Information Sciences Letters
An International Journal

http://dx.doi.org/10.18576/isl/120830

Finding the Optimal Shortest Path in Stochastic Networks Using the Markov Decision Process (MDP)

D. Al-Tayar and Z. Alisa*

Electrical Department, College of Engineer, University of Baghdad, Baghdad, Iraq

Received: 29 Jun. 2023, Revised: 12 Jul. 2023, Accepted: 25 Jul. 2023
Published online: 1 Aug. 2023

Abstract: This work introduces new routing algorithms for stochastic networks. The problem addressed here considers multiple stochastic, time-dependent disruption levels on different links. This type of network makes routing decisions a challenging problem due to its stochastic nature. A novel Markov Decision Process (MDP) has been proposed and developed to handle the issue of multiple disruption levels (four levels were tested in this article). In addition, the developed approach has a hydride policy for the optimal path between two nodes depending on online and offline data and when to switch between them. A modelling framework that implements available online and offline network data and a novel cost structure that estimates the probable change in travel time resulting from expected disruption levels is delivered to get an optimal, reliable path. The proposed offline and online algorithms based on the suggested approaches are proven to efficiently handle the problem of stochastic network routing with multiple disruption levels. Results showed the findings demonstrated the efficacy of the suggested approach, contingent upon the incorporation of the *Expected transition Cost* (*ETC*) function into the primary computational equations. These equations were evolved from the calculation methodology employed for the protocols.

Keywords: Expected transition cost (*ETC*), Markov Decision Process (MDP), multiple disruption levels, Stochastic time-dependent networks.

1 Introduction

In stochastic networks, such as communication and transportation: finding the optimal routing path is a critical problem that seeks to minimize a given objective function, such as the shortest path, the lowest cost, or the maximum flow. The problem of finding optimal routing in stochastic networks has significant practical applications in various fields, including transportation, communication, logistics, and supply chain management. For example, finding the optimal routing path in transportation networks can help minimize travel time, reduce fuel consumption, and improve the network's overall efficiency. Finding the optimal routing path in communication networks can help maximize network performance, reduce congestion, and ensure reliable data transmission [1,2,3,4]. There has been significant research interest in developing efficient and effective algorithms for finding optimal routing of stochastic networks. Finding the path with the lowest cost is possible in deterministic networks using classic methods like Dijkstra's algorithm combined with the Bellman-Ford algorithm, but these techniques may not be well-suited for stochastic networks due to their inherent uncertainty. Advanced algorithms, such as reinforcement learning, genetic algorithms, ant colony optimization, and machine learning-based methods, have been proposed for finding optimal routing in stochastic networks. Despite the significant progress in the field of finding optimal routing, there are still many challenges to be addressed. The dynamic nature of stochastic networks makes it difficult to predict the optimal routing path, and the trade-off between network performance and computational complexity needs to be carefully considered. Additionally, the emergence of new technologies and applications, such as the Internet of Things (IoT) and 5G communication networks, presents new challenges and opportunities for finding optimal routing in stochastic networks New literary works recently published contain dynamic versions of the shortest path to obtain reliable results from disrupted networks, Within the context of the presented research, a prominent challenge encountered by scholars was effectively addressed. This involved the intricate task of quantifying the cost associated with transitioning between disruption levels, accompanied by the incorporation of a pivotal factor *(ETC)* that enhanced the precision of outcomes, particularly for online and hybrid algorithm.

The topology of a route can be divided into offline and online varieties to handle interruptions. The generation of connections is completed before starting the offline architecture. Therefore, changes in disruption levels cannot be taken into account. Thus, either a naive routing algorithm is used where disruption levels are ignored, which means deterministic travel times are considered, or a robust routing algorithm is employed. Disruptions are assumed to occur inevitably, so

*Corresponding author e-mail: *d.al-tayar1702d@coeng.uobaghdad.edu.iq*

the worst disruption level for travel time is implemented. For instance, the disadvantage of a negative routing policy is that, in the event of a disruption, the cost of traveling rises, while the drawback of a robust routing policy is that it relies on worst-case congestion, which takes place with a small probability and leads to a higher cost than a real one. The offline policies cannot cope with the real-time dynamics and cannot trace disruption changes by their nature. Online topology, on the other hand, allows policies to update the routes continuously according to the online realization of disruption levels and brings costs closer to reality. The study focuses on identifying efficient routing strategies in densely populated roadway networks, where trip duration variability adds uncertainty. It integrates discrete random variable analysis and a linear program approach to tackle the complex problem. Empirical data from Manhattan's street network is used to ground the study. A novel hierarchical control system is introduced, using a perimeter controller at the higher echelon and a reinforcement learning algorithm dynamically adapting to evolving traffic patterns. This framework harmonizes control operations across different levels, augmenting urban road network management through advanced computational techniques. A comprehensive methodology is proposed, using empirical traffic data to construct statistical models and fashion a stochastic, time-evolving road network model called STV. The formulation of the least expected time (LET) routing predicament is presented, leading to the generation of time-varying K-fastest pathways for assessing potential trajectories. Empirical experiments validate the prowess of this approach, particularly during peak hours. The study also shifts its focus to stochastic complex networks with mixed delays, focusing on a distinctive pinning impulsive controller for finite-time synchronization (FTS). This controller optimizes control resource allocation and ensures FTS by identifying nodes with substantial error norms. The study broadens the applicability of the proposed framework by expanding the encompassing conditions to encompass both general and mixed delays. These interconnected insights contribute to a deeper understanding of network optimization, traffic management, and synchronization dynamics [5,6,7,8]. All along the way, the shortest paths are generated and updated when the online data is obtained from the available communication systems. In this case, optimization cycles are continuously activated after retrieval of the online information at reached nodes of the network. This research suggests a new approach for finding the optimal path concerning dynamic conditions, and it handles all issues online with multiple stochastic disruption levels.

2 Related works

New techniques for finding the most reliable paths in stochastic and time-varying networks are the focus of [9]. The study evaluates dependability based on the ratio of the chance of reaching the destination on time to a specified arrival time threshold. The authors propose two different algorithms that can determine the optimal strategy and route on the network. The first algorithm utilizes a decreasing time order to identify the optimal strategy to achieve the best route from any node and time combination. The second algorithm uses network pruning and label correction methods to identify the optimal route between the source and sink points for a given departure time. The effectiveness of both algorithms is established through proof of correctness and the computation of complex expressions. The study provides useful insights into applying stochastic and time-dependent networks for reliability-based modelling and analysis, which can benefit large-scale transportation networks. Markov To discover the best course of action in a stochastic network, among other things, decision processes have been widely introduced in the literature. [10]. In [11], The authors made it apparent that models are required for operations research to effectively explain difficulties and facilitate the construction and presentation of solution techniques. Rigorous methodologies have lagged behind rigorous models in dynamic routing, making it challenging for researchers to undertake rigorous science due to the complexity of capturing the combined development of sequential routing decisions and stochastic information. The authors provide a modelling framework that uses the vast literature on route-based planning and optimization and does a good job of bridging the gap between application and technique. The State, action, and reward structures of standard Markov decision processes (MDPs) are expanded in route-based Markov decision processes (MDPs). Because of this, route-based MDPs simplify the conceptual link between dynamic routing problems and the route-based tactics commonly used to handle them, such as generating and revising routes in response to newly discovered information. The authors believe route-based MDPs can help dynamic routing studies be more rigorously scientific, provide academics with a common modelling language, enable better research, and enhance the categorization and description of solution approaches. Many other papers illustrated the concept of route plans and sequential decision models [12], [13], and [14]. The most related work is [15], where different online and offline algorithms are applied using MDP and a derived transition matrix connected to the Probability of disruption levels and its steady State. Finding the best route in congested street networks where travel times are typically uncertain can be a challenging problem with significant practical implications. As most existing methods focus on minimizing the expected travel time as their sole objective, such solutions may not be appropriate when the travel time variance is high. This study presents a new approach that addresses this issue by formulating the problem as finding a routing policy that minimizes the expected travel time while retaining a specified probability of on-time arrival. The study uses a discrete random variable model to represent the stochastic travel time on each segment of the road network, which translates into a Markov decision process model. This approach enables the problem to be viewed as a linear program, making it easier to solve. Additionally, the study includes a case study focused on the streets of Manhattan, New York. Real-world data was used to develop the model of travel times, and the proposed method was used to generate optimal routing policies [16]. This research aims to develop a model that introduces a new representation of the MDP model for offline and online routing policies to handle many disruption-level transitions and get the best results by investing the acquired information about routes, if available elsewhere, depending on offline data, which is the case. The proposed model has novelty in transferring the disruption levels transition to an additional expected transition cost (ETC function, which will be explained in detail in subsection 4.5. The hybrid strategy considers both the time-varying and constant probabilities of network connection transit times and current data. The suggested hybrid routing policy handles the situation where some parts of the network cannot provide online data to clarify their congestion status. It can be used to compare results between all cases. Following the collected disruption level of the link, the updated data will be represented in the modification relating to the length of the link's trip cost, which will be fair enough to change the state value function of MDP and lead to better decisions for constructing a consequent route. The disruption levels and their probabilities are assumed to be pre-collected statistically about links and available to be implemented in the algorithm in a suitable form, whether online or offline. Briefly, some points that distinguish the developed approach are the hydride policy, the many levels of distributed levels for each node, and the transition between offline and online policies [17.18,19,20].

3 Problem Statement

The network is represented as a graph with intersections as nodes and routes as edges, as common in the literature (Zheng, Thangeda, & Savas, 2021), (Thomas & White, 2007). The network is modelled using a direct graph $\mathcal{G}(\mathcal{N},\mathcal{E})$, where $\mathcal{N}=\{1,2,\dots,n_g\}$, are graph nodes and $\mathcal{E}=\{1,2,\dots,n_e\}$ are the graph edges. Each intersection in the network is represented by a node $n\in\mathcal{N}$, and each route between any two intersections is represented by a direct edge $e\epsilon\mathcal{E}$ where $e \stackrel{\text{def}}{=} (n_s,n_d)|n_s,n_d\in\mathcal{N},n_s\neq n_d$. E_v is used to refer to vulnerable links, each vulnerable link can accept any number of pre-modelled disruption levels, and travel time is discrete. Suppose that $\mathcal{V}=\|E_v\|$, using a standard vector that represents the degree of disruption at each potentially weak node $D=\ (D^1,D^2,\dots,D^{\mathcal{V}})$, the element D^i: $i=\{1,2,\dots,\mathcal{V}\}$ and this vector right here symbolizes the i^{th}vulnerable link and it's linked to various degrees of instability $L^e\in\{L^1,L^2,\dots,L^{M_e}\}$ where M_e indicates the link's degree of breakdown, writers in (Gatie, Yew, Teasu, & Puay, 2016) For potentially disruptive edges, think about using two or three levels. $L^e\in\{0,1,2\}$ (0: uncongested, 1: crazily clogged, 2: densely crowded), the advantage of the developed model in this paper is that it accepts as many levels as are required to characterize the disruption status, resulting in a more precise estimation of travel time between two nodes. S: source and D: destination.

The model of the network should provide a probability matrix that models the Probability of moving from one disruption state to another, and every level of disruption is defined with consequent travel time. At specific epochs k, a random variable $\widehat{D}=(\widehat{D}^1,\widehat{D}^2,\dots,\widehat{D}^{\mathcal{V}})$ is used to represent the realization of the current disruption state of each link depending on its congestion state, the online routing status of links is updated continuously, and the absence of online data for some vulnerable links will be handled through this work.

An extended discrete-time finite of the Markov decision process MDP is proposed, where directed and undirected edges can be handled. Discrete random variables represent the stochastic travel times on edges, where each travel time is associated with a disruption level with transition probability. The model below considers time-invariant edge travel time distributions and can be generalized to time-varying distributions without restrictions.

A path Θ is defined as a chain of nodes $\Theta=\{n_1,n_2,\dots,n_k\}\ where\ k\le n_g$. On the graph $\mathcal{G}$. Let π define a policy. This Policy works as a function that determines the next node to head on based on time and currently active nodes. Suppose Θ is the set of all possible paths and Π is the set of valid policies. For any $\theta\epsilon\Theta$, the total travel time along a path is T_θ. The value T_θ Is the sum of the path's edges' trip times, expressed as random variables, i.e., $T_\theta=T_{n_s,n_2}+T_{n_2,n_3}+\cdots+T_{n_{k-1},n_d}$.

Given a source node $n_s\epsilon\mathcal{N}$ and a destination one $n_d\epsilon\mathcal{N}\backslash\{n_s\}$, having T_{sd} as the random variable represents the travel time from n_s to n_d. The goal is to find a policy. $\pi^*\epsilon\ \Pi$ show that in Equation (1).

$$\pi^*=\arg\min\mathbb{E}[T_{sd}] \tag{1}$$

The main example for our approach is a flight system between cities; suppose the cities are the nodes, and edges are the available paths between them. It is important to know that airplanes must follow a specific path. H levels can represent the disruption levels here according to many cases:

- The path is crowded (the number of airplanes that are in the path now (or at the time of flying)
- The path is allowed not because of the relations between countries.
- The climate conditions:
    - Wind.

  - Clarity.
  - Barometric Pressure.
  - Temperature.
- Other unexpected conditions

H disruption levels might represent these conditions. These six levels of disturbance might look like this if the disruption scale were [0-1]:

- Free [0-0.16]
- Almost Free [0.17-0.33]
- Normally Free [0.34-0.50]
- Normally Crowded [0.51-0.67]
- Almost Crowded [0.68-0.84]
- Crowded [0.84-0.1]

And according to these, the optimal path should reconsider all conditions in this stochastic network.

4 Markov Decision Process (MDP)

4.1 Markov Decision Process Definitions

Markov decision processes have been widely used in literature. In work (White, 1993), D.J. White introduces a survey of real applications that uses MDP where it is implemented and affects decisions taken. There were eighteen different fields in the survey, which emphasizes the importance and efficiency of this structure in various applications in the real world, especially where stochasticity is involved.

A Markov decision process (MDP) is a reward or penalty process with decisions. It forms an environment where all states are Markov, and it's represented as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ (Silver, 2015)

$\mathcal{S}$ is a finite set of states.

$\mathcal{A}$ is a finite set of actions

$\mathcal{P}$ is a state transition probability matrix

$\mathcal{P}^{a}_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$

$\mathcal{R}$ is a reward (cost) function, $\mathcal{R}^{q}_{s} = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$

γ is a discount factor $\gamma \epsilon [0,1]$.

4.2 State

The current node $cn_k \epsilon \mathcal{N}$ and the disruption realization vector $\widehat{D}_k$ Display the current system state. k, $s_k \epsilon \mathbb{S}$. $\widehat{D}_k$ Defines the disruption level of disrupted links at stage k. Thus, the possible disruption level change of the disrupted link to adjacent levels can be inferred.

A revised version of the disruption state vector is created at each step. $\widehat{D}_k$ Is implemented. That is an observation result of the realization of the disruptions in the network. $(\widehat{D}^{v}_{k} \epsilon \mathrm{L}^{e})$ at each stage. There is a one-to-one correspondence between the status of the network and the disruption state vector, reflecting congestions resulting from different environment states (mainly accidents, bad weather cases in road networks, and excessive load of link capacity).

Therefore, the System's Condition at Each Step k is: $S_k = (n_k, \widehat{D}_k)$. The process of termination occurs when the destination is reached. The possible goal states are the pairs of the destination and the possible disruption realizations. These goal states form a set $\mathbb{G} \epsilon \mathbb{S}$, where it is expected that all desired outcomes are expensive and time-consuming-free.

4.3 Action

The stage action k, x_k, is to evaluate in the next node to head on under a policy. Given the current State, this protocol seeks to reduce the estimated trip time to the target node, n d. The action set only includes neighbours of the current node, May k, with the same transition probability.

$X_k^{\pi}(S_k)$: Decision function determines, at step k under a Policy π given State S_k , the node at stage $k + 1, (n_{k+1})$.

Π: Set of possible policies. Each $\pi \epsilon \Pi$ represents a different policy where $\{X_k^{\Pi}\ (S_k)\ \}_{\pi\epsilon\Pi}$ is the set of decision functions.

4.4 Cost function

Moving of node $n_k\ to\ x_k = n_{k+1}$ under-realized disruption state vector $\widehat{D}_t$ Outcomes in a nonnegative real cost that, given the present disruption state, equal the deterministic transit time between the two nodes. Let's refer to this known and discrete travel time as $t_{n_k,x_k}(\widehat{D}_k)$. At every node n_k, an instant expense incurred as a result of the action x_k, this cost is $c(S_k, x_k)$ since show that in Equation (2):

$$c(S_k, x_k) = t_{n_k,x_k}(\widehat{D}_k) \tag{2}$$

It is presumable that the trip of the links directly relates to the network's State of interruption at the time the link is entered. The online retrieved data determines the trip and immediate cost incurred due to the action. cn_k if possible, to all surrounding nodes based on the actual disruption state of links cn_kIs reached.

4.5 The state Transition function

In this kind of network, two transitions are considered. The first one concerns nodes, the other one is related to disruption states. take a State S_k as well as a chosen action x_k , Under a given disruption status, a transition is made to the next node, $S_{k+1} = (n_{k+1} = x_k, \widehat{D}_k)$. The state transition results show in Equally (3) from the activity and includes

$$n_{k+1} = x_k \tag{3}$$

The other transition is connected to the disruption status vector transition from $\widehat{D}_k$ to $\widehat{D}_{k+1}$ based on a matrix of Markovian transitions. The arrow D_{k+1} comprises random variables indicating each weak link's network disruption level for the following stage.

Note 1: Something interesting about such systems is that transition to a lower disruption state must minimize the expected cost of traveling across the link. In comparison, transitioning to a higher state must increase the expected cost of traveling while maintaining the same disruption state must not add cost to the expectation of traveling time.

The transition probability matrix described by Equation (4) is supposed to represent a model input. Having $P_{l,l'}^{v}$ the Probability of a unit-time transition between any two weak link disruption levels $P_{l,l'}^{v} = P\{\widehat{D}_{k+1}^{v} = l' \big| \widehat{D}_k^{v} = l\}$, a susceptible link's unit-time transition matrix $v, v \epsilon E_v$, with L^e The following are examples of hypothetical interruption scenarios:

$$\mathrm{P}_{t,t+1}^{v} = \begin{bmatrix} P_{l^1,l^1}^{v} & P_{l^1,l^2}^{v} & \cdots & P_{l^1,l^{Me}}^{v} \\ P_{l^2,l^1}^{v} & P_{l^1,l^2}^{v} & \cdots & P_{l^2,l^{Me}}^{v} \\ \vdots & \vdots & \vdots & \vdots \\ P_{l^{Me},l^1}^{v} & P_{l^{Me},l^2}^{v} & \cdots & P_{l^{Me},l^{Me}}^{v} \end{bmatrix} \tag{4}$$

$\forall t = 0,1,2, \dots \forall\ v\epsilon E_v$

$$\mathcal{T}^{v} = \begin{bmatrix} t_{l^1}^{v} \\ t_{l^2}^{v} \\ \vdots \\ t_{l^{Me}}^{v} \end{bmatrix} \tag{5}$$

$\mathcal{T}^{v}$: Defines the time related to disruption levels of disrupted link v shown in equation (5).

M_e: Maximum disruption level.

The transition involves a specific row depending on the realized disruption level of the link and consequently will add cost to the objective function.

The proposed transition method involves neighbour disruption states depending on the current realized disruption level of the link. In other words, to implement l^{+1}, l, l^{-1} states only. That is to stay at the same disruption level and transit to the previous or next disruption level, as these states are more probable to happen during the same stage. Consequently, this will add cost to the objective function at the same stage as the link (n_i, n_{i+1}) under concerns.

In the realization of this notion considering.

Note 1, this should minimize or maximize the travel time of the link under concerns Equation (7), where $ETC(\widehat{D}_k^{v})$ is the function to calculate the change in travel time.

Note 2: To maintain probability distribution for this State, Probability can be normalized by using factor $\alpha = \frac{1}{p_{l,l^{-1}}+p_{l,l}+p_{l,l^{+1}}}$, if the disruption state is revealed to be in the no-disruption or maximum disruption levels, the related coefficient of the previous and next disruption state is omitted, respectively.

A discount factor $\gamma \in [0,1]$ is used to minimize the effect of transition; its value depends on how much the updated information about links is trusted, in addition to the nature of the network. The ultimate transition cost is the summation of Equation (7) Which contains two parts: Equation (7a), which represents the transition to a higher disruption level, and Equation (7b), which represents the transition to a lower disruption level, which is elucidated by the details provided in Equation (6).

$$ETC\left(\widehat{D}_k^v\right) = \sum_{S_D'} P(D_{k+1}|D_k)\left(tr(D_k)\right) \quad (6)$$

$$ETC\left(\widehat{D}_k^v\right) = \mathrm{a}(\mathrm{A}+\mathrm{B}) \quad (7)$$

$$\mathrm{A} = p_{l,l^{-1}}^v\left(\mathcal{T}^v(l^{-1}) - \mathcal{T}^v(l)\right) \quad (7\text{ a})$$

$$B = p_{l,l^{+1}}^v\left(\mathcal{T}^v(l^{+1}) - \mathcal{T}^v(l)\right) \quad (7\text{ b})$$

$\alpha p_{l,l^{-1}}^v$: is the scaled Probability to transit the previous disruption state of v link.

$\alpha p_{l,l^{+1}}^v$: is the scaled Probability to transit the next disruption state of v link.

4.6 The Optimal Policy and travel cost

The purpose of optimum routing is to reduce the total anticipated travel time in a discrete random horizon **K** from the source node to the destination state. This objective was ascertained by minimizing the value of equation (9).

$$\min_{\pi\epsilon\Pi} E \ \sum_{k=1}^{n} c\left(S_k, X_k^\pi(S_k)\right) \quad (8)$$

where $x_k = X_k^\pi(S_k)$ is the decision based on the decision function $X_k^\pi(S_k)$ under Policy π, given the current State S_k.The computation of the state value function employs a retrospective recursion method through the utilization of the Bellman optimality equation (9a) and (9b).

$$V_k^*(S_k) = \min_{x_k \epsilon \mathcal{X}_k} c(S_k, x_k) + V_{k+1}^*(S_{k+1}) + \gamma \times ETC(\widehat{D}_k^v) \quad (9a)$$

$$V_n(S_n) = 0 \quad (9b)$$

where S_n is the goal state, $S_n \epsilon\ \mathbb{G}$. the relationship is given by Equation (10):

$$x_k^* = arg \min_{x_k \epsilon \mathcal{X}_k} c(S_k, x_k) + \mathbb{E}\left(V_{k+1}(S_{k+1})\right) \quad (10)$$

The anticipated aggregate cost of the optimal trajectory will be presented in Equation (11).

$$T_\theta^* = \sum_{e\epsilon x_k^*} c(e) + \gamma \times ETC(\widehat{D}_k^v, e). \quad (11)$$

e: is a link in the optimal path.

As there is an absorbing cost-free goal state, information is propagated backward using a value iteration algorithm. The condition of termination in this algorithm terminates when all states, $S\ \epsilon \mathbb{S}$ have a fixed cost until the goal state destination is reached. To do this, it is necessary to define a value function. $V_t(S)$ that will retain the anticipated total cost beginning in a Steady State for a range of ***t* steps** till the destination. The algorithm searches for a stable optimal approach that shortens the time required to reach the goal state the following conditions are met, the value iteration is shown to converge to an optimal stationary value function for the stochastic shortest path problem (Prakash, 2020), (Zheng, Thangeda, & Savas, 2021):

1.The state set includes a termination state or a goal ($\mathbb{G}\ \epsilon\ \mathbb{S}$) which is free and absorbent: $P\left(d, \widehat{D}_{k+1}\middle|d, \widehat{D}_k\right) = 1$.

2.There are no absorbing cycles (self-loops) in any non-goal states.

$P\left(n_k = n_{k+1}, \widehat{D}_{k+1}\middle|n_k, \widehat{D}_k\right) = 0, \forall\ S\epsilon\ \mathbb{S}\backslash\mathbb{G}$.

3.From each state, at least one appropriate strategy that achieves the desired State with a limited cost and a probability of 1 ($V_t(S) < \infty, \forall\ S\ \epsilon\ \mathbb{S}$. This is also known as a connectivity assumption, which ensures that every State has at least one path leading to the desired State.

4. Every bad Policy results in an infinite cost from every stage that doesn't reach the desired State.

5. By satisfying these five conditions, the value iteration is guaranteed to converge in a finite number of stages where the graph cannot contain absorption cycles for the non-goal states.

Numerically, we consider ϵ consistency rule for the termination condition where $\epsilon > 0$ and modest enough to purchase a coverage that is certain to expire within a year ϵ of optimum $\epsilon = 0.1$.

5 Suggested algorithms

The suggested algorithm can be shown in the next flowchart:

Initialization

$\forall S \epsilon \mathbb{S}$, define

$$V_0(S) = \begin{cases} 0 & if\ node\ v\ is\ destination, S\epsilon\mathbb{G} \\ \infty & otherwise \end{cases}$$

While loop

Determine link status (disrupted or not)

Get transition probabilities if they existed

Calculate state values backward

$\forall S(n, \widehat{D})\epsilon\mathbb{S} \setminus \mathbb{G}$:

$$V_t^*(S) = \min_{x\epsilon Ne(n),} c(n, \widehat{D}, x) + V_{t-1}^*(S') + \gamma \times ETC(n, \widehat{D})$$

$$V(s) = \min_t V_t(S)$$

Where $(\boldsymbol{Ne(n)} = \boldsymbol{Neigbours(n)})$

Compute stabilization at iteration $\boldsymbol{t}$

$$\Delta V_t(S) = |V_t(S) - V(S)|$$

Test stabilization condition

$$\max \Delta V_t(S) < \epsilon\ and\ V_t(S) < \infty\ \forall\ S\epsilon\mathbb{S}$$

End Loop

Choose the optimal routing policy.

$$\pi^{V_t}(S) = \operatorname*{argmin}_{x\epsilon Ne(n)} c(n, \widehat{D}, x) + V_{t-1}^*(S') + \gamma \times ETC(n, \widehat{D})$$

End of the stage k

After applying this algorithm, the next minimal cost node to visit is picked, and the decision to move to it is taken. Upon reaching the specified node, un-updated information is retrieved about the network. Thus, the algorithm is repeated to update the optimal path until the destination node is reached.

The input to the optimal routing policy is the complete network disruption state information which contains the travel-time-dependent transition probabilities.

The State is represented by using the Nave routing policy. $S_k(n_k, \widehat{D}(k) = (0,0, \dots, 0))$, In addition, the Bellman optimality equation will be transformed as shown in Equation (12).

$$V_k(n_k, \widehat{D}_k) = \min_{x_k \epsilon X_k} c(S_k, x_k) + V_{k+1}(S_{k+1}) \quad (12)$$

Along with Eq.(9b). The route is found as shown in Equation (13):

$$V_t(S) = \min_{x_k \epsilon X_k} c(S, x_k) + V_{t-1}(S) \quad (13)$$

A Robust routing policy represents the State $S_k(n_k, \widehat{D}(k) = (1,1, \dots, 1))$. It assumes that all disruption will certainly occur, and the Bellman optimality equation will be the same as in Equation (12) and Equation (13), considering the change in the direct cost.

The quantity of real-time information that is accessible determines the online routing topology. Indeed, each node has access to the most recent online information regarding the disruption implementations of the susceptible links. $m \leq \mathcal{V}$. Certain connections cannot be found online for whatever reason; hence the estimated values should depend on steady-state Probability. When m same $\mathcal{V}$. Online routing strategies are the greatest choice for considering disruption change on weak links.

The steady-state Probability is used for any vulnerable link where its relevant disruption status is not available.

The chance of being in a state at Steady State $\widehat{D}^{v}$ For any link, v the probability transition matrix's related row serves as the basis for the definition. $P(\widehat{D}^{v}), \widehat{D}^{v} \epsilon L^{e}$. The predicted value of the link v for traveling from node to node n_k to node x_k is denoted by equation (14).

$$\bar{t}_{n_k,x_k} = \sum_{\widehat{D}^{v}} P(\widehat{D}^{v}) t_{n_k,x_k}(\widehat{D}^{v}) \tag{14}$$

At every node n_k, Because online information is unavailable for all disrupted links, the state value function can be determined by employing Algorithm 1 and changing the Bellman optimality Equations (15).

$$V_k^*(n_k, \widehat{D}_k^{n_k}) = \begin{cases} \min_{x_k \epsilon \mathcal{X}} \begin{matrix} c(n_k, \widehat{D}_k^{n_k}, x_k)) \\ +V_{k+1}^*(x_k, \widehat{D}_k^{n_k}) \\ +\gamma \times ETC(\widehat{D}) \end{matrix} & \begin{matrix} if\ S_k \epsilon \mathbb{S}\mathbb{G}\ and \\ information\ updated \end{matrix} \\ \min_{x_k \epsilon \mathcal{X}} \bar{t}_{n_k,x_k} + V_{k+1}^*(x_k, \widehat{D}_k^{n_k}) & \begin{matrix} if\ S_k \epsilon \mathbb{S}\mathbb{G}\ and\ no \\ updated\ information \end{matrix} \\ 0 & if\ S_k \epsilon \mathbb{G} \end{cases} \tag{15}$$

The trajectory is established via Equation (16).

$$V_t(n_k, D^{n_k}) = \begin{cases} \min_{x_k \epsilon \mathcal{X}} \begin{matrix} c(n_k, \widehat{D}_k^{n_k}, x_k)) \\ +V_{t-1}(x_k, \widehat{D}_k^{n_k}) \\ +\gamma \times ETC(\widehat{D}) \end{matrix} & \begin{matrix} if\ S_k \epsilon \mathbb{S}\mathbb{G}\ and \\ information\ updated \end{matrix} \\ \min_{x_k \epsilon \mathcal{X}} \bar{t}_{n_k,x_k} + V_{t-1}(x_k, \widehat{D}_k^{n_k}) & \begin{matrix} if\ S_k \epsilon \mathbb{S}\mathbb{G}\ and\ no \\ updated\ information \end{matrix} \\ 0 & if\ S_k \epsilon \mathbb{G} \end{cases} \tag{16}$$

Hybrid routing policy uses online information if available otherwise, it uses time-independent distribution probability to infer the shortest path of stochastic networks. Our main objective is to design.

6 Result and analysis

First, an illustrative example of proposed routing policies is demonstrated to explain the applicability of the work and its effectiveness. Then, a statistical performance experiment is conveyed, and the consequent Wilcoxon rank test is also delivered for this experiment.

6.1 illustrative example

Considering the network shown in Fig. **1** with 15 nodes and travel times depending on disrupted levels on each link. Travel time and travel probabilities of links are generated randomly.

The network has six disrupted links; 3 of them with two disruption levels,1 with three disruption levels, and 2 with 4. The disruption levels probability matrix for links generated randomly (In a real test, this should be pre-defined) is shown in Table 1.

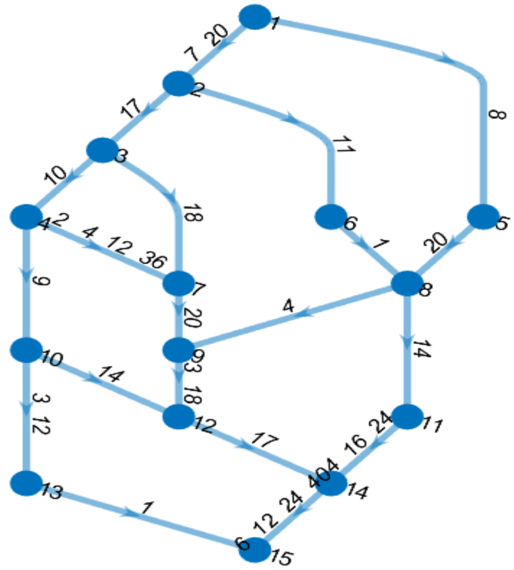

Fig. 1: Graph representation of stochastic networks.

Table 1: Probability factors for each disrubtion levels.

Disrupted link	No disruption Levels	Probability matrix
$e_{1,2}$	2	0.5141,0.4859 0.5679,0.4321
$e_{4,7}$	4	0.220,0.4900,0.1396,0.1497 0.0372,0.7988,0.1126,0.0514 0.2249,0.4832,0.2206,0.0712 0.1683,0.1661,0.5638,0.1018
$e_{9,12}$	2	0.5519,0.4481 0.8050,0.1950
$e_{10,13}$	2	0.6952,0.3048 0.8370,0.1630
$e_{11,14}$	3	0.2403,0.5348, 0.2249 0.1462,0.8148 ,0.0390 0.2716,0.6464,0.0821
$e_{14,15}$	4	0.0990,0.6078,0.1288,0.1644 0.2377,0.4543,0.1000,0.2080 0.0336,0.0151,0.9103,0.0410 0.0811,0.0754,0.7085,0.1350

The travel time for each disruption level is illustrated on links in Fig. **1** ascending as it has assumed the travel time increases with the disruption level. It is also assumed that the travel time for links in naïve and robust construction corresponds to the first and last disruption levels. The source node is Node 1, and the target node is Node 15.

The shortest path resulted from the naïve routing policy evaluated by Equation (13) and Equation (9b) as:

1 2 6 8 11 14 15

The total cost of this path is **43**, where the minimum link cost is considered in Equation (12) for each link. As shown, no disruption levels transition is applied.

The shortest path is evaluated in Equation (13) and Equation (9b) for robust routing policy is:

1 2 3 4 10 13 15

The total cost of the robust path is **69**, where the maximum link cost is considered in Equation (12) for each link. As shown, no disruption levels transition is applied.

Using Online and hybrid algorithms, the route should be updated after each stage from the current node at this stage to the target node.

The optimal shortest path for the online algorithm is obtained using Equation (9a) and Equation (9b).

In the first stage, using the Online algorithm at the first stage yields the optimal route as 1 2 6 8 11 14 15 considering the disruption levels are selected randomly and discount factor $\gamma = 0.7$. While for the next stage, starting from node two after the network has been updated, the optimal shortest path yielded from the online algorithm is 2 3 4 10 13 15.

Table 2: final cost of Online algorithm and transtion betwen the disruption levels.

Stage No	1	2	3	4	5
Start node	1	2	3	4	10
Updated information about disruptions	$\widehat{D}_{e_{1,2}} = 2$ $\widehat{D}_{e_{4,7}} = 3$ $\widehat{D}_{e_{9,12}} = 1$ $\widehat{D}_{e_{10,13}} = 2$ $\widehat{D}_{e_{11,14}} = 1$ $\widehat{D}_{e_{14,15}} = 1$	$\widehat{D}_{e_{1,2}} = 1$ $\widehat{D}_{e_{4,7}} = 4$ $\widehat{D}_{e_{9,12}} = 2$ $\widehat{D}_{e_{10,13}} = 1$ $\widehat{D}_{e_{11,14}} = 2$ $\widehat{D}_{e_{14,15}} = 3$	$\widehat{D}_{e_{1,2}} = 1$ $\widehat{D}_{e_{4,7}} = 4$ $\widehat{D}_{e_{9,12}} = 2$ $\widehat{D}_{e_{10,13}} = 1$ $\widehat{D}_{e_{11,14}} = 1$ $\widehat{D}_{e_{14,15}} = 4$	$\widehat{D}_{e_{1,2}} = 2$ $\widehat{D}_{e_{4,7}} = 3$ $\widehat{D}_{e_{9,12}} = 1$ $\widehat{D}_{e_{10,13}} = 2$ $\widehat{D}_{e_{11,14}} = 1$ $\widehat{D}_{e_{14,15}} = 3$	$\widehat{D}_{e_{1,2}} = 2$ $\widehat{D}_{e_{4,7}} = 2$ $\widehat{D}_{e_{9,12}} = 1$ $\widehat{D}_{e_{10,13}} = 2$ $\widehat{D}_{e_{11,14}} = 1$ $\widehat{D}_{e_{14,15}} = 2$
Optimal Path	1,2, 6,8,11,14,15	2,3,4,10,13 15	3,4,10,13,15	4,10, 13,15	10,13,15
Cost	50.83	40	23	16.72	7.72

It's clear from Table 2 that the path to the destination resulting at stage 1 is different from the one at stage 2 for this example as a result of different disruption level realization at each stage, while the path did not change after stage 2 until the destination was reached.

The application of a hybrid algorithm is similar to the online one, but it can deal with absent information about some or even all disrupted links. It's assumed that half of a disrupted link's updated information has not been obtained. Table 3 introduces the results of a hybrid algorithm besides updated and missed information. Letter (**a**) refers to an absence of information about the correspondent link. The shortest path resulting from the Hybrid policy is

1 2 3 4 10 13 15

It has no changes through stages for the given network under given conditions.

The path from source to destination using a hybrid algorithm is performed using Equation (16), and the total cost of the path using Equation (15) is 62.04 for the first stage. As seen from Table 3, no change to the path occurred during stages for this example.

Table 3: final cost of hybrid algorithm and transtion betwen the disruption levels .

Stage No	1	2	3	4	5
Start node	1	2	3	4	10
Updated information about disruptions	$\widehat{D}_{e_{1,2}} = 2$ $\widehat{D}_{e_{4,7}} = a$ $\widehat{D}_{e_{9,12}} = 1$ $\widehat{D}_{e_{10,13}} = a$ $\widehat{D}_{e_{11,14}} = 3$ $\widehat{D}_{e_{14,15}} = a$	$\widehat{D}_{e_{1,2}} = 1$ $\widehat{D}_{e_{4,7}} = a$ $\widehat{D}_{e_{9,12}} = 2$ $\widehat{D}_{e_{10,13}} = a$ $\widehat{D}_{e_{11,14}} = 2$ $\widehat{D}_{e_{14,15}} = a$	$\widehat{D}_{e_{1,2}} = 1$ $\widehat{D}_{e_{4,7}} = a$ $\widehat{D}_{e_{9,12}} = 2$ $\widehat{D}_{e_{10,13}} = a$ $\widehat{D}_{e_{11,14}} = 1$ $\widehat{D}_{e_{14,15}} = a$	$\widehat{D}_{e_{1,2}} = 1$ $\widehat{D}_{e_{4,7}} = a$ $\widehat{D}_{e_{9,12}} = 2$ $\widehat{D}_{e_{10,13}} = a$ $\widehat{D}_{e_{11,14}} = 1$ $\widehat{D}_{e_{14,15}} = a$	$\widehat{D}_{e_{1,2}} = 2$ $\widehat{D}_{e_{4,7}} = a$ $\widehat{D}_{e_{9,12}} = 1$ $\widehat{D}_{e_{10,13}} = a$ $\widehat{D}_{e_{11,14}} = 3$ $\widehat{D}_{e_{14,15}} = a$
Optimal Path	1,2,3,4,10,13,15	2,3,4, 10,13,15	3,4,10 ,13,15	4,10, 13,15	10,13, 15
Cost	62.04	47.21	30.21	20.21	11.21

6.2 Statistical experiment

This experiment aims to show the performance of the different proposed policies on different network sizes with various disruption rates and vulnerabilities. The conditions of the experiment are set as follows:

Network size: small network 16 nodes, large network 36 nodes.

Network vulnerability: the vulnerability of the network reflects the number of disrupted links. The low percentage is 50 %, and the high percentage is 70%, with different disruption levels.

The travel times range is kept between 1–10-unit time.

Disruption rates have a low Probability of having disruptions [0-0.3], medium [0.4-0.6], and high [0.7-1].

Suggested Network Without losing generosity, the suggested network is shown in Fig. **2**, where at least each node has two neighbor links to choose between them. Such construction prevents the evaluation of unnecessary nodes far from the shortest path. The origin node is at the top left corner, and the destination node is at the bottom left corner.

The evaluation criteria utilized in Equation 17.

$$\Delta(\%) = \frac{|E(CAP) - E(COP)|}{E(COP)} \times 100 \quad (17)$$

$Cost\ alternatiive\ policy: \boldsymbol{CAP}, Cost\ optimal\ policy: \boldsymbol{COP}$

Twelve experiments with different conditions are set in Table 4. Ten replications are generated (120) instances to extract the following results (Table 5), where the $mean\ \ cost$ represents the average cost of the routing Policy under concern. $mean\ \Delta(\%)$, max $\Delta\%$ represent the average and maximum value of the absolute difference between routing policy cost and optimal routing policy cost, respectively.

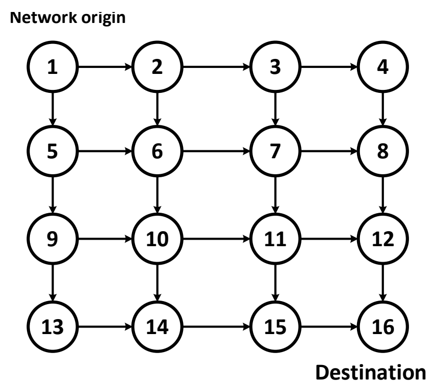

Fig. 2: suggested a network.

The experiments were carried out using just two disruption levels, and disrupted links were selected randomly, and transitions probability kept the disruption rate maintained through random generation.

$$P_{t,t+1}^{v} = \begin{bmatrix} P_{l^1,l^1}^{v} & 1 - P_{l^1,l^1}^{v} \\ 1 - P_{l^2,l^2}^{v} & P_{l^2,l^2}^{v} \end{bmatrix}, \forall t = 0,1,2,\dots \forall\ v \epsilon E_v \quad (18)$$

The number of links with unavailable online information is chosen to be 0.25 of disrupted links. In the case of small networks, this will be two links, while in the case of large networks, it will be 4 or 6 links if the low and high vulnerability condition is implemented, respectively.

Tables 5 and 6 show that the Nave routing policy yields the lowest cost, and the Robust routing policy yields the highest cost, as expected in suggested policies because each does not consider disruption changes and the naive routing algorithm, robust routing algorithm counts the minimum and maximum cost of disrupted arcs, respectively, as both acts offline.

Table 4: All cases of simulations that have been tested.

Experiment	Network size	Network Vulnerability	Disruption rate
1	Low	Low	Low
2	Low	Low	Medium
3	Low	Low	High
4	Low	High	Low
5	Low	High	Medium
6	Low	High	High
7	High	Low	Low

8	High	Low	Medium
9	High	Low	High
10	High	High	Low
11	High	High	Medium
12	High	High	High

The optimal routing algorithm measures the exact cost of disrupted links ($\gamma = 0$), which means it removes the transition cost function (ETC) from consideration. In contrast, the online routing algorithm maintains this transition probability using ($\gamma = 0.7$) to consider the disruption level change. It is noticeable that the mean cost of the online routing algorithm gets closer to the optimal one when the disruption rate increases, which reflects the explicit design goal when using γ.

It's also apparent that minimizing γ eliminates the effect of the ETC function, and the results will get closer to the optimal routing policy in any disruption rates. That can be seen from Table 6, where $\Delta(\%)$ registered the most minimum difference for the online routing policy. For the hybrid routing policy, the same γ The principle can be used. The results reveal that when the disruption rate is low, the hybrid routing algorithm is closer to the optimal result and begins to overcount when the disruption rate grows. It is pretty clear from the algorithm and noticing that it takes the cost of travel times of disruption levels, which maximizes cost when having a higher vulnerability network and a relatively high disruption rate. This higher cost at such conditions reflects the robustness of the result when some online information is missed, which is clear from Table 5.

In conclusion, γ can be considered a trust factor of acquired disruption data for online routing policy. The more trust online data, the less γ can be assigned. As well as low values of γ reflect the slow disruption level changes in the network. Finally, the hybrid routing algorithm gives the best result with a small value of γ . The robustness of results depends on the reality of the balance in the average cost.

The results of the proposed algorithms are logical. They have the property of suppleness in justifications and a low degree of complexity even with multiple disruption levels, which was not included in the results, but this can be inferred from the results.

The time consumed for our calculation has been estimated using the MATLAB environment. It's the same complexity as offline and online algorithms, proving the efficiency of using the ETC function to estimate disruption levels transition cost.

Table 5: results for All cases of simulations that have been tested.

Exp No		Naïve	Robust	Online $\gamma = 0.7$	Hybrid $\gamma = 0.7$	Optimal (online $\boldsymbol{\gamma} = \mathbf{0}$)
1	mean cost	19.11	32.10	22.66	24.40	24.48
	$mean\ \Delta\%$	20.24	34.47	6.638	9.41	0
	max Δ	58.06	126.3	34.96	44.39	0
2	mean cost	18.80	31.38	23.41	25.57	24.10
	$mean\ \Delta\%$	18.60	32.42	2.39	9.04	0
	max Δ	55.17	113.3	19.36	46.44	0
3	mean cost	19.40	31.39	24.01	26.65	24.2
	$mean\ \Delta\%$	17.64	32.28	0.69	12.77	0
	max $\Delta\%$	60.71	207.7	6.96	255.7	0
4	mean cost	19.95	36.45	24.09	26.53	26.34
	$mean\ \Delta\%$	22.48	43.03	7.5	11.36	0
	max $\Delta\%$	59.25	169.2	36.11	50.00	0
5	mean cost	18.45	35.73	24.28	27.50	25.16
	$mean\ \Delta\%$	24.74	46.93	3.14	14.00	0
	max $\Delta\%$	69.69	291.7	16.88	125.3	0
6	mean cost	18.88	35.57	24.90	28.47	25.16
	$mean\ \Delta\%$	22.94	46.37	0.59	15.29	0
	max $\Delta\%$	62.5	158.3	10.83	126.7	0
7	mean cost	30.75	51.53	36.40	38.56	38.99
	$mean\ \Delta\%$	19.9	33.75	5.94	7.35	0
	max $\Delta\%$	49.02	82.86	21.79	23.48	0
8	mean cost	31.05	52.37	38.2	41.82	39.28
	$mean\ \Delta\%$	19.65	35.99	2.49	9.13	0
	max $\Delta\%$	52.46	131.7	15.65	53.41	0

9	mean cost	30.45	50.63	37.80	41.65	38.01
	mean Δ%	18.64	35.30	0.49	11.24	0
	max Δ%	48.93	144.4	5.38	93.00	0
10	mean cost	29.31	57.31	35.67	39.45	38.31
	mean Δ%	22.21	53.03	6.31	11.05	0
	max Δ%	58.97	165.4	23.46	45.63	0
11	mean cost	29.13	57.92	37.69	42.02	39.03
	mean Δ%	24.03	51.37	3.11	12.50	0
	max Δ%	55.14	185.7	16.19	77.10	0
12	mean cost	28.81	55.38	38.62	44.87	38.95
	mean Δ%	24.98	44.32	0.83	16.49	0
	max Δ%	54.17	112.1	7.92	89.30	0
1-6	Time(sec)	0.18	0.18	0.12	0.12	0.12
7-12	Time(sec)	0.66	0.66	0.34	0.34	0.34

6.3 Wilcoxon rank test

Another way to interpret the algorithm's performance is Wilcoxon ranking, which clarifies how one algorithm performs against another by comparing the cost of the resulting path registered by them through experiment instances. Thus, a Positive result is obtained when the cost exceeds the other algorithms. The negative result is when it costs less, and the tie result is obtained when it registers the exact cost. It's intuitive upon suggested routing policies of our work that the Naïve routing policy will yield the minimum cost. In contrast, the Robust routing policy costs the maximum, and these results are clearly shown in Table 6. The introduced Hybrid routing policy tends to be robust compared to Online and Optimal routing policies, with a higher average to get more significant than it.

Table 6: different routing policies with respect to Wilcoxon rank.

	(%)	Robust	Online	Hybrid	Optimal
Naïve	Positive	0	0	0	0
	Negative	99.17	86.5	94.17	86.5
	Ties	0.83	13.5	5.83	13.5
Robust	Positive		95.42	90.58	91.42
	Negative		0	2.83	0
	Ties		4.58	6.59	8.58
Online	Positive			1.17	0
	Negative			58.67	48.75
	Ties			40.16	51.25
Hybrid	Positive				49.67
	Negative				22.92
	Ties				27.42

Comparing the results from Table 5) with [20], the cost for all routing policies, our results are better Table 7)

Table 7: The comparison table with [Sever, D 2013]

	Naïve	Robust	Online	Hybrid	Optimal
Ours [best]	18.80	31.38	23.41	25.57	24.10
[Sever, D 2013]	51.79	35.25	28.25	27.13	27.13

7 List of contributions after challenges.

1- One of the paramount challenges pertains to the approach for computing the ultimate cost while transitioning across disruption levels within the online algorithm, a factor that significantly influences the eventual outcomes. The conclusive resolution was achieved by introducing the ETC factor.

8 Conclusions

This work depends on the stochastic network, where we find the shortest path between the source and the end. The proposed method uses the Markov Decision Process (MDP), which deals with different disruption levels. The transition between them is at the normal level. By moving either by increasing by one level or decreasing by one level, here the transition process is formed randomly, reducing the complexity level in the calculations, which in turn makes routing algorithms work correctly and regularly with more accuracy. Routing algorithms that use the online protocol work well for the accuracy of the data that we deal with, especially with networks that contain large-scale connections. Some protocols that do not support disruption levels, which are the naive and robust protocols, are protocols that deviate from accurate results due to a lack of taking into account the disruption levels, so the results are inaccurate. They were taken to compare the work with them and know the accuracy of their impact. And there are Hybrid protocols in the case of unavailability of data due to losses of connection online due to Internet interruption or any emergency circumstance that prevents access to the data provided by the service. Extended MDP use of online and hybrid routing algorithms can effectively increase or decrease the travel time depending on the time-dependent inactivation level of the broken links, which reflects the quality of the proposed method in solving the shortest path and optimization problem in stochastic networks. Routing experiments on small and large networks with outage rates and different vulnerabilities support the work. The results of experiments on networks showed that a ***basic fact*or (*ETC*)** should be added to calculate the exact time, which is an added factor to the rational return to predict real-time. It helped to give high accuracy. Getting to the desired end work has been meaningfully justified in real-time accounts. Finally, the Wilcoxon rank test was executed on the suggested experiment, substantiating the efficacy of incorporating the ***ETC*** factor into the equations employed for cost computation.

Conflict of interest

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

References

[1] Feng, G., Su, G., & Sun, In Proceedings of the 3rd ACM SIGSPATIAL "Optimal route of emergency resource scheduling based on GIS" Workshop on Emergency Management using (pp. 1-5), Z. (Novembre, 2017).

[2] Krishnan, K. R. " Markov decision algorithms for dynamic routing (telephone networks) ", IEEE Communications Magazine, 28(10), 66-69 (1990).

[3] Ghiani, G., Manni, E., & Thomas, B. W. "A comparison of anticipatory algorithms for the dynamic and stochastic traveling salesman problem “Transportation Science, 46(3), 374-387 (2012).

[4] R Javid, R. J., & Javid, R. J., " A framework for travel time variability analysis using urban traffic incident data", IATSS research, 42(1), 30-38 (2018).

[5] Zheng, W., Thangeda, P., Savas, Y., & Ornik, M. " Optimal Routing in Stochastic Networks with Reliability Guarantees" IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, 2021-September, 3521–3526. (2021).

[6] Su, Z., Chow, A. H., Fang, C., Liang, E., & Zhong, R. "Hierarchical control for stochastic network traffic with reinforcement learning" Transportation Research Part B: Methodological, 167, 196-216. (2023).

[7] Z. Zhang and M. Li, "Finding Paths with Least Expected Time in Stochastic Time-Varying Networks Considering Uncertainty of Prediction Information," in IEEE Transactions on Intelligent Transportation Systems, 10.1109/TITS.3299277 (2023).

[8] Liu, H., Zhao, F., Wang, M., Qiu, J., & Chen, X. "Finite-time pinning impulsive synchronization of stochastic complex networks with mixed delays" Applied Mathematical Modelling, 117, 840-850 (2023).

[9] Prakash, A. A. "Algorithms for most reliable routes on stochastic and time-dependent networks" Transportation Research Part B: Methodological, 138, 202-220 (2020).

[10] Hall, R. W. "The fastest path through a network with random time-dependent travel times" Transportation science, 20(3), 182-188 (1986).

[11] Silver, D. Lecture 2: "Markov decision processes" UCL. Retrieved from www0. cs. ucl. ac. uk/staff/d. silver/web/Teaching files/MDP. pdf. "(2015).

[12] Ulmer, M. W., Goodson, J. C., Mattfeld, D. C., & Thomas, B. W. "On modelling stochastic dynamic vehicle routing problems" EURO Journal on Transportation and Logistics, 9(2), 100008 (2020).

[13] K. Gatie, S. O. Yew, C. Teasu and S. T. Puay, "Solving the Dynamic Vehicle Routing Problem Under Traffic Congestion," IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, (2016).

[14] White, D. J. "A survey of applications of Markov decision processes" Journal of the operational research society, 44(11), 1073-1096 (1993).

[15] M. W. Ulmer. D. C. Mattfeld and F. Köster, "Budgeting Time for Dynamic Vehicle Routing with Stochastic Customer Requests," Transportation Science, vol. 52, no. 1, (2017).

[16] D. Sever, L. Zhao, N. Dellaert and T. Van Woen, "Dynamic Shortest Path Problem with Travel-Time-Dependent Stochastic Disruptions Transportation Research Part C: Emerging Technologies 92, pp. 42-57 (2013).

[17] Thomas, B. W., & White III, C. C., " The dynamic shortest path problem with anticipation" European journal of operational research, 176(2), 836-854 (2007).

[18] Thangeda, P., & Ornik, M. PROTRIP: "Probabilistic risk-aware optimal transit planner" In IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC) (pp. 1-6). IEEE (September, 2020).

[19] Bertsekas, D. P., & Tsitsiklis, J. N. " An analysis of stochastic shortest path problems" Mathematics of Operations Research 16(3), 580-595 (1991).

[20] Sever, D., Dellaert, N., Van Woensel, T., & De Kok, T. ,"Dynamic shortest path problems: Hybrid routing policies considering network disruptions" Computers & Operations Research, 40(12), 2852-2863 (2013).