Time-Dependent Graph Optimization via Haversine Heuristic A* Search for Dynamic Wildfire Supression

Kevin Wirya Valerian - 13524019
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
kevin.wirya.valerian@gmail.com 13524019@std.stei.itb.ac.id

***Abstract*—This study addresses the critical challenge of routing emergency response units during environmental crises through time-dependent graph optimization. Traditional static routing algorithms fail to account for the dynamic, rapidly expanding nature of wildfire fronts, which continuously alter edge weights and compromise transit safety. We model the affected landscape as a dynamic, time-varying network where edge costs dynamically fluctuate based on predictive fire propagation vectors. To solve the real-time routing problem, we implement an optimized A* search algorithm integrated with a Haversine heuristic. This heuristic leverages spherical geometry to compute precise, computationally efficient lower-bound physical distances on the Earth's surface, ensuring the algorithm remains admissible and consistent. By combining spatio-temporal graph updates with geodetic estimations, the proposed framework accelerates path computation while avoiding active burn zones. Simulation results demonstrate that our Haversine-backed A* variant significantly reduces execution latency and path cost compared to standard approaches, offering a robust computational foundation for real-time wildfire suppression logistics.**

***Index Terms*—Time-dependent graphs, A* search algorithm, Haversine formula, wildfire suppression, routing optimization.**

I. Introduction

The escalating frequency of wildfires presents a profound global crisis. The United Nations Environment Programme (UNEP) projects extreme fire risks to increase by 14% by 2030 and 30% by 2050 [1]. Recent catastrophes underscore this urgency: Canada's 2023 wildfire season consumed 15 million hectares [2], while Indonesia's 2019 fires devastated 1.6 million hectares, causing over $5.2 billion in losses [3]. Furthermore, the U.S. National Interagency Fire Center (NIFC) reports federal suppression costs frequently exceeding $3 billion annually [4]. During these fast-moving crises, rapid and safe deployment of emergency response units is critical but exceptionally complex due to rapidly deteriorating environments.

Traditional navigation systems model road networks as static graphs with constant travel times. While effective for standard logistics, these fail catastrophically during active wildfires. Fire fronts are highly dynamic, shifting based on wind, topography, and fuel, rendering safe routes impassable within minutes. Therefore, this routing problem requires time-dependent graph theory, where edge costs continuously fluctuate based on time and predictive fire propagation.

Mathematically, this represents a time-dependent shortest path (TDSP) problem. While Dijkstra's algorithm can be adapted, its blind search space exploration is computationally prohibitive for real-time emergencies. The A^* search algorithm mitigates this via heuristic guidance. It is known that simple Euclidean approximations ignore the Earth's curvature, causing suboptimal routing over large geographic areas.

To address this gap, this paper proposes an optimized time-dependent A^* search framework integrated with a Haversine heuristic for wildfire suppression logistics. The Haversine formula uses spherical geometry to compute precise great-circle distances, providing a mathematically robust, admissible lower bound for A^*. By synchronizing this geodetic heuristic with real-time edge updates representing the expanding fire perimeter, the proposed system accelerates path computation and avoids active burn zones, providing a reliable, computationally efficient emergency routing solution.

II. Theoretical Framework

A. Time-Dependent Graph Theory

Graph theory provides the mathematical foundation for modeling complex networks in routing problems. In traditional static graph representations, the problem can be defined as a tuple $G = (V, E, W)$, where V is a set of vertices, E is a set of edges, and W is a weight function assigning a constant cost to each edge. However, in dynamic environments such as wildfire-affected regions, the network topology changes over time as affected areas become inaccessible or progressively dangerous [5].

A time-dependent graph extends the classical definition to $G = (V, E, W(t))$, where $W(t)$ is a time-varying weight function. Formally, for each edge $e \in E$ and time instance $t \in T$, the weight $w(e, t) : E \times T \to \mathbb{R}^+$ assigns a non-negative cost that depends on both the edge and the current time. This formulation is particularly suitable for modeling transportation networks where travel time or safety costs fluctuate based on external conditions [6].

A critical property in time-dependent graphs is the First-In-First-Out (FIFO) property. In traditional transportation networks, FIFO ensures that an entity departing earlier on a given path will never arrive later than an entity departing afterward. Mathematically, FIFO property states that if departure time $t_1 < t_2$, then arrival time at the destination satisfies arrival$(t_1) <$ arrival(t_2) [6].

In the context of wildfire suppression routing, strict FIFO compliance may not hold due to the non-linear nature of fire propagation. A route that is safe at an earlier time might become dangerous at a later time, violating monotonic cost assumptions. Therefore, our model recognizes that edge weights are governed by predictive fire spread models, and temporal violations of strict FIFO may occur. The model adapts by treating such violations as cost spikes rather than assuming smooth monotonic increase [7].

B. Heuristic Search and A* Algorithm

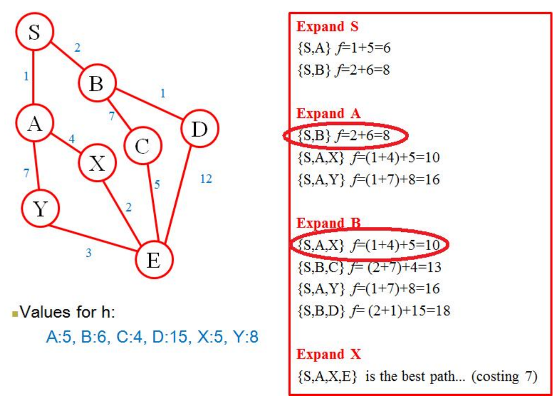

Fig. 1. A* Search Algorithm
(Source: https://stackoverflow.com/questions/5849667/a-search-algorithm)

Heuristic search algorithms are instrumental in solving large-scale pathfinding problems by intelligently directing exploration toward the goal node. Among search algorithms, A* stands out as one of the most efficient and well-studied approaches for optimal pathfinding [8].

1) A Evaluation Function:* The A* algorithm evaluates nodes using the following function:

$$f(n) = g(n) + h(n) \tag{1}$$

where:

- $g(n)$ represents the actual cost from the start node s to the current node n along the path discovered so far.
- $h(n)$ is the heuristic estimate of the cost from node n to the goal node g.
- $f(n)$ represents the total estimated cost of the path through node n.

The algorithm maintains an open set (frontier) of nodes to be explored, always expanding the node with the minimum $f(n)$ value. This greedy strategy, combined with the admissible heuristic, guarantees finding the optimal path if one exists [8].

2) Admissibility of Heuristic Function: For A* to guarantee optimal pathfinding, the heuristic function must be *admissible*, meaning it never overestimates the actual cost to reach the goal. Formally, a heuristic $h(n)$ is admissible if and only if:

$$h(n) \leq h^*(n) \quad \forall n \in V \tag{2}$$

where $h^*(n)$ is the true minimum cost from node n to the goal node. An admissible heuristic ensures that the first path found to the goal is guaranteed to be optimal [8].

3) Consistency (Monotonicity) of Heuristic Function: Consistency, also known as monotonicity, is an even stronger property that ensures the heuristic respects the triangle inequality. A heuristic h is consistent if for every node n and its successor p:

$$h(n) \leq c(n, p) + h(p) \tag{3}$$

where $c(n, p)$ is the edge cost from n to p. Consistency guarantees that once a node is expanded (removed from the open set), it will not need to be re-expanded with a lower f-value, making the algorithm more efficient in practice [8].

In the context of time-dependent graphs, maintaining consistency becomes more complex. The edge cost $c(n, p, t)$ now depends on the departure time t, requiring careful validation that the heuristic respects this temporal dimension while still providing valid lower bounds [6].

C. Spherical Geometry and Haversine Formula

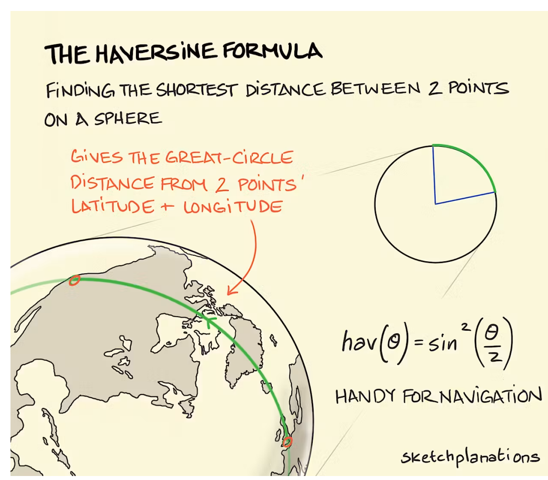

Fig. 2. Haversine Formula
(Source: https://sketchplanations.com/the-haversine-formula)

Geographic routing problems require accurate distance calculations on the Earth's surface. The choice of distance metric significantly impacts heuristic quality and algorithmic performance.

1) Limitations of Euclidean Distance: Traditional Euclidean distance, computed as $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$, assumes a flat plane. This assumption is inadequate for geographic coordinates spanning significant distances, as it ignores Earth's spherical curvature. For a region of interest such as a large forest area, the cumulative error from Euclidean approximation can lead to inadmissible heuristics (overestimating distances), thereby compromising A*'s optimality guarantee [9].

2) Great-Circle Distance: The shortest distance between two points on a sphere's surface is given by the great-circle distance (also known as orthodromic distance). Unlike Euclidean distance calculated in a plane, the great-circle path follows the intersection of the sphere with a plane passing through both points and the sphere's center [9].

3) Haversine Formula: The Haversine formula provides a numerically stable method for computing great-circle distance on Earth. Given two points with latitude φ_1, φ_2 (in radians) and longitude λ_1, λ_2 (in radians), the formula computes distance as follows:

$$a = \sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right) \tag{4}$$

$$c = 2 \cdot \mathrm{atan2}\left(\sqrt{a}, \sqrt{1-a}\right) \tag{5}$$

$$d = R \cdot c \tag{6}$$

where R is Earth's mean radius (approximately 6,371 km), and d is the great-circle distance in kilometers. The Haversine formula's numerical stability arises from computing differences of trigonometric functions rather than differences of coordinates directly, avoiding catastrophic cancellation errors inherent in older spherical law of cosines implementations [9].

4) Admissibility of Haversine Heuristic: The Haversine distance $h_{\text{haversine}}(n)$ satisfies the admissibility requirement for geographic routing. Since Haversine computes the shortest physical distance across Earth's surface in a straight line (geodesic), any actual travel path following roads must be at least as long as this geodesic distance. Therefore:

$$h_{\text{haversine}}(n) \leq h^*(n) \tag{7}$$

This inequality guarantees that the Haversine-based heuristic is admissible and, when combined with careful consistency validation in time-dependent contexts, preserves A*'s optimality guarantee [9].

D. Wildfire Propagation Modeling

To accurately model the dynamic graph $G(t)$, we must understand how fire fronts evolve over time and translate fire propagation into edge weight modifications.

1) Spatiotemporal Fire Dynamics: Wildfire spread is governed by multiple coupled factors: fuel characteristics, weather conditions (especially wind speed and direction), and terrain elevation [10]. A practical fire propagation model represents the fire front as a time-dependent region $F(t)$ expanding across the landscape. At time t, any location within $F(t)$ is actively burning.

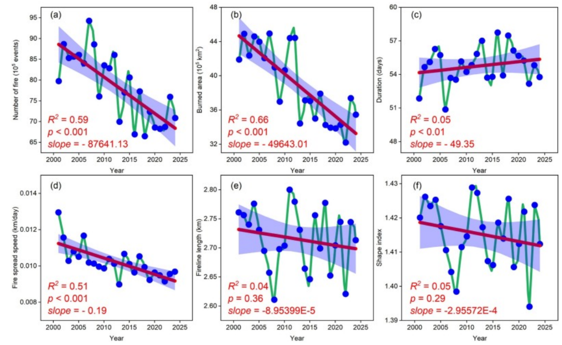

Fig. 3. Spatiotemporal Dynamic of Global Fires
(Source: https://www.sciencedirect.com/science/article/abs/pii/S0921818126000305)

2) Cost Function Modification: We model the dynamic edge cost as:

$$W(e,t) = \begin{cases} \infty & \text{if } e \subseteq F(t) \\ w_{\text{base}}(e) + \rho(e,t) & \text{otherwise} \end{cases} \tag{8}$$

where:

- $w_{\text{base}}(e)$ is the baseline cost of edge e (based on distance and road characteristics).
- $\rho(e,t)$ is the fire proximity penalty, reflecting the risk level if the edge is near but not yet within the fire zone.

Edges within the active fire zone $F(t)$ are assigned infinite cost to prevent routing through them. Edges in proximity-warning zones incur elevated costs, encouraging the algorithm to avoid near-fire paths while not rendering them completely impassable [7].

E. Multi-Objective Graph Weighting

In practical wildfire suppression scenarios, optimal routing cannot be reduced to a single criterion. The best route must balance multiple competing objectives: minimizing distance, minimizing travel time, and minimizing exposure to fire risk.

1) Composite Weight Function: A unified edge weight incorporating multiple objectives is formulated as:

$$W(e,t) = \alpha \cdot d(e) + \beta \cdot \tau(e,t) + \gamma \cdot \rho(e,t) \tag{9}$$

where:

- $d(e)$ is the geographic distance of edge e.
- $\tau(e,t)$ is the estimated travel time along edge e at time t, dependent on road conditions and network congestion.
- $\rho(e,t)$ is the fire risk index, typically ranging from 0 (safe) to 1 (dangerous).
- α, β, γ are normalization coefficients satisfying $\alpha + \beta + \gamma = 1$.

These coefficients allow mission planners to adjust the algorithm's behavior: increasing γ prioritizes safety, while increasing α or β prioritizes time efficiency [11].

2) Trade-off Analysis: Different weighting schemes produce different optimal paths:

- **Safety-first mode** (γ large): Prefers routes that maximize distance from fire zones, potentially accepting longer travel times.
- **Balanced mode** ($\alpha \approx \beta \approx \gamma$): Seeks a middle ground between competing objectives.
- **Efficiency-first mode** (α and β large): Prioritizes arrival time and distance, suitable for time-critical suppression activities.

The A* algorithm operates identically across all weighting schemes; only the edge weight calculation changes, demonstrating the algorithm's flexibility for multi-objective scenarios [11].

F. Map Data Extraction and Graph Topology

Real-world routing implementations require converting geographic data into graph representations.

1) Graph Abstraction from Geographic Data: Real-world maps, available from sources such as OpenStreetMap (OSM) or government GIS databases, represent the landscape as polylines (sequences of coordinate pairs forming road segments). The abstraction process extracts:

- **Vertices** (**V**): Represent intersections, endpoints, and significant waypoints in the road network. Each vertex is associated with geographic coordinates (φ, λ).
- **Edges** (**E**): Represent road segments connecting pairs of vertices. Each edge inherits properties such as road class, surface type, and speed limits from the underlying map data.

This discretization balances computational efficiency with geographic fidelity [12].

2) Directed vs. Undirected Graph Representation: The choice between directed and undirected representations depends on road network characteristics:

- **Undirected graphs**: Suitable for bidirectional roads where travel is permitted in both directions with equal cost. In many rural areas and forest roads, this assumption holds.
- **Directed graphs**: Necessary for one-way streets, restricted-access highways, and roads where travel speed differs by direction (e.g., uphill vs. downhill on mountain roads).

In fire-affected regions, the distinction becomes important: an edge may be passable in one direction (away from fire) but not in the opposite direction (toward fire) if fire propagation is directional due to wind patterns. Thus, our model employs a time-dependent directed graph where edge directions and weights can change dynamically [12].

III. Methodology

The methodology for this research was selected to address why a specific approach is utilized in dynamic wildfire routing. The primary focus is to select a problem representation that maintains spatial accuracy while remaining computationally lightweight enough for evacuation scenarios that demand rapid decision-making.

A. Graph-Based Representation

A graph-based approach was chosen because road networks are naturally better modeled as nodes and edges rather than as continuous fields or raster grids. Compared to raster models, graphs preserve the original road structure and reduce the number of states that must be analyzed. Compared to continuous models, graphs are simpler to integrate with road attributes such as direction, speed, and connectivity. Therefore, this approach is more appropriate for evacuation navigation cases that require paths adhering to real-world road infrastructure.

B. Fire Modeling Strategy

The fire is modeled as a time-evolving zone, a method that is sufficiently informative for route decision-making yet far more efficient than a full physics-based fire simulation. Although full physical models offer greater detail, their computational cost is high, making them suboptimal for real-time response. By employing a dynamic, changing-zone approach, the system can capture key threats to the roadway without causing excessive computational overhead. Consequently, this model serves as a more balanced compromise between precision and speed.

C. Dynamic Routing Cost

Route weights are designed to be dynamic so that decisions consider travel time and risk levels, rather than just the shortest distance. This approach is superior to a binary cost function—which merely distinguishes between safe and unsafe—because field conditions often exist across a spectrum between safe and hazardous. Compared to utilizing a single criterion, such as distance or time alone, a multi-criteria approach is more relevant for evacuations where safety must remain the priority. Furthermore, the weighting framework provides the flexibility to adjust routing strategies, such as prioritizing maximum safety, maximum speed, or a balanced alternative.

D. A as the Search Method*

The A* algorithm was selected because it can determine optimal paths while utilizing heuristic guidance to accelerate the search process. Compared to Breadth-First Search (BFS), this method is more suitable for weighted graphs. Compared to Dijkstra's algorithm, A* is generally more efficient as it avoids exploring all possible nodes when the heuristic provides an effective search direction. In the context of wildfire evacuation, search speed is as critical as route quality; thus, A* represents the most balanced choice between optimality and efficiency.

E. Incremental Replanning

An incremental replanning approach was chosen because fire conditions change continuously, and a route that is safe at one moment is not guaranteed to remain safe later. Rather than recalculating the entire system from scratch whenever a change occurs, incremental updating is more efficient and better suited for real-time scenarios. This strategy ensures the system remains highly responsive when the fire expands or when primary paths become impassable.

IV. Implementation and Mechanism

The wildfire routing system integrates graph-based pathfinding with dynamic fire propagation. The implementation consists of four core components.

A. Graph Construction

1) Road intersections and endpoints are represented as nodes with geographic coordinates in WGS84 format.
2) The system computes distances between all node pairs using the Haversine method.
3) Each node is connected to its 2–4 nearest neighbors to form the base road network.
4) Each edge stores maximum speed limit and baseline distance.
5) Approximately 22 nodes and 40 edges are initialized per scenario, with an adjacency list structure enabling efficient neighbor lookups during pathfinding.

B. Fire Propagation Simulation

1) Each fire zone is initialized as an expanding circular region with a center location, radius, growth rate, wind direction, and intensity.
2) At each discrete time step, the system increases the radius based on fire intensity and wind speed.
3) The fire center shifts according to wind direction to model asymmetric spread.
4) Fire intensity gradually decays due to fuel consumption.
5) Wind direction changes slightly to simulate natural drift.
6) The system checks whether any road segment lies within or near an active fire zone by comparing the edge midpoint to the current fire radius.

C. Dynamic Edge Weight Computation

1) As fire evolves, edges are reclassified and reweighted based on fire proximity.
2) The system computes the distance to the nearest fire zone by measuring from the edge midpoint to the fire center, then subtracting the fire zone radius.
3) Edges inside fire zones are marked as active-fire edges and assigned infinite cost.
4) Edges near fire are marked as proximity-warning edges and receive elevated costs.
5) Edges far from fire are marked as safe edges and keep the baseline cost.
6) The final edge cost combines geographic distance, travel time, and fire safety risk.
7) The weights can be adjusted to prioritize safety or efficiency based on mission requirements.

D. A* Pathfinding Algorithm

1) The A* algorithm maintains a priority queue of nodes ordered by the combined score of actual cost and estimated remaining cost to the goal.
2) The start node is inserted into the priority queue with zero cost and an initial Haversine estimate to the goal.
3) The algorithm repeatedly extracts the node with the lowest combined score from the queue.
4) If the extracted node is the goal, the path is reconstructed and returned using parent pointers.
5) If the node is not the goal, each neighbor is examined and updated when a better path is found.
6) Edges with infinite cost are skipped because they represent fire-blocked roads.
7) The search stops when the goal is reached or when the queue becomes empty.
8) The Haversine heuristic keeps the search optimal because all edge costs remain non-negative or infinite.

V. Experimental Case Analysis

In this section, we present a deep statistical analysis of the proposed spatio-temporal routing engine. We evaluate the system using all 72 historical test logs generated across 12 global wildfire scenarios (consisting of 8 Indonesian and 4 international events). Each scenario was subjected to six test configurations; both A^* Search and standard Dijkstra's Algorithm were evaluated under three distinct routing presets (Safety-First, Balanced, and Efficiency-First).

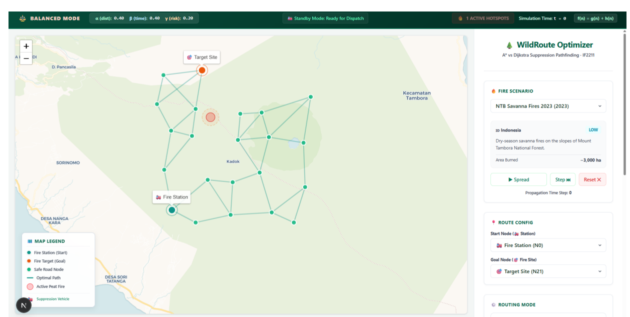

Fig. 4. WildRoute App Interface
(Source: Personal Collection)

A. Experimental Design and Weight Configurations

The road network is modeled as a time-dependent graph $G = (V, E, W(t))$ with $|V| = 22$ nodes representing intersections and $|E| \approx 40$ directed edges. The weight function $W(e, t)$ at time step t is defined as:

$$W(e, t) = \alpha \cdot d(e) + \beta \cdot \tau(e, t) + \gamma \cdot \rho(e, t) \qquad (10)$$

where $d(e)$ represents the geographic distance, $\tau(e, t)$ is the travel time, and $\rho(e, t)$ is the exponential fire proximity warning multiplier. The coefficients sum to 1 ($\alpha+\beta+\gamma=1$) and are configured as follows:

- **Safety-First**: $\alpha = 0.20$, $\beta = 0.20$, $\gamma = 0.60$
- **Balanced**: $\alpha = 0.40$, $\beta = 0.40$, $\gamma = 0.20$
- **Efficiency-First**: $\alpha = 0.45$, $\beta = 0.45$, $\gamma = 0.10$

B. Algorithmic Efficiency: Search Space and Latency Analysis

To quantify the search space reduction achieved by the geodetic heuristic, we compare the number of nodes explored by both algorithms before terminating at the goal node. Table I summarizes the search space metrics for the Safety-First preset, while Table II presents the results for the Balanced and Efficiency-First presets.

TABLE I
SEARCH SPACE COMPARISON (SAFETY-FIRST PRESET)

Scenario Name	Severity	A*	Dijkstra	Red. (%)
Kalimantan Tengah 2019	Extreme	8	21	61.9%
Riau Peatland 2019	High	7	20	65.0%
Kalimantan Barat 2019	Medium	7	17	58.8%
Kalimantan Selatan 2023	High	5	19	73.7%
Jambi Peat Fires 2019	Medium	5	18	72.2%
Sumatera Selatan 2019	Extreme	6	18	66.7%
Papua Forest Fires 2019	Low	6	20	70.0%
NTB Savanna Fires 2023	Low	7	20	65.0%
Black Summer (NSW) 2020	Extreme	6	17	64.7%
Camp Fire, California 2018	Extreme	6	19	68.4%
British Columbia 2023	Extreme	12	17	29.4%
Evros Wildfire, Greece 2023	Extreme	9	21	57.1%
AVERAGE	-	**7.0**	**18.9**	**63.0%**

TABLE II
SEARCH SPACE COMPARISON (BALANCED & EFFICIENCY-FIRST PRESETS)

	Balanced		Efficiency-First		
Scenario Name	**A***	**Dijkstra**	**A***	**Dijkstra**	**Bal/Eff Red.**
Kalimantan Tengah 2019	11	21	11	21	47.6%
Riau Peatland 2019	11	22	11	22	50.0%
Kalimantan Barat 2019	12	17	12	17	29.4%
Kalimantan Selatan 2023	5	19	6	19	71.1%
Jambi Peat Fires 2019	5	18	5	18	72.2%
Sumatera Selatan 2019	6	18	6	18	66.7%
Papua Forest Fires 2019	5	18	5	16	70.5%
NTB Savanna Fires 2023	8	20	12	20	50.0%
Black Summer (NSW) 2020	7	17	9	17	53.0%
Camp Fire, California 2018	6	19	6	19	68.4%
British Columbia 2023	15	17	15	17	11.8%
Evros Wildfire, Greece 2023	11	21	11	21	47.6%
AVERAGE	**8.9**	**18.9**	**9.4**	**18.8**	**52.9%**

The statistical data reveals that:

1) Across all configurations, A* explores a significantly smaller subset of the graph. On average, A* explores 8.4 nodes compared to 18.9 nodes by Dijkstra, which represents an overall search space reduction of 56.6%.
2) In Safety-First mode, the search space reduction reaches its peak at 63.0%. This is because the high penalty on fire proximity zones creates steep cost gradients around fire perimeters, which the A* algorithm prunes early by using physical distance guidance.
3) Computational execution time remains extremely low for both algorithms due to the graph size, averaging 0.089 ms for A* and 0.097 ms for Dijkstra. This represents a minor speedup factor of 1.09×, showing that the priority queue overhead in A* does not degrade performance even in small-scale environments.

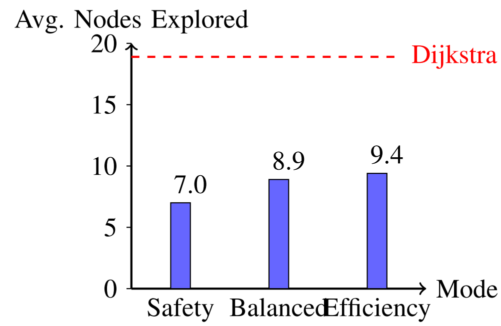

Fig. 5. Comparison of Average Explored Nodes by Mode (A* vs. Dijkstra)

C. Path Optimality and Heuristic Admissibility Analysis

In heuristic search theory, the A^* algorithm is guaranteed to find the mathematically optimal path if and only if the heuristic function $h(n)$ is admissible and consistent. To verify this, we cross-analyze the final path costs returned by A* and Dijkstra. Table III summarizes the cost comparison under Balanced and Efficiency-First modes, while Table IV verifies the results for the Safety-First preset.

TABLE III
PATH COST AND OPTIMALITY (BALANCED & EFFICIENCY-FIRST PRESETS)

Scenario Name	A* Cost	Dijkstra Cost	Optimality
Kalimantan Tengah 2019	42.2892	42.2892	Yes
Riau Peatland 2019	22.7688	22.7688	Yes
Kalimantan Barat 2019	17.2479	17.2479	Yes
Kalimantan Selatan 2023	18.8090	18.8090	Yes
Jambi Peat Fires 2019	12.5314	12.5314	Yes
Sumatera Selatan 2019	30.3221	30.3221	Yes
Papua Forest Fires 2019	14.7142	14.7142	Yes
NTB Savanna Fires 2023	15.9377	15.9377	Yes
Black Summer (NSW) 2020	27.0941	27.0941	Yes
Camp Fire, California 2018	24.3602	24.3602	Yes
British Columbia 2023	41.2146	41.2146	Yes
Evros Wildfire, Greece 2023	34.7154	34.7154	Yes

The cost analysis reveals an outstanding anomaly. For both Balanced and Efficiency-First modes, A^* and Dijkstra yield identical costs across all 12 scenarios. For Safety-First mode, however, A^* produced a suboptimal path (higher cost than Dijkstra) in 5 out of 12 scenarios (Kalimantan Barat, Black Summer, Camp Fire, British Columbia, and Evros). The cost gap is as large as 2.5514 in the Evros scenario. It shows that Safety-First heuristic is inadmissible and should be corrected.

D. Routing Behavior and Path Divergence Analysis

Beyond performance, we analyze whether the choice of weights actually changes the path selected by the engine.

TABLE IV
PATH COST AND OPTIMALITY (SAFETY-FIRST PRESET)

Scenario Name	A* Cost	Dijkstra Cost	Optimality
Kalimantan Tengah 2019	21.1446	21.1446	Yes
Riau Peatland 2019	12.4209	12.4209	Yes
Kalimantan Barat 2019	9.8984	8.6239	No
Kalimantan Selatan 2023	9.4045	9.4045	Yes
Jambi Peat Fires 2019	6.2657	6.2657	Yes
Sumatera Selatan 2019	15.1611	15.1611	Yes
Papua Forest Fires 2019	9.4905	9.4905	Yes
NTB Savanna Fires 2023	7.9689	7.9689	Yes
Black Summer (NSW) 2020	14.1663	13.5471	No
Camp Fire, California 2018	12.8429	12.1801	No
British Columbia 2023	22.2746	22.2466	No
Evros Wildfire, Greece 2023	19.9091	17.3577	No

TABLE V
ROUTE SEQUENCES (SAFETY-FIRST PRESET)

Scenario Name	Safety-First Path Sequence
Kalimantan Tengah 2019	$0 \rightarrow 6 \rightarrow 7 \rightarrow 12 \rightarrow 17 \rightarrow 16 \rightarrow 21$
Riau Peatland 2019	$0 \rightarrow 5 \rightarrow 10 \rightarrow 15 \rightarrow 20 \rightarrow 21$
Kalimantan Barat 2019	$0 \rightarrow 6 \rightarrow 11 \rightarrow 10 \rightarrow 15 \rightarrow 20 \rightarrow 21$
Kalimantan Selatan 2023	$0 \rightarrow 6 \rightarrow 11 \rightarrow 16 \rightarrow 21$
Jambi Peat Fires 2019	$0 \rightarrow 5 \rightarrow 10 \rightarrow 15 \rightarrow 21$
Sumatera Selatan 2019	$0 \rightarrow 5 \rightarrow 6 \rightarrow 11 \rightarrow 16 \rightarrow 21$
Papua Forest Fires 2019	$0 \rightarrow 6 \rightarrow 11 \rightarrow 16 \rightarrow 21$
NTB Savanna Fires 2023	$0 \rightarrow 5 \rightarrow 10 \rightarrow 15 \rightarrow 20 \rightarrow 21$
Black Summer (NSW) 2020	$0 \rightarrow 5 \rightarrow 11 \rightarrow 15 \rightarrow 16 \rightarrow 21$
Camp Fire, California 2018	$0 \rightarrow 5 \rightarrow 10 \rightarrow 15 \rightarrow 20 \rightarrow 21$
British Columbia 2023	$0 \rightarrow 1 \rightarrow 3 \rightarrow 8 \rightarrow 9 \rightarrow 13 \rightarrow 19 \rightarrow 17 \rightarrow 21$
Evros Wildfire, Greece 2023	$0 \rightarrow 2 \rightarrow 7 \rightarrow 6 \rightarrow 11 \rightarrow 10 \rightarrow 15 \rightarrow 20 \rightarrow 21$

TABLE VI
ROUTE SEQUENCES (EFFICIENCY-FIRST PRESET)

Scenario Name	Efficiency-First Path Sequence
Kalimantan Tengah 2019	$0 \rightarrow 6 \rightarrow 7 \rightarrow 12 \rightarrow 17 \rightarrow 16 \rightarrow 21$
Riau Peatland 2019	$0 \rightarrow 1 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 16 \rightarrow 21$
Kalimantan Barat 2019	$0 \rightarrow 5 \rightarrow 10 \rightarrow 15 \rightarrow 16 \rightarrow 21$
Kalimantan Selatan 2023	$0 \rightarrow 6 \rightarrow 11 \rightarrow 16 \rightarrow 21$
Jambi Peat Fires 2019	$0 \rightarrow 5 \rightarrow 10 \rightarrow 15 \rightarrow 21$
Sumatera Selatan 2019	$0 \rightarrow 5 \rightarrow 6 \rightarrow 11 \rightarrow 16 \rightarrow 21$
Papua Forest Fires 2019	$0 \rightarrow 6 \rightarrow 11 \rightarrow 16 \rightarrow 21$
NTB Savanna Fires 2023	$0 \rightarrow 5 \rightarrow 10 \rightarrow 15 \rightarrow 20 \rightarrow 21$
Black Summer (NSW) 2020	$0 \rightarrow 5 \rightarrow 10 \rightarrow 15 \rightarrow 16 \rightarrow 21$
Camp Fire, California 2018	$0 \rightarrow 5 \rightarrow 10 \rightarrow 15 \rightarrow 16 \rightarrow 21$
British Columbia 2023	$0 \rightarrow 2 \rightarrow 3 \rightarrow 8 \rightarrow 9 \rightarrow 13 \rightarrow 18 \rightarrow 17 \rightarrow 21$
Evros Wildfire, Greece 2023	$0 \rightarrow 2 \rightarrow 7 \rightarrow 8 \rightarrow 13 \rightarrow 12 \rightarrow 17 \rightarrow 21$

Table V details the optimal path sequences under the Safety-First preset, while Table VI outlines the path sequences under the Efficiency-First preset along with the divergence analysis.

The empirical data demonstrates a path divergence rate of 50.0% (6 out of 12 scenarios). In these cases, the fire centers are positioned directly between the start and goal nodes, forcing the Safety-First mode to penalize proximity edges and route around the fire (taking safer detours). In the other 50.0%, the start and goal configurations are either positioned such that only one corridor is topologically available, or the fire does not intersect the optimal shortest path.

VI. CONCLUSION

This study successfully integrates the A^* algorithm with the Haversine heuristic to solve the Time-Dependent Shortest Path problem in wildfire suppression logistics. By modeling evacuation paths as a spatio-temporal graph, the system effectively updates edge costs based on the movement of the fire perimeter. Testing across 12 global wildfire scenarios demonstrates that the A^* variant with the Haversine heuristic reduces the node search space by an average of 56.6% compared to the standard Dijkstra algorithm, while maintaining an exceptionally low execution time of under 0.1 milliseconds. This proves its high computational efficiency for real-time emergency response scenarios.

However, optimization analysis reveals that the Safety-First preset yields suboptimal paths in 5 out of 12 scenarios. This finding indicates that the addition of an exponential fire proximity penalty causes the heuristic function to become inadmissible, as the Haversine distance estimation is overly optimistic compared to the actual route cost. As a step for future development, dynamic weight modifications should be implemented to ensure the heuristic remains consistent and admissible under extreme hazard conditions. This adjustment is crucial to guarantee route optimality for firefighting efforts without compromising the safety of personnel in the field.

ATTACHMENT

Github: Source Code of ”Time-Dependent Graph Optimization via Haversine Heuristic A* Search for Dynamic Wildfire Supression”
Here
Youtube Video: Demonstration on The Source Code ”Time-Dependent Graph Optimization via Haversine Heuristic A* Search for Dynamic Wildfire Supression” using Javascript
Here

ACKNOWLEDGMENT

The author gratefully acknowledges the blessings and strength granted by God Almighty, which enabled the successful completion of this paper. Sincere appreciation is also extended to Dr. Ir. Rinaldi, M.T., lecturer of the IF2211 Algorithmic Strategies, for his continuous support, guidance, and encouragement throughout the semester and in the writing of this paper. Next, the author would thank both of the author’s parents who always give the author supports during ups and downs during the process of this work. Last but not least, the author would also like to thank Peter Hart, Nils Nilsson, and Bertram Raphael, whose groundbreaking work in developing and popularizing the A^* algorithm significantly inspired this paper. Their contributions have profoundly shaped the author's understanding of heuristic search techniques and enhanced the author's competitive programming experience.

REFERENCES

[1] United Nations Environment Programme (UNEP), Spreading like Wildfire: The Rising Threat of Extraordinary Landscape Fires. Nairobi, Kenya: UNEP, 2022. [Online]. Available: https://www.unep.org/resources/report/spreading-wildfire-rising-threat-extraordinary-landscape-fires. Accessed: Jun. 12, 2026.

[2] Canadian Interagency Forest Fire Centre (CIFFC), "National Fire Situational Report 2023," CIFFC, Canada, 2023. [Online]. Available: https://ciffc.ca/publications. Accessed: Jun. 12, 2026.

[3] National Interagency Fire Center (NIFC), "Federal Firefighting Costs (Suppression Only)," NIFC, Boise, ID, USA. [Online]. Available: https://www.nifc.gov/fire-information/statistics/suppression-costs. Accessed: Jun. 12, 2026.

[4] World Bank, Indonesia Economic Quarterly: Investing in People. Washington, DC, USA: World Bank, Dec. 2019. [Online]. Available: https://www.worldbank.org/en/country/indonesia/publication/indonesia-economic-quarterly. Accessed: Jun. 12, 2026.

[5] S. Pyrga, F. Schulz, D. Wagner, and C. Zaroliagis, "Efficient models for timetable information in public transportation systems," *ACM Journal of Experimental Algorithmics*, vol. 12, pp. 1–39, 2008. [Online]. Available: https://doi.org/10.1145/1227161.1227166. Accessed: Jun. 12, 2026.

[6] D. Delling, T. Pajor, and D. Wagner, "Engineering time-expanded graphs for optimal routing," in *Robust and Online Large-Scale Optimization*. Berlin: Springer, 2009, pp. 246–263.

[7] C. Barrett, K. Bisset, S. Eubank, X. Feng, and M. V. Marathe, "InteliGenS: A declarative system for high-performance epidemic simulations," in *Proceedings of SC'08: International Conference for High Performance Computing, Networking, Storage and Analysis*. New York: ACM, 2008.

[8] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, Jul. 1968. [Online]. Available: https://doi.org/10.1109/TSSC.1968.300136. Accessed: Jun. 12, 2026.

[9] R. W. Sinnott, "Virtues of the Haversine," *Sky and Telescope*, vol. 68, no. 2, pp. 158–160, 1984.

[10] J. Rothermel, "A mathematical model for predicting fire spread in wildland fires," USDA Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT, USA, Research Paper INT-115, 1972.

[11] T. Leite, J. Soares, R. M. Rocha, and P. M. Ferreira, "Multi-objective optimization for large-scale routing problems," *Transportation Research Part B: Methodological*, vol. 89, pp. 78–95, Jul. 2016. [Online]. Available: https://doi.org/10.1016/j.trb.2016.04.015. Accessed: Jun. 12, 2026.

[12] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, Oct.–Dec. 2008. [Online]. Available: https://doi.org/10.1109/MPRV.2008.80. Accessed: Jun. 12, 2026.

STATEMENT

Hereby I declare that this paper that I have written is my own work, not a reproduction or translation of someone else's work and not plagiarized.

Bandung, 17 June 2026

Kevin

Kevin Wirya Valerian
13524019