



# Maximum Flow Evacuation Planning Problem with Non-Conservation Flow Constraint

Phanindra Prasad Bhandari<sup>1\*</sup>, Shree Ram Khadka<sup>2</sup>

<sup>1,2</sup>Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal

<sup>1</sup>Department of Science and Humanities, Khwopa Engineering College, Bhaktapur, Nepal

\* Corresponding author email: [phanindra.maths@gmail.com](mailto:phanindra.maths@gmail.com)

Received: 13 June 2020 / Revised: 02 August 2020 / Accepted: 20 August 2020 / Published: 04 September 2020

## ABSTRACT

The optimization model of the maximum flow evacuation planning problem efficiently sends a maximum number of evacuees along with the routes of their transshipment from the disastrous zone, the source, to the safe zone, the sink, over a given time horizon. The limitation of the problem with the flow conservation constraint at the intermediate nodes is that even one more evacuee cannot be sent out from the source, if the evacuee cannot reach the sink. However, evacuees must attempt to send out as many evacuees as possible to safer places despite the sink. There may be relatively safe places in between the source and the sink. The limitation is due to the flow conservation constraint. In this paper, we remodel the problem with non-conservation flow constraint and propose an efficient algorithm. With this approach one can send as many evacuees as in the flow conservation case from the source to the sink. Moreover, a maximum number of evacuees can also be sent to the relatively safe places in between the source and the sink. The routes of their transshipment can also be identified.

**Keywords:** Evacuation Planning Problem, Pre-flow-push Algorithm, Network Flow, Disaster Management.

## 1 Introduction

Optimal use of the road network is required to implement efficient evacuation planning over a region in order to send a maximum number of evacuees from the disastrous zone, the source, to the safe zone, the sink, over a given time horizon [1]. The maximum flow evacuation planning problem efficiently sends a maximum number of evacuees from the source to the sink via various intermediate road segments through the road crossings as soon as possible during the evacuation or mitigation of the traffic in a rush hour in the crowded urban area. The motivation of researchers has been increased to study the problem so that the loss of people and their property could be minimized and that the traffic could be efficiently mitigated even in the crowded urban area during the rush hour.

The maximum flow problem on a single-source-single-sink static network, the network in

which only the constant arc capacity is issued in each arc, has been studied in [2] and that on dynamic network, the network in which the constant arc capacity and transit time, i.e., the time a unit of flow takes to reach the neighboring node, has been studied in [3, 4]. A pseudo-polynomial algorithm based on the time-expanded network and a polynomial one based on the temporarily repeated flow with transit time on the arc as a cost coefficient have been investigated to solve the dynamic case of the problem. The solution procedure of [2] has been improved in [5] by choosing the shortest paths from the source to the sink in the residual network and in [6] augmenting all the shortest paths at once in each iteration in a layered sub-network of the residual network. Authors in [7] came up with an efficient procedure, known as push-relabel approach, for solving maximum flow problem that works on a node and its adjacent arcs at a time. From the 1950s to 2013,



when Orlin [8] proved any instance of the problem can be solved in  $O(nm)$  time, various algorithms have been devised for solving this problem. A brief survey of the most important algorithms and their running time bounds is provided in [9]. The other variants of the evacuation planning problem have also been widely studied, [10–12].

The quickest flow problem, closely related to maximum dynamic flow problem, which minimizes the total time to send a given number of evacuees from a single source through a network to a single sink has been investigated in [13]. The algorithm can be executed in a polynomial time. Hoppe [14] extended the problem on a network with multiple sources and sinks. Another variant of the evacuation planning problem which sends a maximum evacuee at every step of time over time horizon is called the earliest arrival flow problem [15]. There exist exact algorithms with exponential time [16, 17]. A fully polynomial time approximation for the problem with a single-source-single-sink case has been investigated in [18], with multiple-sources-single-sink in [19–21] and that on series-parallel network in [22, 23]. A polynomial time algorithm for the problem with time dependent transit times and capacities is proposed in [24].

The evacuation planning problem with contraflow (arc reversal on network) approach has also been studied with a number of efficient solution procedures. The approach increases the outbound capacity of the arc and decreases the evacuation time. The maximum flow on a general static network, the maximum flow and the quickest flow cases on the dynamic network with single-source-single-sink have been investigated in [25] and the earliest arrival flow on a single-source-single-sink series-parallel network in [26] with efficient algorithms. The generalized maximum dynamic contraflow and generalized earliest arrival contraflow problems have been introduced and proposed pseudo-polynomial time algorithm for a single-source-single-sink lossy-network in [27]. The static lexicographic contraflow problem and the dynamic lexicographic contraflow problem with the constant travel time and node capacity that optimizes the feasible flow leaving or entering the

terminals in the given order with polynomial time solutions have been investigated in [28].

The evacuation planning problem in continuous time setting has been introduced in [29] and extensively studied in [30–34] with a number of efficient algorithms. A strongly polynomial time algorithm to the maximum dynamic flow problem with contraflow approach in continuous time setting has been investigated in [35] with the arc reversal at time zero. Authors [36] discussed a number of dynamic contraflow problems in continuous time model with efficient algorithms. For deeper insight about evacuation planning problems, we refer to the survey articles [37, 38].

The existing optimization evacuation planning models with flow conservation constraint at the intermediate nodes has a limitation that even one more evacuee cannot be sent out from the source, if the evacuee cannot reach the sink. However, evacuators must attempt to send out as many evacuees as possible to safer places (despite the sink) in between the source and the sink where the evacuators can provide medical aids or other necessary supports to the evacuees or put the priorities among the evacuees to send to the sink. In this paper, we consider the maximum flow evacuation planning problem incorporating this aspect. We propose an efficient polynomial time solution procedure to solve the problem for static network and extend the solution procedure with a pseudo-polynomial time algorithm for single-source-single-sink dynamic network. Model of the problem and preliminary results without analytical proofs have been given in [39] also.

The paper is organized as follows. Mathematical formulation of the problem is described in Section 2. Section 3 contains solution to the problem with solution over static network in Subsection 3.1, on dynamic network in Subsection 3.2 and with priority based intermediate nodes in Subsection 3.3. An example is presented to implement the solution idea developed in this paper in Subsection 3.4. The last section concludes the paper.

## 2 Problem Formulation

Consider a single-source-single-sink dynamic network  $N = (V, E)$  where  $V$  and  $E$  are set of nodes and arcs, respectively, with  $|V| = n$ ,  $|E| = m$ ;  $n - 1 \leq m$ . The objective function of the maximum evacuation planning problem on  $N$  is to maximize

$$f = \sum_{e \in \delta^-(d)} \sum_{\theta=0}^T f(e, \theta),$$

the total flow units of evacuees, over the given time horizon  $T$ , sent from the disastrous zone  $s$ , the source, to the safe zone  $d$ , the sink, through the possible routes reachable to the sink. The set of route segments  $e$ , the arc, entering into the sink  $d$  is denoted by  $\delta^-(d)$ . The integer flow unit  $f(e, \theta)$ , that flows at time step  $\theta \in \{0, 1, \dots, T\}$  along the arc  $e = (v, w)$  between the two adjacent intersections, the nodes,  $v$  to  $w$  such that  $v, w \in V$ , the set of intersections, follows the following constraints.

The flow units cannot exceed the arc capacity  $c_e$  for any time step within time horizon  $T$ , i.e.,

$$0 \leq f(e, \theta) \leq c_e \\ \forall e \in E \text{ and } \forall \theta \in \{0, 1, \dots, T\}. \quad (1)$$

The flow units that enter into a node  $v$  for each time step may not exit from it at the same time or later within time horizon  $T$ , i.e.,

$$\sum_{e \in \delta^-(v)} \sum_{\theta=0}^{T-\tau_e} f(e, \theta) - \sum_{e \in \delta^+(v)} \sum_{\theta=0}^T f(e, \theta) \geq 0, \\ \forall v \in V \setminus \{s, d\}. \quad (2)$$

where  $\delta^-(v)$  and  $\delta^+(v)$  denote for the set of arcs entering into the node  $v$  and leaving from it, respectively, and  $\tau_e$  is the travel time for a flow unit along the arc  $e \in E$ .

The flow units remained at node  $v$  at time  $\theta \in \{0, 1, \dots, T\}$ , say the excess flow units  $e_v(\theta)$ , satisfies

$$e_v(\theta) \leq \sum_{e \in \delta^-(v)} f(e, \theta) \quad \forall \theta \in \{0, 1, \dots, T\}. \quad (3)$$

Such a node  $v \in V \setminus \{s, d\}$  at which  $e_v(\theta) > 0$  is said to be an active node at time  $\theta$  and the corresponding flow  $f(e, \theta)$  is a pre-flow.

## 3 Solution Procedure

### 3.1 Solution to the problem described on static network

Let us consider the problem described on the network  $N = (V, E, c_e, s, d)$  with no transit time on its arcs. The solution procedure of the problem is based on the pre-flow-push algorithm investigated by Goldberg and Tarjan [7] with necessary modification. The algorithm works on a residual network  $N_f = (V, E_f)$  of the network  $N$  with respect to the pre-flow  $f$ . If an arc  $(v, w) \in E$  and  $f(v, w) < c_{(v,w)}$ , then  $E_f$  contains the arc  $(v, w)$  with residual capacity  $r_{(v,w)} = c_{(v,w)} - f(v, w)$  and, if  $(v, w) \in E$  and  $f(v, w) > 0$ , then  $E_f$  contains the arc  $(w, v)$  with residual capacity  $r_{(w,v)} = f(v, w)$ . The pre-flow  $f(e)$  for the former and the latter case are said to be forward pre-flow and backward pre-flow, respectively.

A label function  $l: V \rightarrow Z^+ \cup \{0\}$  is defined as

$$l(v) := \begin{cases} = |V| & \text{if } v = s, \\ l(w) + 1 & \text{if } (v, w) \in N_f, \\ = 0 & \text{if } v = d. \end{cases}$$

If  $l(v) = l(w) + 1$  with  $e = (v, w) \in N_f$ ,  $\min\{e_v, c_e\}$  units of pre-flow are pushed from  $v$  to  $w$ . The node  $v$  with  $e_v > 0$  is relabeled to be  $l(v) := 1 + \min\{l(w) : (v, w) \in N_f\}$  when  $l(v) \leq l(w)$  such that  $l(v) \leq l(s)$ . The initial label to the node  $v \in V$  is assigned to be the shortest path distance of it from the sink  $d$  with the assumption of unit distance in each arc.

The pre-flow-push algorithm works on a node and its neighbors at a time and repeatedly selects an active node and pushes the flow to neighbors which are closer to the sink. The algorithm described in [7] sends the excess, the flow sent out from the source but is unable to reach to the sink, back to the source. If  $l(v)$  is at least as large as  $|V|$ , the algorithm reaches a stage at which the excess flow at  $v$  must be pushed back towards the source. However, this seems to be not fair during evacuation rather to keep such flow units of evacuees at a temporary shelter, say  $v'$ , at the intermediate node  $v \in V$ . This modifies the existing algorithm and becomes applicable to the evacuation planning problem.

This approach can be advantageous to provide immediate medication to severely injured evacuees, to keep the evacuees who lost their lives on the way at the temporary shelters and to make priorities among the evacuees to send to the sink. The procedure follows the operations as given in Algorithm 1.

---

**Algorithm 1:** Subroutine *PUSH-RELABEL-HOLD*( $v$ )

**Push:** For an arc  $(v, w) \in N_f$  of active node  $v$  with  $l(v) = l(w) + 1$ , push  $\delta =$

$\min \{e_v, c_{(v,w)}\}$  flow units along  $(v, w)$ .

Otherwise,

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

Otherwise,

**Hold:** Hold excess  $e_v > 0$  at  $v'$ .

---

Let us consider a static network  $N = (V, E, c_e, s, d)$ . The solution procedure (cf. Algorithm 2) to compute a maximum flow from the source node  $s$  to the sink node  $d$  is as follows.

---

**Algorithm 2:** *Modified Pre-flow-Push Algorithm for Static Network*

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

**Label Initialization:** For all  $v \in V \setminus \{s\}$ , set  $l(v)$  to be the shortest path distance of  $v$  from  $d$  and set  $l(s) := n$ .

**Pre-flow Initialization:** Set  $f(e) := c_e$  for all  $e \in \delta^+(s)$  and  $f(e) := 0$  for remaining arcs.

**Subroutine Application:** Apply subroutine *PUSH-RELABEL-HOLD*( $v$ ) for each active node  $v \in V$ .

**Output:** Maximum static flow  $f$  with intermediate hold on  $N$ .

---

The maximum flow is decomposed into  $s - d$  chains and that the excess flow, which is held at intermediate node  $v$ , is decomposed into  $s - v$  chains for each  $v \in V$ . One could notice that no cycle of the flow is produced in each iteration on the residual network. The algorithm 2 yields optimal solution in strongly polynomial time.

**Lemma 1.** Let  $v \in V$  be an active node and  $s \in V$  be the source node. There exists no pre-flow which could be pushed to  $s$ .

*Proof:* Let the label at a node  $v \in V$  be  $l(v)$ . A pre-flow  $f$  can be pushed from an active node  $v$  to its neighboring node  $w$  with  $v, w \in V$  only if  $l(v) = 1 + l(w)$ . The Modified Pre-flow-Push algorithm always maintains the label  $l(d) < l(s)$  at the sink  $d$  and  $l(v) \leq l(s)$  at any intermediate node  $v$ .

**Lemma 2.** Let  $v \in V$  be an active node. One of the operations, push or relabel or hold, is applicable to the node  $v$ .

*Proof:* Let  $l$  and  $f$  be any valid labeling and a pre-flow at a node in the residual network  $N_f$ , respectively. For an active node  $v \in V$ , there exists an arc  $(v, w) \in N_f$  such that  $l(v) \leq l(w) + 1$ . The node  $v$  is relabeled if  $l(v) < l(w) + 1$  and a pre-flow is pushed to the node  $w$  if  $l(v) = l(w) + 1$ . If the pre-flow cannot be pushed elsewhere then it is held at the temporary shelter  $v'$  of the node  $v$ .

**Theorem 1.** The Modified Pre-flow-Push Algorithm yields a maximum flow on network  $N$ .

*Proof:* At an active node  $v \in V$ , either there is a push of a pre-flow to a neighboring node or the node is relabeled or the excess is held at the temporary shelter  $v'$ . The pre-flow from  $s$  to  $d$  is feasible. If a pre-flow can neither be pushed nor the node be relabeled, the excess is held at the shelter  $v'$  of the node  $v$ . This makes the node  $v$  to be inactive. The algorithm is executed when there remains no active node in the residual network and pre-flow becomes a flow. The algorithm is terminated when there does not exist any  $s - d$  chain in  $N_f$ . Moreover, a flow is maximum if and only if there is no chain from  $s$  to  $d$  in  $N_f$ , [2].

**Theorem 2.** The subroutine *PUSH-RELABEL-HOLD* runs in strongly polynomial time.

*Proof:* Consider a network  $N$  with  $n$  nodes and  $m$  arcs such that  $n - 1 \leq m$ . The number of relabeling operations is at most  $n$  for each  $n - 2$  nodes. That is, relabeling can be done in  $O(n^2)$  times overall. Number of saturating pushes of pre-flow is at most  $n$  per arc for at most  $m$  arcs and number of non-saturating pushes is at most  $n^2m + n(n - 2)$  over the network, [7]. Thus, the push operations can be accomplished in  $O(n^2m)$  times, a dominating bound of the subroutine. The hold operations do not exceed the relabeling iterations.

Furthermore, the push operation can be improved with time  $O(n^2m)$  by using the first-in, first-out algorithm or the maximum-distance method described in [7]. The dynamic tree data structure modifies the algorithm with time  $O(nm \log(n^2/m))$ , [40].

Computation of distance in Algorithm 2 can be accomplished in time  $O(m+n)$  by using the breadth first search method on the inverse network. The maximum and the excess flow can be decomposed into chains in  $O(mn)$  time, [41]. This is an additional effort to implement the algorithm for the maximum flow evacuation planning problem. Hence one can observe that the algorithm is strongly polynomial.

### 3.2 Solution to the problem described on dynamic network

The solution procedure for the static network can also be applied to the dynamic case of the problem described on the dynamic network  $N = (V, E, c_e, \tau_e, s, d, T)$ . The time-expanded network  $N^T$ , suggested in [3], of network  $N$  for time horizon  $T$  without holdover arcs is given by  $N^T = (V^T, E^T, c_e, s, d)$  where  $V^T = \{v(\theta) : v \in V, \theta \in \{0, 1, \dots, T\}\}$  and  $E^T = \left\{ \left( v(\theta), w(\theta + \tau_{(v,w)}) \right) : v \neq w, v, w \in V \wedge \theta \in \{0, 1, \dots, T - \tau_{(v,w)}\} \right\}$ . For the modification, the sets  $\{(s(\theta), s(\theta + 1)) : \theta \in \{0, 1, \dots, T - 1\}\}$  and  $\{(d(\theta), d(\theta + 1)) : \theta \in \{0, 1, \dots, T - 1\}\}$  of arcs with sufficient capacities are added to the set  $E^T$ . Moreover, each node  $v(T); v \in V \setminus \{s\}$  is connected to  $d(T)$  by an artificial arc with zero capacity. The exact solution procedure that solves the maximum dynamic flow problem is given in Algorithm 3.

---

#### Algorithm 3: Modified Pre-flow-Push Algorithm for Dynamic Network

1. Given network  $N = (V, E, c_e, \tau_e, s, d, T)$ .
  2. Find time expanded network  $N^T$  of  $N$ .
  3. Apply "Modified Pre-flow-Push Algorithm for Static Network" on  $N^T$  where  $s(0)$  is the source and  $d(T)$  is the sink.
  4. Get maximum dynamic flow with intermediate hold on  $N$ .
- 

The solution on the dynamic network can be obtained in a polynomial time with respect to the input size. In particular, the relabel operation, the push operation and the hold operation can be executed in  $(T^2n^2)$ ,  $O(T^3n^2m)$  and  $O(T^2n^2)$  times, respectively which are pseudo-polynomial time complexities.

### 3.3 Solution to the problem with priority based intermediate nodes

The intermediate nodes, in which the excess flow of evacuees might be held, may not be of equal importance with respect to the risk, the distance from the sink, the holding capacity, etc. The intermediate nodes on the residual network  $N_f$  after computing the maximum static flow  $f$  be ordered as  $I_1, I_2, \dots, I_{(n-2)}$  from lower priority to higher one.

We consider a residual sub-network  $N_1$  obtained from  $N_f$  by deleting the source node  $s$ , the sink node  $d$  and the arcs leaving from  $s$  and that entering into  $d$ . The previous labels on the nodes of  $N_1$  are removed and the excess flow at  $I_1$  is pushed to  $I_i; i = (n-2), (n-3), \dots, 2$ , using Algorithm 2 after slight modification. For, we push the flow from the new source (i.e.,  $I_1$ ) that equals to the minimum of excess at  $I_1$  and the sum of the capacities of the arcs leaving it. The procedure is repeated on  $N_2$ , the residual sub-network obtained from  $N_1$  by deleting the node  $I_1$  and the arcs leaving from it and so on until the maximum flow from  $I_{(n-3)}$  to  $I_{(n-2)}$  is procured. The algorithm is executed  $(n-3), (n-4), \dots, 2, 1$  times on the residual sub-networks  $N_1, N_2, \dots, N_{(n-3)}$ , respectively for priority-based holding purpose. Hence the procedure works within  $O(n^4m)$  time.

### 3.4 Implementation of the algorithm

Let us consider the network depicted in Figure 1 for a region with 28 intersections as nodes and 83 road segments as arcs with the source node  $S$  and the sink node  $D$  and with specified road capacity and the transit time, for example, the arc capacity and the transit time for the arc from the node  $S$  to the node  $a$  are 4 and 1, respectively. The zero-transit time means the two nodes are close enough with negligible transit time. Not all routes from the source to the sink out of 174,007 possible routes are equally efficient as the arc capacity and the transit time vary in each arc. The algorithm finds the efficient routes, maximum

flow that reaches to the sink and maximum flow that reaches to the intermediate nodes with the specific nodes.

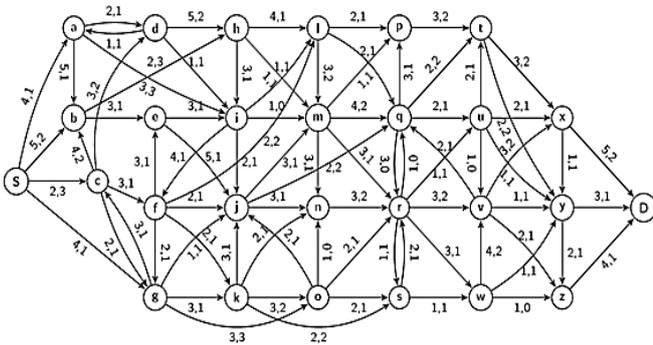


Figure 1: Evacuation Network.

## Result

We have considered the static, the priority based static and the dynamic networks extracted from the Figure 1. 12 units of evacuees as the maximum flow reached into the sink  $D$ , 2 units at the node  $e$  and 1 unit at  $x$  on the static network. The evacuees placed in the intermediate nodes can be sent to the prioritized node, for example, the priority order is  $z, y, x, w$  with the assumption that the nodes near the sink are relatively safe. The flow placed at the node  $e$  can be rerouted to keep one unit of evacuee at the node  $y$  and one unit at the node  $z$ .

The time expanded network developed from the dynamic network in Figure 1, for time horizon  $T = 10$ , consists of 299 nodes and 820 arcs. 20 units of maximum flow and the optimal routes from the source to the intermediate nodes with 121 units of maximum flow to be placed at the end node of the routes.

It took around 2 seconds to run the program for example we considered for time horizon  $T = 10$ . It takes 201.50 seconds and 1226.27 seconds if we set  $T = 30$  and  $T = 50$ , respectively. Detailed output for the latter cases has been omitted here. The algorithm has been coded into Python with version 3.6.1 and was run on the computer with 2.00 GB RAM and i3@2.30 GHZ of processor.

## 4 Conclusion

The evacuation planning problem with flow conservation at intermediate nodes does not allow to send evacuees from risk zone to any other intermediate spots though these are relatively safe even in the case of all evacuees not

being able to reach the destination. Thus, the problem that allows to hold evacuees in such spots, despite sending a maximum flow to the safe destination, is important to investigate for the evacuation planning. In this paper, we proposed the maximum flow evacuation planning problem with relaxed flow conservation constraints at the intermediate nodes with its mathematical formulation and efficient solution procedure. This approach allows holding the evacuees in the relatively safe places during evacuation. The solution procedure is based on the pre-flow-push algorithm, one of the efficient maximum flow algorithms, with necessary modification. Moreover, the algorithm has been extended to work on the dynamic network using the modified time expanded network approach and on the network in which the intermediate nodes are in priority order. The proposed procedure is simple to implement as we described in Subsection 3.4. The limitation associated to current work is that the algorithm solving for dynamic version of the evacuation flow problem leads to a pseudo-polynomial time complexity. The maximum flow aspect of the problem defined for multiple-sources-multiple-sinks network and the other aspects, such as the earliest arrival flow, the quickest flow of the problem, are interesting area for further research.

## 5 Declarations

### 5.1 Acknowledgements

The research of first author is partially supported by University Grants Commission, Nepal under the PhD Fellowship Award scheme (Award No. PhD-72/73 S&T-01) 2016.

### 5.2 Competing Interests

The authors have no conflicts of interest regarding this paper.

### How to Cite this Article:

Phanindra Prasad Bhandari and Shree Ram Khadka "Maximum Flow Evacuation Planning Problem with Non-Conservation Flow Constraint", *Int. Ann. Sci.*, vol. 10, no. 1, pp. 25-32, Sep. 2020.

---

**References**

- [1] A. Kimms and K. C. Maassen, "Optimization and simulation of traffic flows in the case of evacuating urban areas", *OR Spectrum*, vol. 33, pp. 571-593, 2011.
- [2] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network", *Can. J. Math.*, 8, pp. 399-404, 1956.
- [3] L. R. Ford and D. R. Fulkerson, "Constructing maximal dynamic flows from static flows", *Operations Research*, vol. 6, pp. 419-433, 1958.
- [4] L. R. Ford and D. R. Fulkerson, "Flows in Networks", Princeton University Press, 1962.
- [5] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems", *Journal of the Association for Computing Machinery*, vol. 19, pp. 248-264, 1972.
- [6] E. A. Dinic, "Algorithm for solution of a problem of maximum flow in a network with power estimation", *Soviet Math. Dokl.*, vol. 11, pp. 1277-1280, 1970.
- [7] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem", *Journal of the Association for Computing Machinery*, vol. 35, pp. 921-940, 1988.
- [8] J. B. Orlin, "Max flows in  $O(nm)$  time, or better", In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 765-774, 2013.
- [9] P. Kovacs, "Efficient Algorithms for Graph Optimization Problems", PhD thesis, Eotvos Lorand University, Hungary, 2019.
- [10] H. W. Hamacher, S. Heller, W. Klein, G. Köster, and S. Ruzika, "A sandwich approach for evacuation time bounds", In *Pedestrian and Evacuation Dynamics*, pp. 503-513. Springer, Boston, MA, 2011.
- [11] A. Borrmann, A. Kneidl, G. Köster, S. Ruzika, and M. Thiemann, "Bidirectional coupling of macroscopic and microscopic pedestrian evacuation models", *Safety science*, vol. 50, pp. 1695-1703, 2012.
- [12] G. Borradaile, P. N. Klein, S. Mozes, Y. Nussbaum and C. Wulff-Nilsen, "Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time", *SIAM Journal on Computing*, vol. 46, pp. 1280-1303, 2017.
- [13] R. Burkard, K. Dlaska and B. Klinz, "The quickest flow problem", *ZOR-Methods and Models of Operations Research*, vol. 37, pp. 31-58, 1993.
- [14] B. Hoppe, "Efficient Dynamic Network Flow Algorithms", PhD Thesis, Cornell University, 1995.
- [15] D. Gale, "Transient flows in networks", *The Michigan Mathematical Journal*, vol. 6, pp. 59-63, 1959.
- [16] E. Minieka, "Maximal, lexicographic, and dynamic network flows", *Operations Research*, vol. 21, pp. 517-527, 1973.
- [17] W. L. Wilkinson, "An algorithm for universal maximal dynamic flows in a network", *Operations Research*, vol. 19, pp. 1602-1612, 1971.
- [18] B. Hoppe and E. Tardos, "Polynomial time algorithms for some evacuation problems", In *Proceedings of 5th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pp. 433-441, 1994.
- [19] N. Baumann, "Evacuation by Earliest Arrival Flows", PhD Thesis, University of Dortmund, Germany, 2007.
- [20] N. Baumann and M. Skutella, "Solving evacuation problems efficiently: earliest arrival flows with multiple sources", In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 399-410, 2006. IEEE.
- [21] N. Baumann and M. Skutella, "Earliest arrival flows with multiple sources", *Mathematics of Operations Research*, vol. 34, pp. 499-512, 2009.
- [22] M. Steiner, "A Survey on Earliest Arrival Flows and a Study of the Series-parallel Case", Diploma Thesis, University of Kaiserslautern, Germany, 2009.
- [23] S. Ruzika, H. Sperber and M. Steiner, "Earliest arrival flows on series-parallel graphs", *Networks*, vol. 57, pp. 169-173, 2011.
- [24] S. A. Tjandra, "Dynamic Network Optimization with Application to the Evacuation Problem", PhD Thesis, University of Kaiserslautern, Germany, 2003.
- [25] S. Rebennack, A. Arulselvan, L. Eleftheriadou and P. M. Pardalos, "Complexity analysis of maximum flow problem with arc reversal", *Journal of Combinatorial Optimization*, vol. 19, pp. 200-216, 2010.
- [26] T. N. Dhamala and U. Pyakurel, "Earliest arrival contraflow problem on series-parallel graphs", *International Journal of Operations Research*, vol. 10, pp. 1-13, 2013.
- [27] U. Pyakurel, H. W. Hamacher and T. N. Dhamala, "Generalized maximum dynamic contraflow on lossy network", *International Journal of Operations Research Nepal*, vol. 3, pp. 27-44, 2014.
- [28] U. Pyakurel and T. N. Dhamala, "Models and algorithms on contraflow evacuation planning network problems", *International Journal of Operations Research*, vol. 12, pp. 36-46, 2015.
- [29] A. B. Philpott, "Algorithms for Continuous Network Flow Problems", PhD Thesis, University of Cambridge, UK, 1982.
- [30] E. J. Anderson, P. Nash and A. B. Philpott, "A Class of Continuous Network Flow Problems", *Mathematics of Operations Research*, vol. 7, pp. 501-514, 1982.
- [31] A. B. Philpott, "Continuous-time flows in networks", *Mathematics of Operations Research*, vol. 15, pp. 640-661, 1990.
- [32] R. Koch, E. Nasrabadi and M. Skutella, "Continuous and discrete flows over time: a general model based on measure theory", *Mathematical Methods of Operations Research*, vol. 73, pp. 301-337, 2011.
- [33] S. M. Hashemi and E. Nasrabadi, "On solving continuous-time dynamic network flows", *Journal of Global Optimization*, vol. 53, pp. 497-524, 2012.
- [34] L. Fleischer and E. Tardos, "Efficient continuous-time dynamic network flow algorithms", *Operations Research Letters*, vol. 23, pp. 71-80, 1998.
-

- [35] S. R. Khadka and P. P. Bhandari, "Dynamic network contraflow evacuation planning problem with continuous time approach", *International Journal of Operations Research*, vol. 14, pp. 27-34, 2017.
- [36] U. Pyakurel and T. N. Dhamala, "Continuous dynamic contraflow approach for evacuation planning", *Annals of Operations Research*, vol. 253, pp. 573-598, 2017.
- [37] M. Skutella, "An introduction to network flows over time", In: *Research trends in combinatorial optimization*, pp. 451-482. Springer, Heidelberg 2009.
- [38] T. N. Dhamala, "A survey on models and algorithms for discrete evacuation planning network problems", *Journal of Industrial and Management Optimization*, vol. 11, pp. 265-289, 2015.
- [39] S. R. Khadka and P. P. Bhandari, "Model and solution for non-conservation flow evacuation planning problem", *The Nepali Mathematical Sciences Report*, vol. 36, pp. 11-16, 2019.
- [40] D. D. Sleator and R. E. Tarjan, "A data structure for dynamic trees", *Journal of Computer and System Sciences*, vol. 26, pp. 362-391, 1983.
- [41] R. K. Ahujia, T. L. Magnanti and J. B. Orlin, "*Network Flows: Theory, Algorithms and Applications*", New Jersey: Rentice-Hall, 1993.

**Publish your research article in AIJR journals-**

- ✓ Online Submission and Tracking
- ✓ Peer-Reviewed
- ✓ Rapid decision
- ✓ Immediate Publication after acceptance
- ✓ Articles freely available online
- ✓ Retain full copyright of your article.

Submit your article at [journals.aijr.in](http://journals.aijr.in)

**Publish your books with AIJR publisher-**

- ✓ Publish with ISBN and DOI.
- ✓ Publish Thesis/Dissertation as Monograph.
- ✓ Publish Book Monograph.
- ✓ Publish Edited Volume/ Book.
- ✓ Publish Conference Proceedings
- ✓ Retain full copyright of your books.

Submit your manuscript at [books.aijr.org](http://books.aijr.org)