International Journal of Applied Science and Engineering

Temporally repeated fuzzy maximum dynamic flow with intermediate storage

Dipak Babu Amgain [1], Durga Prasad Khanal [2*], Tanka Nath Dhamala [1]

[1] *Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal*
[2] *Saraswati Multiple Campus, Tribhuvan University, Kathmandu, Nepal*

ABSTRACT

The classical network flow models have been extensively studied in deterministic ways without addressing uncertainty, which is not able to capture the real-world systems. Unlike classical models, the fuzzy maximum network flow problem with intermediate storage incorporates uncertainty in arc capacities (due to congestion or other factors) and accounts for storage constraints at intermediate nodes. Recent trends in research are focusing on solving maximum dynamic network flow problems with intermediate storage in fuzzy environments. The solution strategies presented in the literature are not efficient, as the algorithms rely on a time-expanded network rather than the flow decomposition idea to use temporally repeated formulae. In this paper, an efficient algorithm is proposed to solve this problem by taking the time-invariant arc capacities over the given time horizon in fuzzy environment. For this, we model arc capacities, demands, and storage capacities using triangular fuzzy numbers to address real-world variability. We formally define the fuzzy maximum flow problem and the earliest arrival flow problem with intermediate storage, assuming crisp transit times at all time steps. Our solutions employ a temporally repeated flow algorithm, where flow-balancing paths are extracted from an excess flow network derived from static intermediate storage solutions. Additionally, we apply this approach to solve the earliest arrival flow problem with intermediate storage in a series-parallel network.

Keywords: Fuzzy flow, Dynamic network, Intermediate storage, Temporally repeated flow, Earliest arrival flow.

Received: July 18, 2025
Revised: September 7, 2025
Accepted: October 25, 2025

Corresponding Author:
Durga Prasad Khanal
durga.khanal@smc.tu.edu.np

Publisher:
Chaoyang University of Technology
ISSN: 1727-2394 (Print)
ISSN: 1727-7841 (Online)

1. INTRODUCTION

The study of network flow problems has been a central topic in operations research, optimization, and graph theory for decades. These problems are foundational in modeling the transportation and distribution of vehicles, goods, data, resources, etc. across interconnected systems (Ahuja et al., 1993). However, traditional formulations assume ideal conditions where flows passing through the network such as capacities, demands, and costs are fixed and deterministic. In reality, these assumptions often do not hold, and the resulting models may not accurately capture the complexities of real-world systems. Real-world networks are often subject to uncertainties, such as fluctuating link capacities, random packet losses, and varying demands. These factors complicate the traditional network flow models, necessitating a more flexible approach. Han et al. (2014) investigated the concept of the uncertainty theory to the maximum flow problem. A significant work exists on network flow problems under uncertainty (Pangeni and Dhamala, 2025). Fuzzy logic provides an effective framework for modeling such uncertainties (Zadeh, 1965). Fuzzy numbers are used to represent the uncertain capacity of network links, as they allow for the modeling of situations where exact values are difficult to specify (Klir and Yuan, 1995). Fuzzy optimization has become an important tool for dealing with uncertainty in network flow problems.

Most of the existing works on network flow models assume crisp capacities and travel times, while only limited studies address the uncertainty by reflecting the real-world systems in fuzzy environment. On the other hand, flow models incorporating intermediate storage represent a novel and rapidly evolving area. These scenarios motivate us to model and develop temporally repeated flow methods for fuzzy dynamic flow problems: the maximum flow problem and the earliest arrival flow problem, incorporating intermediate storage to address real-world problems efficiently.

Besides network flow problems, fuzzy logic can also be used in AI-based models and algorithms. Khan et al. (2023) integrated fuzzy logic with generative adversarial networks to achieve secure video compression and encryption. Recent work by Laghari et al. (2023) applies deep residual-dense network with a bidirectional recurrent neural network in deep learning model for the automated atrial fibrillation detection with high accuracy. Zhao et al. (2025) proposed a novel AI-based model which applies the enhancement for multimodal sentiment analysis. Fuzzy genetic algorithms have been used to general reliable approximate solution for maximum fuzzy flow, fuzzy regression and fuzzy controller design (Akrami, 2025). Yan et al. (2023) presented an interactive random-fuzzy optimal power flow method to enhance static voltage stability and security in transmission-distribution network. Almutairi and Zhang (2024) highlighted the need for a reliable and efficient path finding solution for low-power and lossy networks with impact on packet delivery and routing.

1.1 Crisp Models

The theory of crisp (non-fuzzy) maximum flow problem (MFP) dates back to the work of Ford and Fulkerson, who formulated the MFP as a means of determining the maximum flow from a source node to a sink in a network (Ahuja et al., 1993). Since then, a variety of extensions have been proposed, and many efficient algorithms have been developed to solve the MFP with/without the time parameter (Dhamala, 2015). MFP with arc-travel time is known as the maximum dynamic flow problem (MDFP), which can be solved either in a time-expanded network (a static form of dynamic network) or by using the method of temporally repeated flow (TRF) of Ford and Fulkerson (1962). The network flow problem with intermediate storage is to store flow at the sink together with intermediate nodes temporally (Dhamala et al., 2025) or permanently (Khanal et al., 2025), which cannot completely be sent to the sink node. Pyakurel and Dempe (2021) and Khanal et al. (2025) contributed to the solution of the maximum dynamic flow problem with prioritized intermediate storage. Later, this problem has been generalized in lossy networks (see Dhamala et al., 2024; Adhikari and Dhamala, 2025). The contraflow model is another variant of MFP, which has been widely accepted in evacuation planning (Pyakurel and Dhamala, 2016).

1.2 Fuzzy Models

The fuzzy flow problems extend traditional flow models by introducing uncertainty in the networks' capacity and flow parameters (Chanas, 1987; Bozhenyuk et al., 2017). Fuzzy network flow problems were first explored by researchers like Zimmermann (1991), who used fuzzy set theory to handle imprecise capacities and demands in flow networks. Later, fuzzy linear programming models were developed to solve such problems, where both the capacities and demands are modeled as fuzzy numbers (Ghanbari et al., 2020). One of the first papers on fuzzy networks, by Kim and Roush (1982), developed the fuzzy flow theory, presenting on fuzzy matrices. In the papers (Chanas and Kolodzijczyk, 1982; Chanas and Kolodzijczyk, 1986), the authors discussed this theory by means of minimum cut techniques. Diamond (2001) provided an algorithm for the fuzzy maximum flow problem (FMFP) based on Edmonds-Karp's algorithm and proved the interval-valued max-flow min-cut theorem. Bozhenyuk et al. (2017) discussed an algorithm of FMFP based on the classical maximum flow algorithm. They also defined the FMFP in time-dependent dynamic network.

Based on a linear programming problem, Soltani and Haji (2007) introduced a modified backward pass (MBP) approach to solve the project scheduling problem, avoiding the negative solution for time-related project parameters in trapezoidal fuzzy numbers. Similarly, Kumar and Kaur (2012) solved the problem by the ranking method and used the MBP approach, a non-standard subtraction method, to address the non-negativity of the flow value in fuzzy.

Zhang et al. (2002) addressed the fuzzy linear programming model of network flow problems in the presence of uncertainty and proposed a fuzzy version of the simplex algorithm. In (Chanas and Kolodzijczyk, 1982), a solution of FMFP was proposed based on interval arc capacities. Chanas and Kolodzijczyk continuously studied the fuzzy flow theory by considering arc capacity as upper and lower bounds with a satisfaction function (Chanas and Kolodzijczyk, 1982; Chanas and Kolodzijczyk, 1986). They also used the minimum cut approach to solve the FMFP. Similarly, Hernandes et al. (2007) defined the α-cut method to solve this problem in the residual network. There has been a lot of work done on uncertain graph and network optimization (Peng et al., 2024). Gerasimenko (2024) presented a dynamic flow algorithm designed for emergency evacuation scenarios, allowing for lane reversals and storage at intermediate nodes in fuzzy environments. The author incorporated an incomplete intuitionistic fuzzy preference relation to handle uncertainties and multiple conflicting criteria in decision-making. As in the crisp model, there are two ways of solving the fuzzy maximum dynamic flow problem (FMDFP): with the time-expanded network (converting dynamic network into static) (see Bozhenyuk et al., 2017; Gerasimenko, 2024) and via the minimum cost flow algorithm, considering arc-travel time as arc-travel cost. The second one is significantly more efficient than the first one because the running time in the time-expanded network depends on the time horizon.

1.3 Research Gap

Although Gerasimenko (2024) considered fuzzy arc capacities and travel times with intermediate storage in maximum dynamic network flow models, the author mainly used time-expanded networks while solving the problems. This approach is inefficient due to the limitations of using a time-expanded network. On the other hand, Khanal et al. (2025) established a temporally repeated solution using flow balancing paths in network flow problem with intermediate storage, however, they do not model the fuzzy environment. In contrast, our paper aims to bridge this gap by developing a temporally repeated flow model that integrates fuzzy arc capacities and crisp transit time in a dynamic flow model with intermediate storage.

1.4 Our Contribution

The contributions of this research are summarized as follows:

- We propose a flow-balancing path approach to solve the fuzzy maximum dynamic flow model with intermediate storage.
- We extend this approach to the earliest arrival flow problem with intermediate storage in series-parallel networks.
- We provide numerical illustrations to make a comprehensive discussion about the efficiency and applicability of the proposed methods.

1.5 Organization of the Paper

The remainder of this paper is structured as follows: Section 2 sets the basic terminologies and notations related to the fuzzy network flow problems discussed in this paper. Section 3 discusses the problem setup and provides a formal mathematical formulation of the fuzzy maximum dynamic flow problem with intermediate storage (FMDFIS). In Section 4, we present algorithms for the solution of FMDFIS based on temporally repeated formula and give numerical illustrations. Section 5 includes the mathematical model of fuzzy earliest arrival flow with intermediate storage (FEAFIS) in a series-parallel network with its solution strategy. Section 6 includes the results discussion and Section 7 concludes the paper, suggesting the importance of FMDFIS in the real-world scenario of transportation networks.

2. BASIC TERMINOLOGIES AND NOTATIONS

Throughout this paper, we use $\tilde{A}$ instead of A to distinguish the fuzzy set from the crisp set. A fuzzy set $\tilde{A}$ in the universe of discourse X is characterized by its membership function $\mu_{\tilde{A}}: X \to [0,1]$ which gives the membership degree of the element in X. It is said to be normal if and only if there exists a $z \in X$ such that $\mu_{\tilde{A}}(z) = 1$. Otherwise, it is *subnormal*. A fuzzy number $\tilde{a}$ is a fuzzy subset of the real line $\mathbb{R}$. Triangular and trapezoidal fuzzy numbers are the common fuzzy numbers preferred in most of the applications (Klir and Yuan, 1995). A fuzzy number can be converted to a crisp interval as per the degree values. For $\alpha \in [0,1]$, the set $\tilde{A}_\alpha$ of all elements $z \in X$ such that $\mu_{\tilde{A}}(z) \geq \alpha$ is called α-*cut* (for more detail, see Dubois and Prade, 1978).

The fuzzy numbers are compared on the basis of their rank values. A ranking function R maps each fuzzy number into a real number. There is no unique way to identify the rank of numbers. Several methods of ranking fuzzy numbers have been proposed (Dubois and Prade, 1978; Kaufmann and Gupta, 1985; Liou and Wang, 1992; Mishra, 2021), and comparisons of these methods have been made by Liou and Wang (1992).

2.1 Fuzzy Arithmetic

The basic arithmetic operations on fuzzy numbers are addition, subtraction, multiplication, division, and scalar multiplication, defined by Zadeh (1965). Addition and subtraction operations preserve the geometrical shape of the membership functions of triangular and trapezoidal fuzzy numbers, but multiplication and division operations might not preserve (Klir and Yuan, 1995). The fuzzy numbers that are considered in this paper are normal triangular fuzzy numbers $\tilde{a} = (a_1, a_2, a_3)$ with membership function

$$\mu_{\tilde{a}}(z) = \begin{cases} \frac{z-a_1}{a_2-a_1} & \text{for } a_1 \leq z < a_2, \\ 1 & \text{for } z = a_2, \\ \frac{a_3-z}{a_3-a_2} & \text{for } a_2 < z \leq a_3, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Let $\tilde{b} = (b_1, b_2, b_3)$ be another triangular fuzzy number. Then addition and subtraction operations of $\tilde{a}$ and $\tilde{b}$ are given as follows:

$$\tilde{a} + \tilde{b} = (a_1 + b_1, a_2 + b_2, a_3 + b_3) \text{ and } \tilde{a} - \tilde{b} = (a_1 - b_3, a_2 - b_2, a_3 - b_1) \tag{2}$$

Similarly, the multiplication of a triangular fuzzy number by a crisp number $k \in \mathbb{R}$ is defined by $k\tilde{a} = (ka_1, ka_2, ka_3)$ if $k > 0$ and $k\tilde{a} = (ka_3, ka_2, ka_1)$ if $k < 0$. Here, we use the ranking function R as given by Liou and Wang (1992), which is defined as

$$R(\tilde{a}) = \frac{1}{4}(a_1 + 2a_2 + a_3) \tag{3}$$

When computing residual capacities in fuzzy network flows using the general fuzzy subtraction method presented in Equation (2), negative components in fuzzy numbers may arise. It is a physically invalid outcome for flow quantities. The MBP method can be used to prevent negative flows in fuzzy network computations Kumar and Kaur (2012). This non-standard subtraction method is defined in Equation (4).

$$\tilde{a} - \tilde{b} = (c_1, c_2, c_3), \text{where } c_3 = \max\{0, a_3 - b_3\}, c_2 =$$

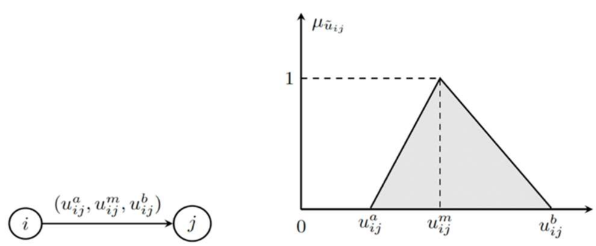
$\max\{0, \min(c_3, a_2 - b_2)\}$

$$c_1 = \max\{0, \min(c_2, a_1 - b_1)\} \tag{4}$$

Throughout the paper, we adopt the MBP method in network flow calculation instead of Zadeh's extension principle for subtraction.

2.2 Network with Fuzzy Capacities

Consider a directed network $N = (V, A, s, d)$, where V is the set of nodes and $A \subseteq V \times V$ is the set of directed edges (arcs) (i, j) connecting nodes i and j, and s and d are the source node and sink node, respectively. Let $|V| = n$ (number of nodes) and $|A| = m$ (number of arcs). We denote the network N with fuzzy arc-capacity $\tilde{u}_{ij}$ as $\tilde{N}$. Making it more realistic, the membership function parameters were adjusted to reflect the expected distribution of arc-capacities on the basis of historical data or expert assessments (Bozhenyuk et al., 2017). As shown in Fig. 1 the fuzzy arc-capacity in triangular form $\tilde{u}_{ij} = (u_{ij}^a, u_{ij}^m, u_{ij}^b)$ can be characterized with the core (most certain value u_{ij}^m) and support (possible values) u_{ij}^a, u_{ij}^b.



(a) (b)

Fig. 1. (a) Triangular arc-capacity $\tilde{u}_{ij}$. (b) Graph of membership function of $\tilde{u}_{ij}$.

Let $\tilde{x}$ be a fuzzy arc-flow of a network $\tilde{N}$. A residual network of $\tilde{N}$ is an incremental network $\tilde{N}(\tilde{x})$ with residual arc capacity $\tilde{r}_{ij} = \tilde{u}_{ij} - \tilde{x}_{ij}$ along an arc (i, j) and $\tilde{r}_{ji} = \tilde{x}_{ij}$ along the (j, i), where $\tilde{x}_{ij}$ is a fuzzy flow value on an arc (i, j).

(a) (b)

Fig. 2. (a) Arc capacities of fuzzy network. (b) Residual arc capacities.

Considering $\tilde{U}_j$ as the storage capacity of the node $j \in V \setminus \{s, d\}$ and $\tilde{x}_j(\sigma)$, the flow stored at node j over the time $\sigma \in \boldsymbol{T}$, we use the notation $\tilde{N} = (V, A, s, d, \tilde{u}, \tilde{U}, t)$ for (discrete-time) dynamic network. The denotations of the above discussed terminologies are listed in Table 1.

3. FUZZY MAXIMUM DYNAMIC FLOW WITH INTERMEDIATE STORAGE

The concept of FMDFIS is to store flow at intermediate node together with sink node within the dynamic network, where the intermediate nodes allow temporary/permanent collection (accumulation) of flows out from the source node over the given time horizon, enabling more flexibility in managing network flow under uncertain conditions. This model is particularly useful in the evacuation network as well as supply chains and energy systems, where intermediate storage can help manage fluctuations in capacity and demand (Pyakurel and Dempe, 2021; Khanal, 2024; Khanal et al., 2025). In an evacuation network, it aims to transport as many evacuees as possible in case of emergency from a dangerous area (source) to a safe location (sink) in a fuzzy environment, which allows intermediate storage at prioritized nodes with less risk.

Table 1. Basic notations and their definitions

Notation	Definition
$\tilde{A}$	Fuzzy set
$\tilde{a}$	Fuzzy number
$\mu_{\tilde{A}}$	Membership function of $\tilde{A}$
$\tilde{u}_{ij}$	Fuzzy capacity of the arc (i, j)
$\tilde{U}_j$	Storage capacity of the node j
$\tilde{x}_{ij}$	Fuzzy flow along the arc (i, j)
$\tilde{x}_j$	Stored flow at node j
$\tilde{r}_{ij}$	Fuzzy residual capacity of the arc (i, j)
$R(\tilde{a})$	Rank of the fuzzy number $\tilde{a}$
$\tilde{N}$	Network with fuzzy parameter $\tilde{u}$
(a_1, a_2, a_3)	Triangular fuzzy number with supports a_1 and $a_3$and core a_2
$\tilde{A}_\alpha$	α-cut
V	Set of vertices
A	Set of arcs
s and d	Source node and sink node
t	Transit time
T	Time horizon
$\boldsymbol{T}$	Set of discrete times

3.1 Mathematical Model of FMDFIS

Based on the paper of Pyakurel and Dempe (2021), we define the FMDFIS as follows.

$$\text{Maximize } \tilde{v}(T) = \sum_{i:(i,d)\in A} \sum_{\theta=t_{id}}^{T} \tilde{x}_{id}(\theta - t_{id}) + \sum_{j\in V\setminus\{s,d\}} \tilde{x}_j(T) \tag{5}$$

$$\text{Subject to } \sum_{j:(s,j)\in A} \sum_{\theta=0}^{T} \tilde{x}_{sj}(\theta) = \tilde{v}(T) \tag{6}$$

$$\sum_{i:(i,j)\in A} \sum_{\theta=t_{ij}}^{\sigma} \tilde{x}_{ij}(\theta - t_{ij}) - \sum_{k:(j,k)\in A} \sum_{\theta=0}^{\sigma} \tilde{x}_{jk}(\theta) \geq \tilde{0} \ \forall j \in V \setminus \{s, d\} \ \forall \sigma \in T \tag{7}$$

$$\tilde{0} \leq \tilde{x}_{ij}(\sigma) \leq \tilde{u}_{ij} \; \forall \, (i,j) \in A \; \forall \, \sigma \in \boldsymbol{T} \tag{8}$$

$$\tilde{0} \leq \tilde{x}_{j}(\sigma) \leq \tilde{U}_{j} \; \forall \, j \in V \setminus \{s,d\} \; \forall \, \sigma \in \boldsymbol{T} \tag{9}$$

The objective (5) is to maximize the total flow $\tilde{v}(T)$ reaching the destination node d first by time T and total flow stored at intermediate nodes at the end of the time horizon T according to the priority order. Constraints (6) ensure all flow leaving the source s either reaches d (after delays t_{id}) or remains stored in prioritized intermediate nodes within the time window T. Constraints (7) imply that the total outgoing flow should not exceed the total incoming flow for each intermediate node j and time σ. The arc-capacity constraints (8) say that flow $\tilde{x}_{ij}(\sigma)$ on arc (i,j) at time σ must be non-negative and within the fuzzy capacity $\tilde{u}_{ij}$. Similarly, the constraints (9) ensure that stored flow $\tilde{x}_j(\sigma)$ at node j must respect the fuzzy storage limit $\tilde{U}_j$.

Here, we consider the storage capacity of intermediate nodes must be greater than or equal to T times the sum of the capacity of incoming arcs at the intermediate node so that all the feasible flows can be stored at each storage node (Khanal, 2024). The problem FMDFIS becomes static whenever the time parameter is excluded or simply put $T = 0$ and $t_{ij} = 0$ for all $(i,j) \in A$. In this case, we simply use the notation of fuzzy static flow by $\tilde{x}$. We suppose that any flow stored at an intermediate node is permanently stored at that node, ensuring that the fuzzy flow remains feasible throughout the process. In this paper, the original network with excess flows at storage nodes is called excess flow network and denoted by $\tilde{N}_\epsilon$.

If it is not allowed to store excess flows at any intermediate node, this problem simply becomes FMDFP. In FMDFP, the value of $\sum_{j\in V\setminus\{s,d\}} \tilde{x}_j(T)$ is zero. So, the constraint (9) is not required, but attention must be given to the flow conservation at each intermediate node at the end of the time T. Before going to discuss the solution strategy for FMDFIS, let's move through some important terminologies that are used in the solution procedure.

3.2 Prioritization of Intermediate Nodes

While solving FMDFIS, we need to calculate the excess flows at each intermediate node with respect to their storage capacity. The excess flows at storage nodes depend on the priority of the storage nodes. So, we first aim to prioritize all the storage nodes, including the sink node. Prioritization of storage nodes depends upon the nature of the problem (Pyakurel and Dempe, 2021; Dhamala et al., 2024; Gerasimenko, 2024; Khanal et al., 2025). From the application point of view (especially in evacuation planning), it is better to prioritize the nodes based on the distances (Pyakurel and Dempe, 2021). Calculating the shortest distances from the source node by using Dijkstra's algorithm, we can set the priority order $V_p = \{w_0 \prec w_1 \prec w_2 \prec \cdots \prec w_p\}$, where sink node $d = w_0$ is always first in priority and the node with longest distance has high priority, keeping in mind that the sink is the most appropriate place to send the evacuees as much as possible. In this paper, the priority order of storage nodes is taken as given input.

4. PROPOSED ALGORITHM FOR FMDFIS

In this section, we propose an efficient solution technique for FMDFIS based on the idea of Khanal et al. (2025) on the maximum dynamic network flow problem with intermediate storage in non-fuzzy environment. Our solution strategy is based on the idea of finding augmenting path by using breadth-first-search (BFS). We divide our complete solution technique for FMDFIS into three parts. First, we develop an algorithm (Routine I), which solves this problem without time (static). Then we will have Routine II to construct the flow balancing paths. Finally, flow-balancing paths are used in the formula of generalized temporally repeated flow developed by Khanal et al. (2025).

4.1 Algorithm for Excess Flow Network

We design the algorithm, Routine-I, to compute the excess flow network $\tilde{N}_\epsilon$, where we incorporate the maximum flow algorithm and the minimum cost flow algorithm. Here, we apply the maximum flow algorithm in static form based on the flow augmenting path (see Kumar and Kaur, 2012; Bozhenyuk et al., 2017). Then we use the successive shortest path algorithm as a minimum cost flow algorithm to distribute the flow in the network with minimum cost, where arc-travel time is taken as arc-cost per unit flow. Since we already have maximum flow distribution, minimum-cost flow distribution can be done by cancelling cycle of negative cost in the network (see Ahuja et al., 1993).

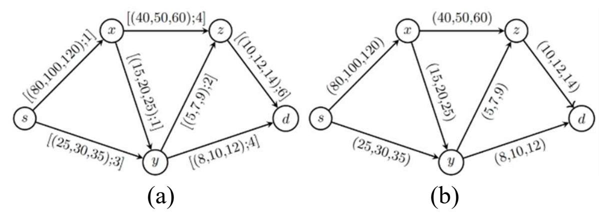


Fig. 3. Network with triangular fuzzy arc capacities and crisp transit time. (a) Dynamic network (with time) (b) Static network (without time)

Algorithm 1 (Routine-I)

Input: Dynamic network $\tilde{N} = (V, A, s, d, \tilde{u}, \tilde{U}, t)$ with V_p (prioritized storage nodes);

1. Set residual network $\tilde{N}(\tilde{x}) = \tilde{N}$ and zero flows
2. Set excess flow network $\tilde{N}_\epsilon = \tilde{N}$;
3. For $k = 0,1,\cdots,p$ do;
4. Find $\tilde{v}(w_k)$, maximum flow at w_k in $\tilde{N}(\tilde{x})$ by applying static maximum flow algorithm;

---

5. Add $\tilde{v}(w_k)$ as excess flow to the node w_k in $\tilde{N}_\epsilon$, say $\tilde{\epsilon}(w_k)$;
6. If $\tilde{\epsilon}(w_k) > \tilde{0}$ do;
7. Apply minimum-cost flow algorithm with arc-cost t_{ij};
8. Update the residual network $\tilde{N}(\tilde{x})$;
   Output: Fuzzy excess flow network $\tilde{N}_\epsilon$.

Routine-I begins with a set V_p in dynamic network $\tilde{N}$. Initially, the algorithm sets the excess flow network $\tilde{N}_\epsilon$ zero excess flows at storage nodes. The maximum flow value for each w_k in the residual network $\tilde{N}(\tilde{x})$ is stored as the excess flow at w_k, denoted as $\tilde{\epsilon}(w_k)$ and distribute the flow in minimum cost by updating the residual network at each iteration. After processing all storage nodes according to priority order, we get the final $\tilde{N}_\epsilon$.

To validate the algorithm, Routine-I, we consider one numerical example as follows.

Example 1. Consider the dynamic network in Fig. 3(a) with triangular fuzzy arc-capacities and crisp transit time, and set of storage nodes with priority order $d \prec z \prec y \prec x$. We first find the static maximum flow to each storage node starting from sink d as follows.

Using BFS algorithm in Fig. 3(b) we get the augmenting path $P_1: s \rightarrow y \rightarrow d$. For path flow $\tilde{\delta}(P_1)$, we compare the ranks of the arc capacities of the path P_1 using Equation (3) as follows

$$\min\left(R(\tilde{r}_{sy}), R(\tilde{r}_{yd})\right)$$

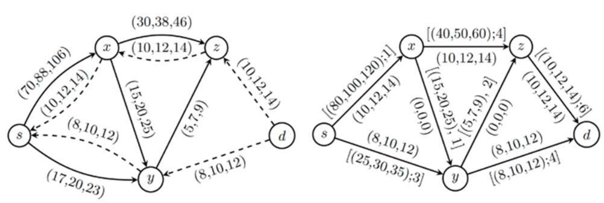
$$= \min\left(\frac{1}{4}(25 + 2 \times 30 + 35), \frac{1}{4}(8 + 2 \times 10 + 12)\right)$$

$$= \min(30, 10) = R(\tilde{r}_{yd})$$

$$\therefore \tilde{\delta}(P_1) = \min(\tilde{r}_{sy}, \tilde{r}_{yd}) = \tilde{r}_{yd} = (8,10,12)$$

After updating flow and arc-capacities along the path, we get another residual network, where we have another shortest path $P_2: s \rightarrow x \rightarrow z \rightarrow d$.

As above we can get $\tilde{\delta}(P_2) = \min(\tilde{r}_{sx}, \tilde{r}_{xz}, \tilde{r}_{zd}) = (10,12,14)$.

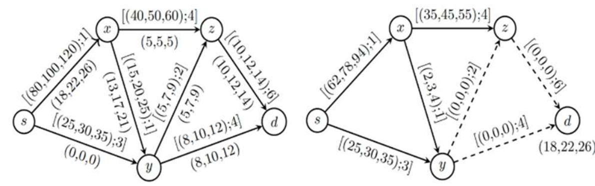


(a) (b)

Fig. 4. (a) Residual network after maximum flow at d. (b) Maximum flow distribution in $\tilde{N}(\tilde{x})$.

There is no fuzzy augmenting path from s to d in the resulting network (see Fig. 4(a)). So, the maximum flow and hence excess flow at d is $(18, 22, 26)$. Now, we set $(18, 22, 26)$ as the demand at node d and supply at source s in Fig. 3 (a), and applying minimum-cost flow algorithm to distribute the flow in network to minimize the total cost we obtain the residual network as in Fig. 5(a). After reducing the minimum cost flows in Fig. 5(a), we get the updated network as in Fig. 5(b).

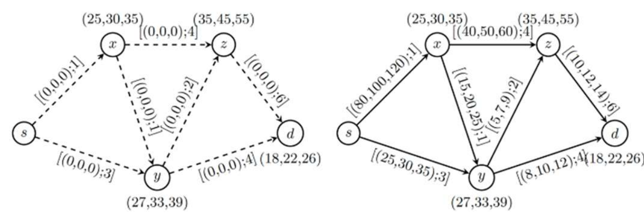
Again, applying maximum flow algorithm in Fig. 5(b) for node z we get the excess flow at z as $(35, 45, 55)$ and using minimum-cost flow algorithm for demand value $(35, 45, 55)$ at z.



(a) (b)

Fig. 5. Flow distribution in $\tilde{N}$ after maximum flow at d. (a) Minimum-cost flow distribution. (b) Residual network.

In the similar manner as above, we obtain the excess flow at each storage node by performing the iterations of Routine-I. Then the final residual network after all iterations of Routine-I is shown in Fig. 6(a). Finally, we get the excess flow network as in Fig. 6(b) by adding excess flows at each storage nodes of $\tilde{N}$.



(a) (b)

Fig. 6. (a) Final residual network. (b) Excess flow network after Routine-I.

4.2 Algorithm for Finding Flow-Balancing Paths

Here, we design Routine-II (Algorithm 2) to decompose the excess flows of each storage nodes into flow balancing paths in $\tilde{N}_\epsilon$. We use the following rule to construct the flow balancing path in excess flow network.

Rule of constructing flow-balancing path:

We decompose the fuzzy flow along a minimum-cost path P_{w_k} from source node s to a storage node w_k in the excess flow network which includes stored fuzzy flow at intermediate nodes. We construct the flow-balancing path $\bar{P}_{w_k}$ along P_{w_k} following the steps by Khanal et al. (2025) given below.

- Calculate the path flow $\tilde{\delta}(P_{w_k})$ of P_{w_k}.
- Balance the flow $\tilde{\delta}(P_{w_k})$ from last node w_k of the path back to the source node s ensuring that the sum of inflow and excess flow equals the outflow at each intermediate node in the residual network of excess flow network.

Algorithm 2 (Routine-II)

Input: $\widetilde{N}_\epsilon$ and V_p.

1. Set $\bar{\boldsymbol{P}} = \emptyset$ (set of flow balancing paths)
2. Set $\widetilde{N}'_\epsilon = \widetilde{N}_\epsilon$;
3. For $k = 0,1,\cdots,p$ do;
4. While $\tilde{\epsilon}(w_k) > \tilde{0}$ do;
5. Find the minimum-cost path P_{w_k} (from s to w_k) in $\widetilde{N}'_\epsilon$ using Dijkstra's algorithm;
6. Construct flow balancing path $\bar{P}_{w_k}$ along P_{w_k};
7. Add $\bar{P}_{w_k}$ in $\bar{\boldsymbol{P}}$;
8. Update the network $\widetilde{N}'_\epsilon$ by reducing node excesses and arc flows along the path $\bar{P}_{w_k}$;

Output: Set of flow balancing paths, $\bar{\boldsymbol{P}}$.

The algorithm, Routine-II, begins with fuzzy excess flow network and fixed priority order of storage nodes. It then iterates over a sequence of nodes w_k for $k = 0,1,\cdots,p$. For each storage node w_k with positive excess flow $\tilde{\epsilon}(w_k)$, it finds the minimum-cost path P_{w_k} in the excess flow network. The flow balancing path $\bar{P}_{w_k}$ of P_{w_k} is constructed based on the rules outlined as above. This path $\bar{P}_{w_k}$ is added to the set $\bar{\boldsymbol{P}}$ along with its associated path flows $\tilde{\delta}(P_{w_k})$, node excesses $\tilde{\epsilon}$, and the cost from the source $t(P_{w_k})$. The network $\widetilde{N}_\epsilon$ is then updated by reducing the node excesses and arc flows along the path $\bar{P}_{w_k}$. The algorithm provides the set of flow balancing paths $\bar{P}_{w_k}$ with the minimum travel time $t(\bar{P}_{w_k})$, the excess $\tilde{\epsilon}$ of intermediate nodes, and the maximum flow $\tilde{\delta}$ from the source to each node along the paths.

Example 2 (Numerical validity of Routine-II). Consider the excess flow network in Fig. 6(b) with priority order set $V_p = \{d \prec z \prec y \prec x\}$ of storage nodes. We apply Routine-II to get the flow balancing paths as in Figs. 7(a), 7(c) and 8(b).

Flow balancing paths for d

Using Dijkstra's algorithm, we find the minimum-cost path $P_d^1 : s \to x \to y \to d$ with path flow $\tilde{\delta}(P_d^1) = (8,10,12)$ and cost $t(P_d^1) = 6$. Then the flow balancing path of P_d^1 is constructed as in Fig. 7(a). After reducing the flow values along the path in Fig. 7(a), we get the updated network as in Fig. 7(b).

Again, we find another shortest s-d path P_d^2. The flow of the path P_d^2 is $(10,12,14)$ and we obtain the flow balancing path $\bar{P}_d^2$ as in Fig. 7(c). Then we update the network in Fig. 7(a) by reducing the excess flow and arc flows from flow balancing path, we get the updated network as in Fig. 8(a), which has no flow augmenting path from s to d. Now, we go for another node z as per priority order.

Flow balancing paths for z: constructing flow balancing path of $\bar{P}_z^1$ as in Fig. 8(b). After updating the network of Fig. 8(a) by reducing the excess flow and arc flows from flow balancing path $\bar{P}_z^1$, we get the updated network as in Fig. 8(c). There is no any augmenting path from source to any other nodes (d, z, y, x), so Routine- II terminates with the flow balancing paths as listed in Table 2.

Theorem 1. Algorithm 2 correctly balances the flow in

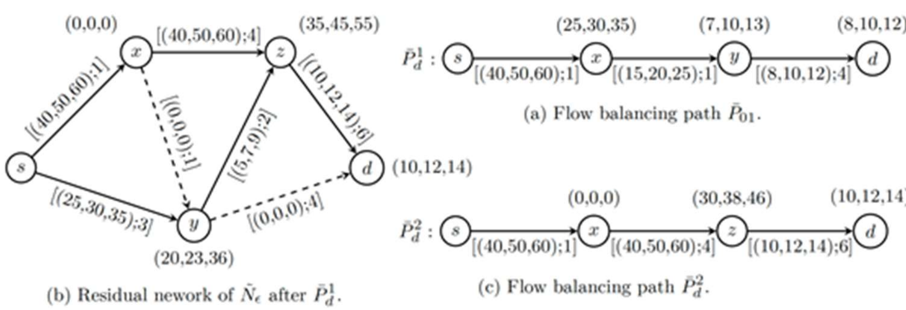


Fig. 7. Residual excess flow network and flow balancing paths.

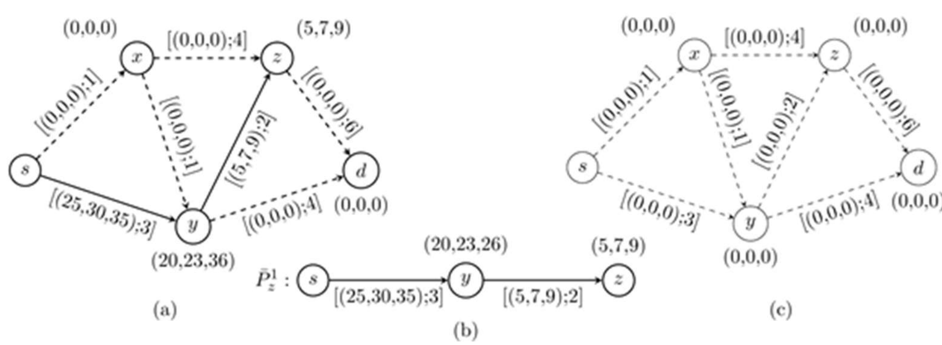


Fig. 8. Residual excess flow network and flow balancing paths.

Table 2. List of flow balancing paths

FBP	$\tilde{\epsilon}(x)$ | $\tilde{\delta}(x)$ | $t(\bar{P}_{[s\to x]})$	$\tilde{\epsilon}(y)$ | $\tilde{\delta}(y)$ | $t(\bar{P}_{[s\to y]})$	$\tilde{\epsilon}(z)$ | $\tilde{\delta}(z)$ | $t(\bar{P}_{[s\to z]})$	$\tilde{\epsilon}(d)$ | $t(\bar{P}_{[s\to d]})$
$\bar{P}_d^1$	(20,30,35) | (40,50,60) | 1	(7,10,13) | (15,20,25) | 2	-	(8,10,12) | 6
$\bar{P}_d^2$	(0,0,0) | (40,50,60) | 1	-	(30,38,46) | (40,50,60) | 4	(10,12,14) | 11
$\bar{P}_z^1$	-	(20,23,26) | (25,30,35) | 3	(5,7,9) | (5,7,9) | 5	

the excess flow network $\tilde{N}_\epsilon$ while minimizing the travel time t. Specifically, the algorithm terminates with a set of flow balancing paths $\bar{\boldsymbol{P}}$ that ensure the excess flow at each node is reduced to zero.

Proof. To prove the correctness of Algorithm 2, we demonstrate the reasons into the following steps.

The algorithm iterates over nodes w_k for $k = 0, 1, \cdots, p$ and processes each node with positive excess flow $\tilde{\epsilon}(w_k) > \tilde{0}$. In each iteration, the algorithm selects a minimum-cost path P_{w_k} and constructs a flow balancing path $\bar{P}_{w_k}$. The network $\tilde{N}_\epsilon$ is updated by reducing the node excesses and arc flows along $\bar{P}_{w_k}$. Since the excess flow is reduced in each iteration and the network has a finite number of nodes and arcs, the algorithm must terminate with a set of flow balancing path(s) after a finite number of steps.

By the rules outlined above, the flow balancing paths ensure that the excess flow is redistributed in a way that reduces the excess at each node to zero. This is achieved by iteratively processing nodes with positive excess and updating the network accordingly. Since the algorithm terminates after a finite number of steps, correctly balances the flow by reducing excess flow at each node to zero, and minimizes the travel time t, we conclude that Algorithm 2 is correct. □

By following this process, Algorithm 2 ensures that the static excess flow at each prioritized node is both feasible and optimal. The use of successive shortest paths and iterative balancing guarantees that the flow is distributed efficiently while respecting capacity constraints. This makes the algorithm a reliable method for computing optimal static excess flow in networks with prioritized nodes.

4.3 Fuzzy Temporally Repeated Flow

TRF is a concept in dynamic network flow where a static flow pattern is repeated across multiple time steps to model movement in a time-expanded network with fixed arc travel times (Ford and Fulkerson, 1962). Based on the generalized TRF of Khanal et al. (2025), we use fuzzy TRF formula to compute total fuzzy dynamic flow, where fuzzy flow values are distributed over discrete time intervals along paths, while respecting arc transit times. Our focus is on maximizing the fuzzy flow from the source over the time horizon T, allowing excess fuzzy flow $\tilde{\epsilon}(w_k)$ to be permanently held at storage nodes w_k when it cannot advance further via temporal repetition.

We use Equation (10), based on the generalized TRF formula provided by Khanal (2024), to calculate the total fuzzy maximum dynamic flow with intermediate storage. The proposed TRF formula, Equation (10), allows us to calculate the maximum dynamic flow at each priority node within a given time horizon T. The formula considers the flow along sub-paths from the source s to a storage node w_k within the path $\bar{P}$. It accounts for the travel time t and the excess flow $\tilde{\epsilon}$ at each node, ensuring that the flow is dynamically adjusted based on the time constraints and the structure of the network.

Fuzzy TRF for a storage node w_k:

Let $\bar{P}_{[s\to w_k]}$ be a sub-path from source to a node w_k along the path $\bar{P}$ (i.e. $\bar{P}_{[s\to w_k]} \subseteq \bar{P}$). Then the total fuzzy flow at w_k can be calculated by Equation (10):

$$\tilde{v}(w_k) = \sum_{\substack{w_k\in\bar{P}\\ \bar{P}\in\bar{\boldsymbol{P}}}} \Big[\big(T - t(\bar{P}_{[s\to w_l]}) + 1\big)\tilde{\epsilon}(w_k) + \Big(t(\bar{P}_{[s\to w_l]}) - t(\bar{P}_{[s\to w_k]})\Big)\ \tilde{\delta}(w_k)\Big] \quad (10)$$

where $(w_k, w_l) \in \bar{P}$ (i.e. $\bar{P}_{[s\to w_k]} \subset \bar{P}_{[s\to w_l]} \subseteq \bar{P}$). But if w_k is the last node of the path $\bar{P}$ we use $t(\bar{P}_{[s\to w_k]})$ instead of $t(\bar{P}_{[s\to w_l]})$.

Example 3 (Temporally repeated flow at sink node and intermediate nodes). Here, we take the time horizon, $T = 13$, for temporally repeated flow so that all the possible paths can be used in the fuzzy TRF formula to calculate the total dynamic flow for each storage node according to their priority order. First, we calculate the TRF value at sink node d considering the excess flow as path flow $\tilde{\delta}$.

At node d:

$\tilde{v}(d) = [(13-6+1)(8,10,12) + (6-6)(8,10,12)]$
$+ [(13-11+1)(10,12,14) + (11-11)(10,12,14)]$
$= [(8)(8,10,12) + (0)(8,10,12)] + [(3)(10,12,14)$
$+(0)(10,12,14)]$
$= (64,80,96) + (0,0,0) + (30,36,42) + (0,0,0)$
$= (94,116,138)$

At node z:

$\tilde{v}(z) =$
$[(13-11+1)(30,38,46) + (11-4)(40,50,60)]$
$+[(13-5+1)(5,7,9) + (5-5)(5,7,9)]$
$= (3)(30,38,46) + (7)(40,50,60) + (9)(5,7,9)$
$+(0)(0,0,0)$
$= (90,114,138) + (280,350,420) + (45,63,81)$
$+(0,0,0) = (415,527,639)$

At node y:

$\tilde{v}(y) = [(13-6+1)(7,10,13) + (6-2)(15,20,25)]$
$+ [(13-5+1)(20,23,26) + (5-3)(25,30,35)]$
$= [(8)(7,10,13) + (4)(15,20,25)] + [(9)(20,23,26)$
$+(2)(25,30,35)]$

$= (56,80,104) + (60,80,100) + (180,207,234)$
$+(50,60,70) = (346,427,508)$
At node x:
$\tilde{v}(x) = [(13-2+1)(25,30,35) + (2-1)(40,50,60)]$
$+[(13-4+1)(0,0,0) + (4-2)(40,50,60)]$
$= [(20)(25,30,35) + (1)(40,50,60)] + [(10)(0,0,0)$
$+(2)(40,50,60)]$
$= (300,360,420) + (40,50,60) + (0,0,0)$
$+(80,100,120) = (420,510,600)$

5. FUZZY EARLIEST ARRIVAL FLOW WITH INTERMEDIATE STORAGE

Most real-world problems involving flow optimization under uncertainty, such as evacuation planning, supply chain management, and energy distribution, benefit from intermediate storage that enhances flexibility, while fuzzy sets effectively model the unpredictability of capacities, demands, or travel times. For example, the model maximizes the number of evacuees reaching a safe place at every time step, utilizing shelters (intermediate storage) to temporarily accommodate people when road capacities (arcs) are uncertain due to damage or congestion (Gerasimenko, 2024). In this section, we introduce the FEAFIS problem in a series-parallel network based on the crisp earliest arrival flow (EAF) with intermediate storage. The objective of FEAFIS is to maximize the flow from source to sink at every time step $\sigma \in \boldsymbol{T}$ in a dynamic network. EAF with intermediate storage can be considered as a multi-objective problem (Pyakurel and Dempe, 2021). There are several solution strategies to solve this problem in different cases. For example, Minieka (1973) developed the lexicographic approach by showing that EAF in single-sink networks are equivalent to lexicographically maximum flows in an expanded network. TRF cannot be applied to general networks because of the use of non-standard chain decomposition. So, it does not guarantee the optimal solution of EAF in a general network with flow conservation constraints even without intermediate storage (Khanal, 2024).

The objective value of FEAFIS is evaluated at all time points, ensuring fuzzy flow is prioritized to reach d as early as possible. The fuzzy earliest arrival objective ensures efficient resource allocation at every time step more realistically. FEAFIS is defined as follows.

$$\text{Maximize } \tilde{v}(\sigma) = \sum_{i:(i,d)\in A} \sum_{\theta=t_{\{id\}}}^{\sigma} \tilde{x}_{id}(\theta - t_{id}) + \sum_{j\in V\setminus\{s,d\}} \tilde{x}_j(\sigma) \;\forall\, \sigma \in \boldsymbol{T} \quad (11)$$

$$\text{Subject to } \sum_{j:(s,j)\in A} \sum_{\theta=0}^{\sigma} \tilde{x}_{sj}(\theta) = \tilde{v}(\sigma) \;\forall\, \sigma \in \boldsymbol{T} \quad (12)$$

$$\sum_{k:(j,k)\in A} \sum_{\theta=t_{jk}}^{\sigma} \tilde{x}_{jk}(\theta - t_{jk}) - \sum_{i:(i,j)\in A} \sum_{\theta=0}^{\sigma} \tilde{x}_{ij}(\theta) = \tilde{x}_j(\sigma) \;\forall\, j \in V \setminus \{s,d\} \;\forall\, \sigma \in \boldsymbol{T} \quad (13)$$

$$\tilde{0} \le \tilde{x}_{ij}(\sigma) \le \tilde{u}_{ij} \forall\, (i,j) \in A \;\forall \sigma \in \boldsymbol{T} \quad (14)$$

$$\tilde{0} \le \tilde{x}_j(\sigma) \le \tilde{U}_j \;\forall j \in V \setminus \{s,d\} \;\forall \sigma \in \boldsymbol{T} \quad (15)$$

The objective function (11) is to maximize the fuzzy flow $\tilde{v}(\sigma)$ arriving at the destination d by every time $\sigma \in \boldsymbol{T}$, while accounting for stored flow in intermediate nodes. Constraints (12) ensure that, for every time σ, the total flow leaving the source s equals the flow reaching d plus stored flow in intermediate nodes. Constraints (13) ensure precise accounting of storage at each intermediate node for all time steps. Constraints (14) and (15) ensure that flow on each arc (i,j) at time θ must respect fuzzy capacity bounds $\tilde{u}_{ij}$ and stored flow at any node $j \in V \setminus \{s,d\}$ must remain within fuzzy storage limits $\tilde{U}_j$.

A series-parallel graph is an acyclic graph that can be constructed through two fundamental operations: series composition and parallel composition.

- The series composition $S(S_1, S_2)$ is obtained by identifying the sink t_1 with the source s_2, resulting in a series-parallel graph with source s_1 and sink t_2.
- The parallel composition $S(S_1, S_2)$ is obtained by merging the sources s_1 with s_2 and the sinks t_1 with t_2, yielding a series-parallel graph with source $s_1(= s_2)$ and sink $t_1(= t_2)$.

In a series-parallel network, the optimal solution of EAFIS is possible by TRF, but finding EAFIS using TRF in a non-series-parallel network is not possible (Khanal, 2024). So, we consider a series-parallel network to apply the TRF formula for fuzzy FEAFIS with prioritized intermediate storage in a dynamic network.

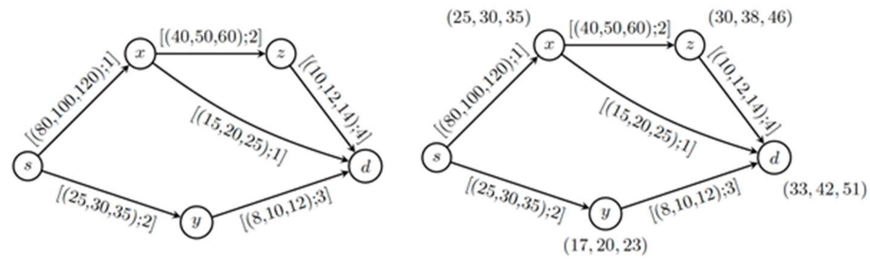
Example 3. Consider a series-parallel dynamic network as in Fig. 9(a). Set the priority order of the storage nodes as $d \prec z \prec y \prec x$. Applying Routine-II in Fig. 9(b) to get the following flow balancing paths as in Fig. (10). Using the fuzzy TRF formula in Equation (10) for intermediate storage, we get the earliest arrival flow at each time step $\sigma \in \{0,1,2,3,4,5,6,7\}$ as shown in Table 2.

Algorithm 3 TRF Method for FEAFIS

Input: Series-parallel network $\tilde{N} = (V, A, s, d, \tilde{u}, \tilde{U}, \tau, T)$ and V_p.
1. Obtain excess flow network $\tilde{N}_\epsilon$ by applying Routine-I with V_p;
2. Using Routine-II determine the flow balancing paths;
3. Apply TRF for T formula to calculate the total dynamic
4. flow at each storage node;
5. Output: Fuzzy EAF with intermediate storage.

Note that if the travel time of the flow balancing path is greater than the time step σ, then we will not consider the path for those nodes for which the travel time is greater than σ. But it can be considered for the nodes in the path whose travel time is less than or equal to σ.

For instance, when $\sigma = 5$. There are only two paths $\bar{P}_d^1$ and $\bar{P}_d^2$ from s to d. But the path $\bar{P}_d^3$ should not be



(a) (b)

Fig. 9. A two-terminal series parallel network with $[\tilde{u}_{ij};\tilde{t}_{ij}]$ and excess flow network.

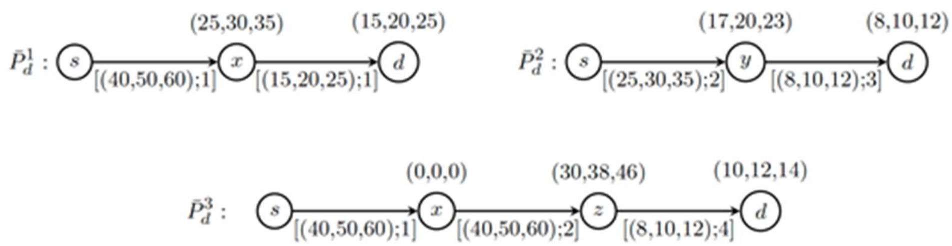


Fig. 10. Fuzzy flow balancing paths $\bar{P}_d^1, \bar{P}_d^2$ and $\bar{P}_d^3$ after Routine-II

Table 3 Calculation of earliest arrival flow with intermediate storage for $\boldsymbol{T=7}$

Storage node	$\sigma=0$	$\sigma=1$	$\sigma=2$	$\sigma=3$	$\sigma=4$	$\sigma=5$	$\sigma=6$	$\sigma=7$
d	(0,0,0)	(0,0,0)	(15,20,25)	(30,40,50)	(45,60,75)	(68,90,112)	(91,120 ,149)	(124,162, 200)
z	(0,0,0)	(0,0,0)	(0,0,0)	(40,50,60)	(80,100 ,120)	(120,150 ,180)	(160,200 ,240)	(190,238, 286)
y	(0,0,0)	(0,0,0)	(0,0,0)	(75,90,115)	(75,90 ,105)	(92,110 ,128)	(109,130 ,151)	(126,150, 174)
x	(0,0,0)	(80,100 ,120)	(130,160 ,190)	(195,240 ,295)	(195,240 ,285)	(220,270 ,320)	(245,300 ,355)	(270,330, 390)
Total flow	(0,0,0)	(80,100 ,120)	(145,180 ,215)	(350,420 ,520)	(395,490 ,585)	(500,620 ,740)	(605,750 ,895)	(710,880,105 0)

considered as $t(\bar{P}_d^2)=7$. In this case, we consider the path $\bar{P}_d^3$ only from s to z with capacity (40,50,60) and travel time $t(\bar{P}_d^3)=3$.

At node d:

$$\tilde{v}_5(d) = [(5-2+1)(15,20,25) + (5-5+1)(8,10,12)]$$
$$= [(4)(15,20,25) + (1)(8,10,12)]$$
$$= (60,80,100) + (8,10,12) = (68,90,112)$$

At node z:

$$\tilde{v}_5(z) = [(5-3+1)(40,50,60)](120,150,180)$$

At node y:

$$\tilde{v}_5(y) = [(5-5+1)(17,20,23) + (5-2)(25,30,35)]$$
$$= [(1)(17,20,23) + (3)(25,30,35)]$$
$$= (17,20,23) + (75,90,105) = (92,110,128)$$

At node x:

$$\tilde{v}_5(x) = [(5-2+1)(25,30,35) + (2-1)(40,50,60)] + [(3-1)(40,50,60)]$$
$$= [(4)(25,30,35) + (1)(40,50,60)] + [(2)(40,50,60)]$$
$$= (100,120,140) + (40,50+60) + (80,100,120)$$
$$= (220,270,320).$$

Therefore, the fuzzy maximum flow at each time step $\sigma=0,1,2,3,4,5,6,7$ are given in Table 3.

Here, we consider a series-parallel network that does not contain any cycle. So, each minimum cost path is used as a flow balancing path with intermediate excess. This result leads to the following theorem.

Theorem 2. For the series-parallel graph, Algorithm 3 provides an optimal fuzzy solution to EAF with intermediate storage using the fuzzy TRF formula in polynomial time.

The proof of this theorem is directly followed by a crisp model in (Khanal et al., 2025).

6. RESULTS AND DISCUSSION

The results obtained from our proposed model demonstrate that FMFIS address uncertainty in network optimization problems by allowing the arc capacities and storage capacities in fuzzy environment. We summarize the results of the above numerical experiments as follows:

I. The total maximum static flow with intermediate storage (based on the priority order of storage nodes) of the network in Fig. 3 is (18, 22, 26) + (35, 45, 55) + (27, 33, 39) + (25, 30, 35) = (105, 130, 155). This implies that the total stored static flow value ranges between 105 and 155 with the core value is 130.
II. The total maximum dynamic flow with intermediate storage of the network in Fig. 3 for T = 13 is (94, 116, 138) + (415, 527, 639) + (346, 427, 508) + (420, 510, 600) = (1275, 1580, 1885). The total flow value that can be stored at nodes is between 1275 and 1885 with core value 1580.
III. The total earliest arrival flow with intermediate storage of the series-parallel network in Fig. 9 (a) for T = 7 is (710, 880,1 050). This implies that the total stored earliest arrival flow value ranges between 710 and 1050 and the core value is 880. It also represents the total maximum dynamic flow of the network for T = 7.

We formulated the network flow problem with intermediate storage in a hybrid framework by incorporating fuzzy arc capacities with crisp transit time. We considered the FMDFIS problem in a single-source, single-sink directed network with fuzzy capacities and crisp transit times, whereas the FEAFIS has been considered in a series-parallel network. Based on the temporally repeated flow model and the flow balancing path, efficient algorithms for the solution of the problems FMDFIS and FEAFIS have been developed as discussed in Sections 4 and 5, respectively. The theoretical foundation of the proposed algorithms has been validated through numerical examples. We have considered discrete-time dynamic networks to calculate maximum dynamic flow, where arc transit times remain constant over the given time period. Algorithmic correctness has been proved in Theorem 1, further supported by numerical illustrations validating their effectiveness. In Example 3, we have calculated the accumulated flows at each storage node according to their priority.

This paper focuses on implementing the recently developed efficient idea of generalized TRF approach in the crisp network flow model within fuzzy framework. Most of the literature used defuzzification methods to apply classical approach for solving FMFP and its variants (Chanas and Kolodzijczyk, 1986; Hernandes et al., 2007; Gerasimenko, 2024). In contrast, we employed fuzzy operations without defuzzification, following Kumar and Kaur (2012), who used nonstandard subtraction operation to avoid negative flow in the calculations. We also used ranking method to compare the fuzzy numbers while computing effective path flows. We extended the study of Khanal et al. (2025) on TRF for maximum dynamic flow problem with intermediate storage to a fuzzy environment. The whole solution procedure is divided into three parts: constructing an excess flow network by solving FMDFIS in static form by using Routine I, finding flow balancing paths in excess flow network and calculating total dynamic flow by using fuzzy TRF formula. The proposed method robustly determines the maximum dynamic flow at each priority node within a time horizon under fuzzy conditions. We were able to calculate maximum dynamic flow at each prioritized node using the flow balancing path approach of Khanal et al. (2025) in a fuzzy environment. This idea is more efficient than the construction of the time-expanded network method. As suggested in literature related to intermediate storage, we strongly suggest that our solution idea is more reliable to apply in uncertain evacuation planning where evacuees should be allowed to be kept in intermediate nodes. The theoretical comparison of the literature with our contribution based on the variants of problem and solution procedures is presented in Table 4.

Table 4. Summary of problems and solution procedures

Literature	Problem	Solution procedure
Chanas (1987)	Maximum flow in a network with fuzzy arc capacities	Generalization of max-flow min-cut theorem, Dijkstra's shortest path finding approach.
Diamond (2001)	Maximum flow with interval-valued version of maximum flow.	Max-flow min-cut theorem, algorithm based on Edmonds approach.
Kumar and Kaur (2012)	Fuzzy maximum flow	Augmenting path approach, ranking method and nonstandard subtraction operation.
Pyakurel and Dempe (2020)	Maximum static and dynamic flow with intermediate storage.	Distance based priority order, contraflow, augmenting path
Pyakurel and Dempe (2021)	Earliest arrival flow with intermediate storage.	Distance based priority order, contraflow, augmenting path
Khanal et al. (2025)	Maximum dynamic flow with intermediate storage	Flow balancing path, TRF.
Gerasimneko (2024)	Dynamic flow with intermediate storage in fuzzy environment.	Defuzzification, incomplete intuitionistic fuzzy preference ordering, contraflow.
This paper	FMDFIS and FEAFIS	Flow balancing path, distance based priority order, augmenting path, TRF.

7. CONCLUSION

Classical dynamic flow models are deterministic approaches used to address the optimal movement of flow through a network over time. Later, the realization of the uncertainty in the real-world scenario of network flow problems led to the fuzzy approaches. Now, the concept of intermediate storage has been incorporated into the fuzzy dynamic flow models which allow holding flows at intermediate nodes (either temporally or permanently depending upon the problem type). In literature, the fuzzy dynamic flow models with intermediate storage were studied in a time-expanded network with time-dependent attributes, which was not efficient as compared to the TRF method.

We not only extended the classical approach but also improved the practical application point of view of dynamic flow models with intermediate storage in uncertain emergency situations. This is particularly effective in time-sensitive scenarios like evacuation planning, where the danger zone is set as the source node, the safe zone as the sink, and intermediate nodes, representing relatively safe areas, are prioritized based on safety for efficient evacuation.

DECLARATION OF COMPETING INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

ACKNOWLEDGMENT

The first author would like to acknowledge University Grant Commission (UGC), Nepal for the Ph.D. Fellowship (Award No.: PhD-78/79-S & T-11) and the Alexander von Humboldt Foundation for the partial support under the Research Group Linkage Program of the Foundation between KIT-IOR-DOL, Germany, and the Central Department of Mathematics at Tribhuvan University, Nepal (2024).

REFERENCES

Adhikari, M.C., Dhamala, T.N. 2025. Generalized maximum flow with excess storage on series parallel lossy network. Journal of the Operations Research Society of China.

Ahuja, R., Magnanti, T., Orlin, J. 1993. “Network Flows: Theory, Algorithms and Applications”. Prentice Hall.

Akrami, A. 2025. Investigation of some fuzzy optimization problems with fuzzy genetic algorithms. Current Applied Sciences, 3(2), 83–88.

Almutairi, H., Zhang, N. 2024. A survey on routing solutions for low-power and lossy networks: toward a reliable path-finding approach. Network, 4(1), 1–32.

Bozhenyuk, A.V., Gerasimenko, E.M., Kacprzyk, J., Rozenberg, I.N. 2017. Flows in networks under fuzzy conditions. Studies in fuzziness and soft computing 346, Springer International Publishing Switzerland.

Chanas, S. 1987. Fuzzy optimization in networks. In: Kacprzyk, J., Orlovski, S.A. (eds) Optimization Models Using Fuzzy Sets and Possibility Theory. Theory and Decision Library, 4, 303–327.

Chanas, S., Kolodziejczyk, W. 1982. Maximum flows in a network with fuzzy arc capacities. Fuzzy Sets and Systems, 8, 165–173.

Chanas, S., Kolodziejczyk, W. 1986. Integer flows in a network with fuzzy arc capacities. Networks, 16, 17–31.

Dhamala, T.N. 2015. A survey on models and algorithms for discrete evacuation planning network problems. Journal of Industrial and Management Optimization, 11(1), 265–289.

Dhamala, T.N., Adhikari, M.C., Khanal, D.P., Pyakurel, U. 2024. Generalized maximum flow over time with intermediate storage. Annals of Operations Research. 335, 111–134.

Dhamala, T.N., Khanal, D.P., Dempe, S. 2025. Network restructuring for dynamic flow improvement. Annals of Operations Research, 347, 1213-1247.

Diamond, P.A. 2001. Fuzzy max-flow min-cut theorem. Fuzzy Sets and Systems, 119, 139–148.

Dubois, D., Prade, H. 1978. Operations on fuzzy numbers. International Journal of Systems Science, 9, 613–626.

Ford, L.R., Fulkerson, D.R. 1962. “Flows in networks”. Princeton University Press.

Gerasimenko, E. 2024. Dynamic flow algorithm with intermediate storage for emergency evacuation allowing lane reversal based on incomplete intuitionistic fuzzy preference relation. Open Transportation Journal, 18: e18744478281682.

Ghanbari, R., Ghorbani-Moghadam, K., Mahdavi-Amiri, N., De Baets, B. 2020. Fuzzy linear programming problems: models and solutions. Soft Computing, 24, 10043–10073.

Han, S., Peng, Z., Wang, S. 2014. The maximum flow problem of uncertain network. Information Sciences, 265, 167–175.

Hernandes, F., Lamata, M.T., Takahashi, M.T., Yamakami, A., Verdegay, J.L. 2007. An algorithm for the fuzzy maximum flow problem. IEEE International Fuzzy Systems Conference, pp. 1–6.

Kaufmann, A., Gupta, M.M. 1985. “Introduction to fuzzy arithmetic theory and applications”. Van Nostrand Reinhold; New York.

Khan, A.A., Laghari, A.A., Elmannai, H., Shaikh, A.A., Bourouis, S., Hadjouni, M., Alroobaea, R. 2023. Gan-iotvs: A novel internet of multimedia things-enabled video streaming compression model using gan and fuzzy logic. IEEE Sensors Journal, 23, 29434–29441.

Khanal, D.P. 2024. “Multi-commodity dynamic flow problems with intermediate storage and varying transit times”. Ph.D. Thesis.

Khanal, D.P., Pyakurel, U., Dempe, S. 2025. Temporally repeated maximum dynamic flow with intermediate storage. Operations Research Forum, 6, 101.

Kim, K., Roush, F. 1982. Fuzzy flows on network. Fuzzy Sets and Systems, 8(1), 35–38.

Klir, G. J., Yuan, B. 1995. "Fuzzy sets and fuzzy logic: Theory and application". Prentice Hall PTR, ISBN: 0131011715, 9780131011717.

Kumar, A., Kaur, M. 2012. An improved algorithm for solving fuzzy maximal flow problems. International Journal of Applied Science and Engineering, 10(1), 19–27.

Laghari, A.A., Sun, Y., Alhussein, M., Zurangzed, K., Anwar, M.S., Rashid, M. 2023. Deep residual-dense network based on bidirectional recurrent neural network for atrial fibrillation detection. Scientific Reports, 13(1), 15109.

Liou, T.S., Wang, M.J. 1992. Ranking fuzzy numbers with integral value. Fuzzy Sets and Systems, 50, 247–255.

Minieka, E. 1973. The lexicographic maximum flow problem. Operations Research, 21(2), 507–525.

Mishra, K.K. 2021. Maximum flow problem in fuzzy environment. Malaya Journal of Matematik, 9(1), 64–70.

Pangeni, B.P., Dhamala, T.N. 2025. Maximum flow in hybrid network with intermediate storage. OPSEARCH, Springer; Operational Research Society of India, 62(2), 833–849.

Peng, J., Zhang, B., Chen, L., Hui, L. 2024. A survey on uncertain graph and uncertain network optimization. Fuzzy Optim. Decis Making, 23, 129–153.

Pyakurel, U., Dhamala, T.N. 2016. Continuous time dynamic contraflow models and algorithms. Advances in Operations Research, 2016, 902460.

Pyakurel, U., Dempe, S. 2020. Network flow with intermediate storage: Models and algorithms. SN Operations Research Forum, 1, 37.

Pyakurel, U., Dempe, S. 2021. Universal maximum flow with intermediate storage for evacuation planning. In: Kotsireas I.S., Nagurney A., Pardalos P.M., Tsokas A. (eds) Dynamics of Disasters. Springer, Optimization and Its Applications vol. 169, Springer, Cham.

Soltani, A., Haji, R. 2007. A project scheduling method based on fuzzy theory. Journal of Industrial and Systems Engineering, 1(1), 70–80.

Yan, Q., Gao, W., Ma, R. 2023. Multi-objective random-fuzzy optimal power flow of transmission-distribution interaction considering security region constraints. Electric Power Systems Research, 224, 109715.

Zadeh, L.A. 1965. Fuzzy sets. Information and Control, 8(3), 338–353.

Zhang, J., Lim, J.H., Kim, J.S. 2002. Fuzzy linear programming approaches for network flow problems. Journal of Optimization Theory and Applications, 112(3), 467–482.

Zhao, L., Yu, J., Laghari, A.A., Fang, K., Gadekallu, T.R. 2025. Feature jointly-based knowledge enhancement model for multimodal sentiment analysis. IEEE Transactions on Computational Social Systems, 11, 404

Zimmermann, H.J. 1991. "Fuzzy Set Theory and Its Applications". Kluwer Academic Publishers.