

Efficient Solution Approach to Maximum Flow Evacuation Planning Problem without Flow Conservation Aspect

Phanindra Prasad Bhandari, Shree Ram Khadka

Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal

Corresponding author: phanindra.maths@gmail.com

Received: April 1, 2017

Revised: May 22, 2017

Accepted: May 27, 2017

Abstract: An attempt of shifting as more people as possible and/or their logistics from a dangerous place to a safer place is an evacuation planning problem. Such problems modeled on network have been extensively studied and the various efficient solution procedures have been established. The solution strategies for these problems are based on source-sink path augmentation and the flow function satisfies the flow conservation at each intermediate node. Besides this, the network flow problem in which flow may not be conserved at node necessarily could also be used to model the evacuation planning problem. This paper proposes a model for maximum flow evacuation planning problem on a single-source-single-sink static network with integral arc capacities with holding capability of evacuees in the temporary shelter at intermediate nodes and extends the model into the dynamic case.

Keywords: Evacuation planning, network flow, maximum flow problem, preflow-push algorithm.

1. Introduction

Human casualties as well as massive destruction of properties due to natural and human created disasters have drawn the attention of researchers to find efficient emergency management procedures so that the casualties and the destructions could be reduced. Besides the disasters, efficient route planning becomes crucial in mass-meetings management and in mitigation of the traffic in a busy traffic hours. An evacuation planning problem asks to find an optimal evacuation plan in a realistic flow model where each evacuee is supposed to be evacuated in an efficient way. We take the network flow optimization approach to deal the evacuation planning problem. An evacuation network consists of the source (dangerous place), the sink (safe place), the transshipment nodes (intermediate junctions) and the arcs (path-segments between any two nodes) with some given capacities. Moreover, in the proposed modification, the intermediate nodes serve with temporary shelters with sufficient capacities. The source contains evacuees and the sink waits them with sufficient capacities. The movement of the evacuees in the path-segment is the flow in the arc.

The first maximum flow problem that obtains the maximum amount of flow from the source to the sink satisfying the flow conservation at every intermediate nodes was due to Ford and Fulkerson [6]. Their solution to the problem is based on arbitrary source-sink path augmentation on the residual networks and runs only in pseudo polynomial time. Edmonds and Karp [5] improved the running time of this algorithm to polynomial time by choosing a (one of the) shortest source-sink path in the residual network in each iteration for augmentation. The idea of augmenting all the shortest paths at once in each iteration in a layered sub-network of the residual network is given by Dinic [4]. This further improves the efficiency of the algorithm in [5].

The dynamic version of the maximum flow problem was first studied by the authors in [7, 8]. They showed that the general maximum dynamic flow problem is equivalent to the static flow problem in the time-expanded network and solved the problem in two-terminal network by temporarily repeating the static solution in the given network itself considering the transit times of the arcs as cost coefficients. Burkard et al. [2] studied the two-terminal quickest flow problem that asks to minimize the total time to send the given flow from the source to the sink and gave a polynomial time solution. Hoppe [11] studied the multi-terminal version of the problem known as quickest transshipment problem, minimizing the time taken to satisfy the given supplies and demands. Baumann [1] considered the earliest arrival flow problem to deal the evacuation planning problem that asks to maximize flow into the sink at each time step within the time horizon. The existence of a solution to this problem was proved by Gale [9]. Minieka [15] and Wilkinson [22] gave exact algorithms for the problem which needs an exponential time. Hoppe [11] reviews these algorithms and represents as chain decomposable flows. A fully polynomial time approximation scheme for this problem on a two-terminal network was presented by Hoppe and Tardos [12]. Steiner [21] and Ruzika et al. [20] presented a polynomial time algorithm for the earliest arrival flow problem on series-parallel networks.

Evacuation planning problems based on network contraflow approach that reconfigurates the network identifying the ideal direction and reallocating the available capacity for each arc to minimize the evacuation time from source to sink have also been extensively studied. Rebennack et al. [19] studied the maximum static contraflow problem for general networks, the maximum dynamic contraflow problem and that the quickest contraflow problem for two terminal networks and presented the polynomial time algorithms, based on network transformation but the quickest transshipment contraflow problem and fixed switching cost contraflow problems are shown NP-hard. Dhamala and Pyakurel [3] formulated a mathematical model of the earliest arrival contraflow problem and gave strongly polynomial time algorithm to solve it on a two-terminal series-parallel network. Pyakurel et al. [18] present a pseudo-polynomial time algorithm for the generalized maximum dynamic contraflow problem and the generalized earliest arrival contraflow problem for a two terminal lossy network. Pyakurel and Dhamala [16] considered both static and dynamic lexicographic contraflow problems that optimize the feasible flow leaving or entering the terminals in the given order after reversing the direction of arcs with polynomial time solutions. For the dynamic problem, they have considered

the constant travel time and node capacity but arc reversal capability is assumed at each integer time point, unlike in Dhamala and Pyakurel [3]. Khadka and Bhandari [14] have developed the model for evacuation planning problem with contraflow approach in continuous time setting by allowing the arc reversibility at time zero with strongly polynomial time algorithm. They also have generalized the problem that allows arc reversibility at any subinterval of the time horizon and presented a pseudo polynomial time algorithm. Pyakurel and Dhamala [17] have discussed different dynamic contraflow problems namely; maximum dynamic, earliest arrival, lexicmaximum dynamic, earliest arrival transshipment and quickest transshipment contraflows on particular networks and presented efficient algorithms to solve the problems in continuous time model.

The concept of preflow, pushing as much flow as possible out of the source and trying to get it to the sink without satisfying the flow conservation by the flow function at nodes necessarily, was first introduced by Karzanov [13]. Later, Goldberg and Tarjan [10] discussed the preflow concept as an efficient tool (at least not harder than any known path based algorithms) to solve the maximum flow problem in network and gave the preflow push algorithm (sometimes 'push relabel algorithm' in literature) to solve it. Preflow-push algorithm works on a node at a time and its neighbors. This repeatedly selects an active node and pushes its flow to neighbors which are close to sink. This approach, however, sends the excess flow (evacuees) that is blocked to send into sink back to the source and deletes from entire evacuation network. We propose a relaxation on this property to allow holding evacuees in the temporary shelter at intermediate nodes instead of sending them back to the source. This relaxation could be more advantageous in many ways. A detail is given in Section 4.

The paper is organized as follows. In section 2, we define a maximum evacuation planning problem with holding capability of evacuees in temporary shelter at intermediate nodes based on non-conserving flow model. The preflow-push algorithm has been presented in Section 3. A simple modification on the procedure to the problem has been proposed in Section 4. Section 5 concludes the paper.

2. Problem Formulation

We model an evacuation planning problem on a network $N = (V, E)$ where V ($|V| = n$) is for the set of the nodes, E ($|E| = m$) for the set of the arcs joining any two nodes such that $m \geq n - 1$. We assign some positive capacity c_e for each arc $e = (v, w) \in E$. Unsafe place where accident has occurred or going to be occurred soon is taken as the source s and the safe place where the people are to be evacuated is assumed as the sink d . The source and the sink are terminal nodes on N . The source node contains evacuees, and the sink node waits them for shelter with enough space. We assume the temporary shelter v' at every node $v \in V - \{s, d\}$ with sufficient capacity of $c_{v'}$. We compare the movement of evacuees on the road with flow on the network N . See Example 1 for detail.

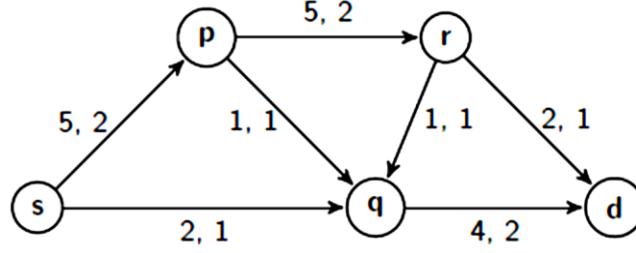


Fig. 1: An evacuation network. First and second numbers next to the arc represent the initial arc capacity and the transit time respectively.

A preflow is an assignment $f: E \rightarrow R^+ \cup \{0\}$ satisfying the following constraints.

The capacity constraints as

$$0 \leq f(e) \leq c_e \quad \forall e \in E, \tag{1}$$

the skew symmetry of the flow in arc as

$$f(v, w) = -f(w, v) \quad \forall v, w \in V \tag{2}$$

and the weakened form of the flow conservation constraints known as the non-negativity constraints in the form

$$\sum_{w \in V} f(w, v) - \sum_{w \in V} f(v, w) \geq 0 \quad \forall v \in V - \{s, d\}. \tag{3}$$

We consider a non-negative excess at node v defined as

$$e_v = \sum_{w \in V} f(w, v) \tag{4}$$

and call the node v an active node if $e_v > 0$. We define the nonnegative overload at node v as

$$o_v = \sum_{w \in V} f(w, v) e_v - \sum_{w \in V} f(v, w) \geq 0 \quad \forall v \in V - \{s, d\} \tag{5}$$

$$\text{Satisfying } o_v \leq c_v \quad \forall v \in V - \{s, d\}. \tag{6}$$

The maximum static $s - d$ flow problem in a network can be defined as

$$\text{maximize } f_s = \sum_{v \in V} f(v, d) \tag{7}$$

satisfying the constraints (1) – (3) and allowing to hold o_v at $v' \quad \forall v \in V$.

For a dynamic network $N = (V, E, c_e, \tau_e, c_{v'}, s, d, T)$ with the time horizon T and the travel time τ_e along the arc $e, e \in E$ we define preflow f as $f: E \times \{0, 1, \dots, T\} \rightarrow R^+ \cup \{0\}$ which satisfies the capacity constraints as

$$f(e, \theta) \leq c_e \quad \forall \theta \in \{0, 1, \dots, T\}, \quad (8)$$

the skew symmetry of the flow at each time step $\theta \in \{0, 1, \dots, T\}$ along the arc as

$$f(v, w) = -f(w, v) \quad \forall v, w \in V \quad (9)$$

and the constraint with no flow conservation as

$$\sum_{e \in \delta^+(v)} \sum_{\theta=0}^T f(e, \theta) \geq \sum_{e \in \delta^-(v)} \sum_{\theta=0}^T f(e, \theta). \quad (10)$$

The objective of the problem in dynamic case is to maximize the dynamic flow

$$f_d = \sum_{e \in \delta^+(d)} \sum_{\theta=0}^T f(e, \theta). \quad (11)$$

The overload flow at every node $v \neq s, d$ is allowed to hold at the temporary shelter v' of the node v for each time step $\theta \in \{0, 1, \dots, T\}$.

Example 1. Consider a two terminal evacuation network as depicted in Fig 1. Each oval (a node) represents crossing of two or more routes and each straight line with an arrow (an arc) a path-segment of the route that joins two nodes. Node s is the dangerous place (source), d is the safe place (sink) and the remaining are the intermediate nodes. The source contains evacuees and the sink waits them with sufficient capacity. The movement of the evacuees (possibly cars) on the path segment is the flow on the arc. The first number next to each arc is the arc capacity and the second number is the time needed for a flow unit to travel the arc. For example, an arc between nodes p and r has capacity 5 and transit time 2. That is, 5 cars can pass simultaneously through arc (p, r) within 2 hours.

3. The Preflow-Push Algorithm

Unlike the ordinary flow, the preflow does not necessarily maintains the flow conservation at every node in the network. Starting with a preflow, the preflow-push algorithm in [10] pushes the excess flow closer towards the sink and the overload flow at any node is sent back to the source. Eventually, preflow becomes a flow and in fact the maximum $s - d$ flow.

The preflow-push is an iterative algorithm that updates the preflow in the residual network in each iteration. For the network $N = (V, E, c_e)$, a residual network $N_f = (V, E_f, c_e)$ is constructed with respect to the preflow f . For an arc $e = (v, w)$ with capacity c_e and carrying

preflow $f(e)$, N_f includes arcs in forward direction of arc e with residual capacity $r_e = c_e - f(e)$ and arcs in reverse direction of arc e with residual capacity $r_e = f(e)$. Arcs with zero capacity are omitted from N_f . The algorithm starts with the initialization of the given network N . At the beginning, the preflow f on each arc leaving the source is equal to the arc capacity of the respective arcs and zero on the remaining arcs. And all the intermediate nodes will be assigned an initial label 0 that can then be increased through the relabel operation. A function $l: V \rightarrow N$ is a label function satisfying $l(s) = n (= |V|)$, $l(d) = 0$ and $l(v) \leq l(w) + 1$ for every residual arc $e = (v, w) \in E_f$. The labels of the source and the sink will remain unaltered throughout the algorithm and the labels for remaining nodes increases up to $2n - 1$. Whenever the algorithm finds an active node v (i.e. $e_v > 0$) and if there is an arc in N_f from node v to node w such that $l(v) = l(w) + 1$, then v pushes some excess less or equal to $c_{e=(v,w)}$ from v to w . If $l(w) \geq l(v) \forall w \in V$, label of v is to be increased. We increase the label of an active node v as $l(v) := 1 + \min\{l(w) : (v, w) \in N_f\}$. If node v satisfies $0 \leq l(v) < n$ then there is possibility of getting paths from v to d . If $l(v) \geq l(s)$, the sink is not reachable from v in N_f and starts to send the flow back towards the source. The algorithm repeatedly performs these operations whenever applicable. At the termination, the algorithm gives a preflow with $e_v = 0 \forall v \in V - \{s, d\}$. That is, a valid ordinary flow in the network. This flow is the maximum flow in the original network N .

Now we present the preflow-push algorithm of Goldberg and Tarjan [10].

Algorithm 1. Preflow-Push Algorithm

Input: Network $N = (V, E, c_e, s, d)$

Initialize: $l(s) := n = |V|$; $l(v) := 0 \forall v \in V - \{s\}$.

For each $(s, v) \in E$, $f(s, v) := c_{(s,v)}$ and $f(e) := 0$ for all other arcs $e \in E$.

Push Operation: For arc $(v, w) \in E_f$ of active node $v \neq s, d$ with $l(v) = l(w) + 1$, push $\delta = \min\{e_v, c_{(v,w)}\}$ units of flow along (v, w) . Otherwise,

Relabel Operation: For an active node v , set $l(v) := 1 + \min\{l(w) : (v, w) \in N_f\}$.

Output: Maximum $s - d$ flow.

Example 2. Consider the evacuation network of Fig 1 without the transit times on it. Start the procedure by initializing as $l(s) = 5, l(v) = 0 \forall v \in \{p, q, r, d\}$; $f(s, p) = 5, f(s, q) = 2$ and $f(e) = 0$ for remaining arcs e . This makes p and q active with $e_p = 5$ and $e_q = 1$. For active node p , relabel it to be $l(p) = 1 + \min\{l(q), l(r)\} = 1$ since $l(p) \neq 1 + l(q)$ and $l(p) \neq 1 + l(r)$. Now, push $\delta = \min\{e_p, c_{(p,r)}\} = 5$ units of flow along (p, r) , since $l(p) = 1 + l(r)$ holds. This makes e_r active with $e_r = 5$. Relabel r to be $l(r) = 1$. Now, push $\delta = \min\{e_r, c_{(r,d)}\} = 2$ unit of flow along (r, d) since $l(r) = 1 + l(d)$ holds. Also, we push $\delta = \min\{e_r, c_{(r,q)}\} = 1$ unit along (r, q) since $l(r) = 1 + l(q)$ holds. This makes $e_q = 3, e_p = 0, e_r = 2, e_d = 2$. Relabel the active node q to be $l(q) = 1$ since $l(q) \neq 1 + l(d)$. Now,

push $\delta = 3$ units of flow along (q, d) since $l(q) = 1 + l(d)$ holds. The only active node now is r with $e_r = 2$ and relabel it to be $l(r) = 2$ since $l(r) \neq 1 + l(p)$. Now, push $\delta = 2$ units of flow along (r, p) . This makes p an active with $e_p = 2$. Relabel p as $l(p) = 1 + l(q) = 2$. Now, $l(p) = 1 + l(q)$ holds. Push $\delta = 1$ unit of flow along (p, q) and then $\delta = 1$ unit of flow along (q, d) . Now, p only is an active node. Relabel p as $l(p) = 3$. Now, $l(p) = 1 + l(r)$ holds. Push $\delta = 1$ unit of flow along (p, r) . Now, only node r is an active. Relabel r as $l(r) = 1 + l(p) = 4$. Push $\delta = 1$ unit along (r, p) . Now, only p is active node. Relabel p as $l(p) = 5$. Now, $l(p) = 1 + l(r)$ holds. Push $\delta = 1$ unit of flow along (p, r) . Again, only r is an active node. Relabel r to be $l(r) = 6$, and push flow to p . Relabel p to be $l(p) = 6$, so that $l(p) = 1 + l(s)$ holds and flow is pushed to source s . This makes all intermediate nodes empty and $e_d = 6$. The procedure terminates with maximal $s - d$ flow of value 6.

4. Modification of the Preflow-Push Algorithm

It is not wise always (at the time of evacuation for example) to force sending the overload flow (evacuees) to the source but better to hold in temporary shelter at intermediate node. This could be advantageous in many ways; for instance, to provide immediate medication to highly injured evacuees by keeping in temporary shelter, sending evacuees with high priority to sink and leaving the evacuees with less priority. More importantly, those evacuees which cannot be brought into sink due to some arc capacity limitation can be kept in temporary shelters at intermediate nodes which could be safer as compared to source in many evacuation scenarios. During evacuation people may loss their lives on the way. This circumstance can also be addressed by this model which is clearly designed for non-conservation of the flow at intermediate nodes.

We propose a modification on the algorithm of Goldberg and Tarjan [10] that solves the maximum evacuation planning problem with holding capability of flow at temporary shelter in intermediate nodes. We consider the temporary shelter v' for each intermediate node $v \in V$ with sufficient capacity to make our model simpler. However, it does not exceed the sum of the capacities of outgoing arcs of the source minus the maximum flow. The modified procedure does not send the evacuees reaching once at intermediate nodes back again to the source, a dangerous place but pushes back to any intermediate nodes. For an active node v , we hold the overload flow o_v in the temporary shelter v' at node v if the push operation and the relabel operation are not applicable and even if $l(v) \leq l(s)$ does not satisfy after relabeling it. The modified preflow-push algorithm gives the maximum flow at its termination since the flow which we send back to the source in the preflow-push algorithm of Goldberg and Tarjan [10] is kept at temporary shelter and can be treated as if this is deleted from entire network. At the meantime, the max-flow-min-cut theorem of Ford and Fulkerson [6] guarantees that the final flow is the maximal flow. Moreover, Maximum flow obtained from the modified algorithm exactly meets the maximum flow obtained by the algorithm of Ford and Fulkerson [6].

Time plays an important role in evacuation planning problem. The method we proposed for static case can be used to solve a dynamic version of the problem with necessary modifications on the procedure. The investigation of Ford and Fulkerson [7], which states that a feasible flow on N is equivalent feasible flow of the problem on the corresponding time expanded network, assures that the problem can be solved by converting the dynamic network N into the time expanded network N^T over the time horizon T . The time expanded network N^T is defined as $N^T = (V^T, E^T, c_e, c_{v'}, \tau_e, T)$, where

$$V^T = \{v(\theta) : v \in V \text{ and } \theta \in \{0, 1, \dots, T\}\}$$

and

$$E^T = \{(v(\theta), w(\theta + \tau_{(v,w)})) : v \neq w, v, w \in V \text{ and } \theta \in \{0, 1, \dots, T - \tau_e\}\}.$$

We connect each $s(\theta)$ and $d(\theta)$; $\theta \in \{0, 1, \dots, T\}$ by a super source S and a super sink D , respectively with capacities $c_{(S,s(\theta))} := \sum_{e=(s,v) \in E} c_e$ and $c_{(d(\theta),D)} := \sum_{e=(v,d) \in E} c_e \quad \forall \theta \in \{0, 1, \dots, T\}$ so that the dynamic evacuation planning problem can be solved on the underlying time expanded (static) network using the procedure we proposed.

5. Concluding Remark

This paper highlights the maximum flow problem with non-conserving flow model introduced by Goldberg and Tarjan [10] and shows the importance of holding the flow that arrives at intermediate node and is blocked for being sent into the sink instead of sending them back to the source. We gave the mathematical model for the problem and discussed its importance in evacuation planning. The current paper models the evacuation planning problem with holding capability of evacuees at intermediate nodes and proposes the solution procedure in static case as well as in dynamic case with the aid of time expanded network. Exact solution procedure for the problem in static case and more efficient solution procedure for dynamic version of the problem would be the future research works related to the topic.

Acknowledgements: The first author thanks to University Grants Commission, Nepal for awarding by PhD Fellowship 2016 and the DAAD support to the partnership program between TU Kaiserslautern, Germany; Tribhuvan University, Nepal and Mindanao State University, IIT-Iligan, Philippines.

References

- [1] Baumann N (2007), Evacuation by earliest arrival flows. PhD Thesis, Department of Mathematics, University of Dortmund, Germany.
- [2] Burkard R, Dlaska K and Klinz B (1993), The quickest flow problem. *Mathematical Methods of Operations Research*, **37(1)**: 31-58.
- [3] Dhamala TN and Pyakurel U (2013), Earliest arrival contraflow problem on series parallel graphs. *International Journal of Operations Research*, **10(1)**: 1-13.

- [4] Dinic EA (1970), Algorithm for solution of a problem of maximum flow in networks with power estimation. *Sov. Math. Dokl.*, **11**: 1277-1280.
- [5] Edmonds J and Karp RM (1972), Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, **19(2)**: 248-264.
- [6] Ford LR and Fulkerson DR (1956), Maximal flow through a network. *Can. J. Math.*, **8**: 399-404.
- [7] Ford LR and Fulkerson DR (1958), Constructing maximal dynamic flows from static networks. *Operations Research*, **6(3)**: 419-433.
- [8] Ford LR and Fulkerson DR (1962), *Flows in Networks*. Princeton University Press.
- [9] Gale D (1959), Transient flows in networks. *Michigan Mathematical Journal*, **6**: 59-63.
- [10] Goldberg AV and Tarjan RE (1988), A new approach to the maximum-flow problem. *Journal of ACM*, **35(4)**: 921-940.
- [11] Hoppe B (1995), Efficient dynamic network flow algorithms. PhD Thesis, Cornell University.
- [12] Hoppe B and Tardos E (1994), Polynomial time algorithms for some evacuation problems. *Procc. of 5th Ann. ACM-SIAM Symp. on Discrete Algorithms*, 433-441.
- [13] Karzanov AV (1974), Determining a maximal flow in a network by the method of preflows. *Doklady Akademii Nauk SSSR*, **215**: 49-52, in Russian. (English translation in *Soviet Math. Dokl.*, **15(2)**: 434-437).
- [14] Khadka SR and Bhandari PP (2017), Dynamic network contraflow evacuation planning problem with continuous time approach. *International Journal of Operations Research*, **14(1)**: 27-34.
- [15] Minieka E (1973), Maximal lexicographic and dynamic network flows. *Operations Research*, **21**: 517-527.
- [16] Pyakurel U and Dhamala TN (2015), Models and algorithm on contraflow evacuation planning network problems. *International Journal of Operations Research*, **12(2)**: 36-46.
- [17] Pyakurel U and Dhamala TN (2016), Continuous dynamic contraflow approach for evacuation planning. *annals of operations research*, 1-26, DOI: 10.1007/s10479-016-2302-5.
- [18] Pyakurel U, Hamacher HW and Dhamala TN (2014), Generalized maximum dynamic contraflow in lossy network. *International Journal of Operations Research Nepal*, **3**: 27-44.
- [19] Rebennack S, Arulselvan A, Elefteriadou L and Pardalos PM (2010), Complexity analysis of maximum flow problem with arc reversal. *Journal of Combinatorial Optimization*, **19**: 200-216.
- [20] Ruzika S, Sperber H and Steiner M (2011), Earliest arrival flows on series-parallel graphs. *Networks*, **10**: 169-173.
- [21] Steiner M (2009), A Survey on Earliest Arrival Flows and a Study of the Series-Parallel Case. Diploma Thesis, Department of Mathematics, University of Kaiserslautern, Germany.
- [22] Wilkinson WL (1971), An algorithm for universal maximal dynamic flows in a network. *Operations Research*, **19**: 1602-1612.