Evacuation Planning Problems with Intermediate Storage

Phanindra Prasad Bhandari* and Shree Ram Khadka

Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal

* Corresponding author's email: phanindra.maths@gmail.com

doi:https://doi.org/10.21467/proceedings.100.9

ABSTRACT

An evacuation planning problem gives a plan on existing road network for disaster management that attempts to send all evacuees from the dangerous zone to the safer zone efficiently. The network flow problems provide important tools for modeling the evacuation tasks. The problems based on the model with flow conservation constraint, that permits an evacuee to be taken out of the disastrous zone only if it can be sent into the safe zone, have been extensively studied for various evacuation scenarios. In this paper, we study dynamic flow problems based on weak flow conservation constraints that allow for an intermediate node to serve as a temporary shelter also with three distinct objectives and propose efficient solution procedures. The first is to maximize the number of evacuees into the safe zones in priority order within the specified time horizon. The second is to achieve the first objective at every time point within the time horizon. And, the third is to fulfill the demand (number of evacuees) at each of the safe zones in minimum possible time horizon in priority order.

1 Introduction

The maximum flow problem, investigated by L. R. Ford and D. R. Fulkerson in 1950s, is the foundation of all the mathematical optimization based evacuation planning problems. Time plays crucial role in modeling real-world evacuation scenarios. The network flow problem known as *maximum dynamic flow (MaxDF) problem*, aiming to send the maximum number of flow unit from the source into the sink in specified time horizon, has been introduced in [7, 8]. There is a pseudo-polynomial algorithm based on the time-expanded network and a polynomial one based on the temporally repeated flow with transit times on the arc as a cost coefficients to solve MaxDF problem on two terminal network. The network flow problems with continuous time setting have been studied in [14, 6, 1, 15].

A problem closely related to a maximum dynamic flow problem is the *quickest flow (QF) problem* that sends a given units of flow from the source to the sink in minimum possible time. This problem can be solved in polynomial time by incorporating the algorithm to solve a maximum dynamic flow problem in a binary search framework. Using Megiddo's method of parametric search [12], a faster algorithm which solves the quickest flow problem in strongly polynomial time can be obtained [4].

The problem that attempts to send a maximum number of evacuees from the source to the sink as earliest as possible within given time horizon is the *earliest arrival flow (EAF) problem*. Gale [9] introduced EAF problem to obtain the maximum amount of flow for every discretized time steps of evacuation time horizon. There exist exponential-time exact algorithms also for the problem [13, 21]. A solution technique for the problem over two terminal series parallel (TTSP) networks that runs with polynomial time complexity has been proposed in [20, 18].

Contraflow approach, reversing the direction of arcs, has also been considered in evacuation planning problems that increases the outbound capacity of the arc and decreases the evacuation time. Here, arcs represent the lanes of a road within the evacuation zone. The first analytical solutions for the maximum contraflow problem on static as well as dynamic network are due to [17]. The evacuation planning problem with contraflow approach in continuous time setting has been studied in [11] and [16]. For a broader overview on evacuation planning problems, we refer to the survey articles [19] and [5].

All the flow problems discussed so far are based on the model with flow conservation at intermediate nodes for which no evacuee is sent out of the source if it cannot reach the sink. There may be some intermediate nodes over

Proceedings DOI: 10.21467/proceedings.100; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-942709-6-6 (eBook)

evacuation network with holding capacity and are relatively safe as compared with the source which are useful to support more evacuees. This paper considers evacuation planning problems over a network which consists of some prioritized intermediate nodes with given storage capacities. The priority depends on how safe the intermediate place is and/or how much capacity does it have. The flow may not be conserved at such intermediate nodes rather can be held at them. The problem without node capacity can be solved efficiently using the notion of a temporally repeated flows (TRFs) generated by repeating all possible source to sink path flows, see [7, 8]. As far as author know, there is no polynomial time method to compute a temporally repeated flow that solves the problem on general network with limited node capacity at intermediate nodes of given priority order.

We revisit the lexicographic maximum dynamic flow (LexMaxDF) problem introduced in [3] in Section 2 and discuss its solution idea in Section 3. The lexicographic earliest arrival flow (LexEAF) problem and the lexicographic quickest flow (LexQF) problem are introduced and solution procedures to them are proposed in Sections 4 and 5, respectively. Section 6 extends the results in continuous time setting. Section 7 concludes the paper.

2 Model Description

An evacuation scenario is represented by a network $N = (V, A, c(a), k(v), \tau_a, T)$ with $|V| = n$, $|A| = m$ where V is the set of nodes v denoting the crossings of road segments, A the set of arcs $a = (v, w), v, w, \in V$ denoting the road segments, $c(a) : c(a) \in Z^+ \cup \{0\}$ is the arc capacity which is the upper bound for the evacuees to pass along the arc a in a unit time, $k(v) : k(v) \in Z^+ \cup \{0\}$ is the node capacity which is the upper bound of evacuees to be held at node v, τ_a a non-negative integer, the transit time which is the time required for an evacuee to travel along arc a and T is the time horizon within which the evacuation process is supposed to be completed. Special nodes denoted by s and d are the source and the sink, respectively.

For a discrete dynamic flow model, the non-negative flow variables $f : A \times \{0, 1, \ldots, T\} \to Z^+ \cup \{0\}$ specify the flow over time in the network N. More precisely, the number $f(a,t)$ equals the number of flow units entering arc a at time step t. The number of flow units entering arc a at time step t is assumed to be bounded by the capacity of an arc, i.e.,

$$0 \leq f(a,t) \leq c(a) \ \forall a \in A \text{ and } \forall \ t \in \{0, 1, \ldots, T\}. \tag{1}$$

In each time step $t \in \{0, 1, \ldots, T\}$, the flow entering a node $v \in V \backslash \{s, d\}$ has to be at least as large as the flow exiting out of it, i.e.,

$$\sum_{a \in \delta^-(v)} f(a, t - \tau_a) - \sum_{a \in \delta^+(v)} f(a,t) \geq 0 \quad \forall \, v \in V \backslash \{s, d\} \text{ and } \forall \, t \in \{0, 1, \ldots, T\}. \tag{2}$$

Here, $\delta^-(v)$ and $\delta^+(v)$ denote the set of arcs entering and leaving node $v \in V$, respectively. Additionally, it is allowed that flow is held at some node $v \in V$ if $k(v) \neq 0$. To this end, we introduce variables $h(v, T)$ for all $v \in V \backslash \{s\}$ and require

$$0 \leq h(v,T) = \sum_{t=0}^{T} \sum_{a \in \delta^-(v)} f(a, t - \tau_a) - \sum_{t=0}^{T} \sum_{a \in \delta^+(v)} f(a,t) \ \ \forall \, v \in V \backslash \{s\}. \tag{3}$$

The total flow of evacuees leaving source s equals the total flow of the evacuees held at any node $v \in V \backslash \{s\}$ over the time horizon T, i.e.,

$$\sum_{t=0}^{T} \sum_{a \in \delta^+(s)} f(a,t) = \sum_{v \in V \backslash \{s\}} h(v,T). \tag{4}$$

With respect to the constraints from (1) to (4), the *lexicographic maximum dynamic flow problem* asks to send as many flow units from source to sink as possible for each time step $t \in \{0, 1, ..., T\}$, and as a secondary objective, a maximum number of flow units to nodes other than the sink in the same manner. The latter is subjected to a prioritization of the nodes $v \in V$ from lower to higher priority as $s = v_1 \preceq v_2 \preceq \ldots \preceq v_n = d$. This sorting reflects the fact that certain destinations in an evacuation process have different priority. Thus, the objective function of the LexMaxDF problem asks to lexicographically maximize the number of flow units held at the nodes within the pre-specified time horizon T where the nodes are sorted in a given prioritization, i.e.,

$$\text{lex max } \left(h(v_n, T), h(v_{n-1}, T), \ldots, h(v_2, T)\right). \tag{5}$$

In the context of evacuation modeling, this objective function can be interpreted as follows. It is $v_n = d$, and thus, a maximum flow from s to d has to be found in the first place. Since $k(d) = +\infty$, the value of this flow is not bounded by the node capacity. Then, let $v_i \neq d$ be the node with highest priority (other than the sink) having positive node capacity $k(v_i)$. Due to the lexicographical optimization, the problem asks for a flow sending as much flow as possible to node v_i among the set of maximal s-v_i flows. This idea is repeated until the flow to the node with lowest priority and positive node capacity is eventually considered.

3 Solution to Lexicographic Maximum Dynamic Flow Problem

Consider a uniform path length (UPL) network $N = (V, A, c(a), k(v), \tau_a, T)$ with prioritized nodes $v_i \in V$ sorted as $s = v_1 \preceq v_2 \preceq \ldots \preceq v_n = d$. A directed dynamic network is a uniform path length (UPL) network for which the sum of the transit times on arcs on any possible path from the source s to the node v_i, for all $v_i \in V$, is equal, see Fig. 1. The goal is to solve the lexicographic maximum dynamic flow problem on N in polynomial time using temporally repeated flows. The main idea of the solution procedure of the problem is to find $s - v_i$ paths, for all $v_i \in V : k(v_i) > 0$, at all possible time steps $t \in \{0, 1, \ldots, T\}$ with corresponding flow value and send as many units of flow as possible along the paths as long as possible. Such paths can be found by decomposing the flow on solving the *lexicographic minimum cost flow (LexMinCF) problem* on N. The LexMinCF problem asks to lexicographically minimize the cost B_i for sending the number of flow units $\mathbf{f}(v_i)$ at each of the prioritized nodes $v_i \in V$, i.e.,

$$\text{lex min } \left(B_n(\mathbf{f}(v_n)), B_{n-1}(\mathbf{f}(v_{n-1})), \ldots, B_1(\mathbf{f}(v_2))\right) \tag{6}$$

where the transit time $\tau_a \; \forall \, a \in A$ is switched into the cost b_a.

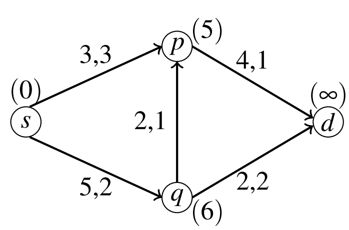

Fig. 1 A uniform path length (UPL) network N with source node s, arc capacity and transit times next to each arc and the node capacity inside the parenthesis near by each node.

The minimum cost flow algorithm in [10], for example, can be applied to solve LexMinCF problem on N repeatedly for each $v_i \in V : k(v_i) > 0$ in given priority order on the corresponding residual network of N with additional arc (v_i, s) with capacity equal to $k(v_i)$ and transit time $-(T+1)$. This yields a set of all $s - v_i$ paths that could be temporally repeated from time step zero, denoted as Γ_{v_i}, for each $v_i \in V : k(v_i) > 0$. It is noteworthy to mention that the path γ_{v_i} is a chain of nodes and arcs in the network N starting at the source s and terminating at node v_i.

The limitation of the temporally repeated flow along path on Γ_{v_i} is that it may not induce an optimal solution to the problem or the flow becomes infeasible for some node v_i on N due to fixed node capacity. Thus, it is necessary to find an extended set $\Gamma^E_{v_i}$ that contains all minimum cost $s - v_i$ paths that exist at any time $t \in \{0, 1, ..., T\}$ on the residual network of N with respect to the optimal flow $\mathbf{f}(v_{i+1})$ at previous immediate prioritized node v_{i+1}. It is also necessary to push flow units of corresponding values along each path as long as possible, unless $k(v_i)$ is satisfied. Moreover, the flow is pushed along the paths in $\Gamma^E_{v_i}$ with the strategy of saving unused paths for the use of next less prioritized node v_{i-1} without violating the optimality at v_i. This is assured by selecting the path with highest $F_t(\gamma_{v_i})$, the time step at which the flow along γ_{v_i} stops to get repeated, among the paths $\gamma_{v_i} \in \Gamma^E_{v_i}$ with highest $I_t(\gamma_{v_i})$, the time step at which the flow along γ_{v_i} starts to get repeated, at the first and so on. This procedure yields an optimal solution to the LexMaxDF problem on UPL network N in polynomial time.

4 Lexicographic Earliest Arrival Flow Problem

Since it is usually not known when the disaster will actually happen, it is desirable to organize an evacuation in such a way that as many evacuees as possible are saved. An earliest arrival flow problem aims to optimize the evacuation process for every time step within pre-specified time horizon T. A LexMaxDF problem that fulfills the objective function (5) at each time step $t \in \{0, 1, \ldots, T\}$ together with the constraints (1) to (4) is a *lexicographic earliest arrival flow problem*. That is, the objective of a LexEAF problem is to send a maximum number of evacuees at the possible earliest time from the disastrous zone to the safety zone together with relatively safe zones within the given time horizon.

It is clear that every earliest arrival flow is a maximum dynamic flow for given time horizon. However, the converse is not always true for general network. In the following, a solution procedure is proposed that obtains a lexicographic maximum dynamic flow on a typical network and claimed that this flow schedule has an earliest arrival property.

Let us consider the LexMaxDF problem on a uniform path length two terminal series parallel (UPL-TTSP) network $N = (V, A, c(a), k(v), \tau_a, T)$ with prioritized nodes $v_i \in V$ sorted as $s = v_1 \preceq v_2 \preceq \ldots \preceq v_n = d$ with $k(v_i) \in \{0, +\infty\}$. The solution procedure discussed in Section 3 is applied to solve the LexMaxDF problem where the minimum cost flow algorithm [2] is applied to solve the LexMinCF problem. The extended set $\Gamma^E_{v_i}$ induces an optimal dynamic flow for each v_i on N in polynomial time. Moreover, the network N being a two terminal series parallel in structure, this flow has an earliest arrival property [18].

5 Lexicographic Quickest Flow Problem

Let us restrict the node capacity $k(v)$ to be fulfilled as an upper bound as well as a lower bound by the total flow value that is supposed to be held at the node v_i on $N = (V, A, c(a), \tau_a, k(v))$ in the LexMaxDF problem discussed in Section 3. Then the limited node capacity $k(v)$ can be taken as demand, say, $\mu(v)$) at $v : \forall v \in V \backslash \{s\}$. This consideration allows to see a dynamic flow problem on N with demands at nodes and asking for a minimum time to satisfy these demands in given priority order. In the following, we formally define this problem which is termed as *lexicographic quickest flow problem*.

Consider a UPL network $N = (V, A, c(a), \tau_a, \mu(v))$ with prioritized nodes $v_i \in V$ $s = v_1 \preceq v_2 \preceq \ldots \preceq v_n = d$ such that $\sum_i \mu(v_i) = 0$ where $\mu(v_i) \in Z^+ \cup \{0\}$ is the demand at the node v_i. The negative demand at the source s is termed as supply. Moreover, we restrict the arc capacity $c(a)$ for each arc $a \in A$ to be strictly positive. Then the LexQF problem finds a feasible dynamic flow f_{v_i} of given value $\mu(v_i)$ on the network N with prioritized nodes v_i from the source s to the node v_i which sends the given $\mu(v_i)$ units of flow from s to v_i in the minimum number $T(\mu(v_i))$ of time units obeying the capacity constraints (1), the weak flow conservation constraints (2) for time horizon $T(\mu(v_i))$ and the modified form of constraint (3) as

$$\sum_{t=0}^{T(\mu(v))} \sum_{a\in\delta^-(v)} f(a,t-\tau_a) - \sum_{t=0}^{T(\mu(v))} \sum_{a\in\delta^+(v)} f(a,t) = \mu(v) \ \forall v \in V \backslash \{s\} \tag{7}$$

where $T(\mu(v))$ is the minimum time that is required to send $\mu(v)$ units of flow from the source to the node v. Moreover, the Equation (4) holds true due to our consideration $\sum_i \mu(v_i) = 0$. Together with these assumptions, the objective of LexQF problem is

$$\text{lex min } \left(T(\mu(v_n)), T(\mu(v_{n-1})), \ldots, T(\mu(v_2))\right). \tag{8}$$

The existence of lexicographic quickest flow on N follows from the fact that N is a connected network and capacity $c(a)$ is positive for each $a \in A$. The solution procedure to the LexQF problem is similar to the binary search method of solving a quickest flow problem in [4]. Since we are interested in finding such minimum time T_m for each node $v_i \in V : \mu(v_i) > 0$ in a priority order, the maximum flow computation technique developed in Section 3 is adopted as a subroutine of the procedure with necessary modification. Due to the nature of the construction of a maximum flow using this technique, the maximum flow of value $\mu(v_i)$ obtained for time horizon T, could also be possible to find in lesser time horizon T' for some nodes v_i. That is, it cannot be guaranteed that the time T at which the dynamic flow of value $\mu(v_i)$ can be sent to v_i is the minimum time to attain this flow value. One should check whether the same flow value is attained for some lesser time point T'.

6 Solutions with Continuous Time Setting

The lexicographic maximum dynamic flow problem, the lexicographic quickest flow problem and the lexicographic earliest arrival flow problem modeled on the network N with continuous time setting for time horizon T can also be solved efficiently by applying the notion of natural transformation of flows over discrete time setting [6]. The notion states that the amount of flow, say f_d, that arrives at the node w through the arc $a = (v,w)$ at time step t in the discrete time setting is equal to the amount of flow, say f_c, arriving at w through the arc $a = (v,w)$ during the unit interval of time at the beginning of time step t, i.e., $f_d(a,t) := f_c(a,[t,t+1))$ for all $t \in \{0,1,\ldots,T-1\}$.

7 Concluding Remark

The domain of evacuation planning problems based on the network flow model has been flourished with efficient solutions with various network attributes. A common feature of the problems is that the flow function obeys flow conservation constraints at each intermediate node. In particular, maximum dynamic flow problem, earliest arrival flow problem and quickest flow problem have great applicability in evacuation planning problems due to realization of time constraint. In this paper, we studied these problems that lexicogaphically achieve the goals on the network with prioritized intermediate shelters of given capacities. Evacuation problems with intermediate shelters could be extra benefit during disasters. We proposed polynomial time solution techniques for the LexMaxDF problem and LexQF problem modeled on UPL network and for LexEAF problem modeled on UPL-TTSP network. Investigation of solution to these problems modeled on more general network would extend the domain and scope of their applicability in real world evacuation plans.

Acknowledgments: The first author would like to thank University Grants Com-mission, Nepal for partial financial support under PhD Fellowship Award 2016.

References

1. Anderson, E.J., Nash, P., Philpott, A.B.: A class of continuous network flow problems. Mathematics of Operations Research **7**(4), 501–514 (1982)

2. Bein, W.W., Brucker, P., Tamir, A.: Minimum cost flow algorithms for series-parallel networks. Discrete Applied Mathematics **10**(2), 117–124 (1985)
3. Bhandari, P.P., Khadka, S.R., Schafer, L., Ruzika, S.: Lexicographically maximum dynamic flow with vertex capacities. (Submitted) (2019)
4. Burkard, R.E., Dlaska, K., Klinz, B.: The quickest flow problem. Zeitschrift fur Operations Research **37**(1), 31–58 (1993)
5. Dhamala, T.N.: A survey on models and algorithms for discrete evacuation planning network problems. Journal of Industrial & Management Optimization **11**(1), 265–289 (2015)
6. Fleischer, L., Tardos, E.: Efficient continuous time dynamic network flow algorithms. Operations Research Letters **23**(3-5), 71–80 (1998)
7. Ford, L.R., Fulkerson, D.R.: Constructing maximal dynamic flows from static flows. Operations research **6**(3), 419–33 (1958)
8. Ford, L.R., Fulkerson, D.R.: Flows in networks. Princeton University Press (1962)
9. Gale, D.: Transient flows in networks. Michigan Mathematical Journal **6**, 59–63 (1959)
10. Goldberg, A.V., Tarjan, R.E.: Finding minimum-cost circulations by canceling negative cycles. Journal of the ACM (JACM) **36**(4), 873–886 (1989)
11. Khadka, S.R., Bhandari, P.P.: Dynamic network contraflow evacuation planning problem with continuous time approach. International Journal of Operations Research **14**(1), 27–34 (2017)
12. Megiddo, N.: Combinatorial optimization with rational objective functions. Math. Oper. Res. **4**, 414—424 (1979)
13. Minieka, E.: Maximal, lexicographic, and dynamic network flows. Operations Research **21**(2), 517–527 (1973)
14. Philpott, A.B.: Algorithms for continuous network flow problems. Doctoral dissertation, University of Cambridge, UK (1982)
15. Philpott, A.B.: Continuous-time flows in networks. Mathematics of Operations Research **15**(4), 640–661 (1990)
16. Pyakurel, U., Dhamala, T.N., Dempe, S.: Efficient continuous contraflow algorithms for evacuation planning problems. Annals of Operations Research **254**, 335–364 (2017)
17. Rebennack, S., Arulselvan, A., Elefteriadou, L., Pardalos, P.M.: Complexity analysis for maximum flow problems with arc reversals. Journal of Combinatorial Optimization **19**(2), 200–216 (2010)
18. Ruzika, S., Sperber, H., Steiner, M.: Earliest arrival flows on series-parallel graphs. Networks **57**(2), 169–173 (2011)
19. Skutella, M.: An introduction to network flows over time. In: Research trends in combinatorial optimization, pp. 451-482. Springer, Heidelberg (2009)
20. Steiner, M.: A survey of earliest arrival flows and a study of the series-parallel case. Diploma dissertation, University of Kaiserslautern (2009)
21. Wilkinson, W.L.: An algorithm for universal maximal dynamic flows in a network. Operations Research **19**(7), 1602–1612 (1971)