{k}_{ij}$, so arc (i, j) exists in $G(\bar{f})$ with $+$ label and capacity $\check{k}_{ij} - \bar{f}_{ij} > 0$, and since $\hat{f}_{ij} = \bar{f}_{ij}$, we must have $\bar{g}_{ij} = 0$. So put (i, j) as a forward arc on \mathcal{P}_1 .
2. If (i, j) is a forward arc on \mathcal{P} with $\hat{f}_{ij} > \bar{f}_{ij}$, from the definition of \hat{f} , (i, j) must be a $+$ labeled arc in $G(\bar{f})$ with $\bar{g}_{ij} = \hat{f}_{ij} - \bar{f}_{ij} > 0$, and since $\hat{f}_{ij} = \bar{f}_{ij} + \bar{g}_{ij} < \check{k}_{ij}$, we have $\bar{g}_{ij} < \check{k}_{ij} - \bar{f}_{ij} = \check{\kappa}_{ij}$. So put (i, j) as a forward arc on \mathcal{P}_1 .
3. If (i, j) is a forward arc on \mathcal{P} with $\hat{f}_{ij} < \bar{f}_{ij}$, from the definition of \hat{f} , (j, i) must be a $-$ labeled arc in $G(\bar{f})$ with $\bar{g}_{ji} > 0$. Put (j, i) as a reverse arc on \mathcal{P}_1 .
4. If (i, j) is a reverse arc on \mathcal{P} with $\hat{f}_{ij} = \bar{f}_{ij}$, we have $\bar{f}_{ij} > \check{l}_{ij}$, so arc (j, i) must be a $-$ labeled arc in $G(\bar{f})$ with capacity $\check{\kappa}_{ji} = \bar{f}_{ij} - \check{l}_{ij} > 0$, and from the definition of \hat{f} , $\bar{g}_{ji} = 0$. Put (j, i) as a forward arc on \mathcal{P}_1 .
5. If (i, j) is a reverse arc on \mathcal{P} with $\hat{f}_{ij} > \bar{f}_{ij}$, from the definition of \hat{f} , (i, j) must be a $+$ labeled arc in $G(\bar{f})$ with $\bar{g}_{ij} = \hat{f}_{ij} - \bar{f}_{ij} > 0$. Put (i, j) as a reverse arc on \mathcal{P}_1 .

6. If (i, j) is a reverse arc on \mathcal{P} with $\hat{f}_{ij} < \bar{f}_{ij}$, from the definition of \hat{f} , (j, i) must be a $-$ labeled arc in $G(\bar{f})$ with $\bar{g}_{ji} = \bar{f}_{ij} - \hat{f}_{ij} > 0$; and since $\bar{f}_{ij} - \bar{g}_{ji} = \hat{f}_{ij} > \bar{\ell}_{ij}$, we have $\bar{g}_{ji} < \bar{f}_{ij} - \bar{\ell}_{ij} = \bar{\kappa}_{ji}$. Put (j, i) as a forward arc on \mathcal{P}_1 .

It can be verified that the path \mathcal{P}_1 constructed using statements 1 to 6 above is a path from \check{s} to \check{t} in $G(\bar{f})$, with the forward, reverse orientations for arcs on it as specified in these statements, and that it is an FAP from \check{s} to \check{t} in $G(\bar{f})$ with respect to \bar{g} . This contradicts the hypothesis that \bar{g} is a maximum value flow from \check{s} to \check{t} in $G(\bar{f})$. So there does not exist any FAP from \check{s} to \check{t} in G with respect to \hat{f} . Hence \hat{f} is a maximum value flow in G . ■

The theorem shows that the two phase procedure described here always finds a maximum flow in the given network G .

References

- [1] L. R. Ford Jr. and D. R. Fulkerson, 1956, "Maximum Flow through a Network", *Canadian Journal of Mathematics*, **8**(399–404).
- [2] A. V. Goldberg and R. E. Tarjan, 1986, "A New Approach to the Maximum Flow Problem", *Proceedings of the 18th Symposium on the Theory of Computing*, (136–146).
- [3] A. V. Goldberg and R. E. Tarjan, 1988, "A New Approach to the Maximum Flow Problem", *Journal of the Association for Computing Machinery*, **35**, no. 4(921–940).
- [4] K. G. Murty, 1992, *Network Programming*, to appear.