

# Dynamic Risk-Aware A\* Path Planning System

## 1 Introduction

This document describes the complete mathematical and algorithmic foundation of the A\* (A-Star) pathfinding algorithm adapted for a dynamic fire-evacuation system.

The objective of the system is to compute the safest path from every room in a building to the nearest safe destination (exit or safe room), using real-time sensor data.

The algorithm runs periodically (for example, every 5 seconds), updates risk values based on sensors, and produces optimal evacuation paths.

## 2 Graph Model of the Building

The building floor is modeled as a weighted graph:

$$G = (V, E)$$

Where:

- $V$  = set of nodes (rooms, corridor junctions, exits, safe rooms)
- $E$  = set of edges (walkable connections)

Each edge  $(u, v) \in E$  has base distance:

$$D(u, v)$$

Each node  $v \in V$  has:

- Coordinates  $(x_v, y_v)$
- Sensor values
- Computed risk value  $R(v)$

### 3 Sensor Inputs

Each node contains real-time sensor data:

- Flame percentage:  $F(v) \in [0, 100]$
- Smoke percentage:  $S(v) \in [0, 100]$
- Temperature in °C:  $T(v)$

### 4 Risk Function

#### 4.1 Hard Safety Cutoff

If flame exceeds a critical threshold:

$$F(v) > F_{\text{critical}} \Rightarrow R(v) = \infty$$

#### 4.2 Continuous Risk Model

If flame is below the critical threshold:

$$R(v) = \alpha S(v) + \beta \max(0, T(v) - T_{\text{safe}}) + \gamma F(v)$$

Where:

- $\alpha$  = smoke weight
- $\beta$  = temperature weight
- $\gamma$  = flame weight
- $T_{\text{safe}}$  = safe temperature threshold

### 5 Objective of A\*

A\* minimizes the evaluation function:

$$f(n) = g(n) + h(n)$$

Where:

- $g(n)$  = accumulated cost from start to node  $n$
- $h(n)$  = estimated remaining cost to goal

## 6 Accumulated Cost

For a transition from node  $u$  to node  $v$ :

$$g(v) = g(u) + D(u, v) + R(v)$$

Total path cost:

$$g(n) = \sum D(u_i, u_{i+1}) + \sum R(u_{i+1})$$

## 7 Heuristic Function

Euclidean distance is used:

$$h(n) = \min_{g \in G} \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2}$$

This heuristic is admissible and ensures optimality.

## 8 A\* Algorithm Steps

### 8.1 Initialization

Given start node  $s$ :

$$g(s) = 0$$

$$f(s) = h(s)$$

Open Set =  $\{s\}$  Closed Set =  $\emptyset$

### 8.2 Main Loop

While Open Set is not empty:

1. Select node  $u$  with minimum  $f(u)$ .
2. Remove  $u$  from Open Set.
3. Add  $u$  to Closed Set.
4. If  $u$  is goal, terminate.

### 8.3 Neighbor Expansion

For each neighbor  $v$  of  $u$ :

$$g_{\text{tentative}} = g(u) + D(u, v) + R(v)$$

If  $v$  not in Open Set, add it and set parent.

If:

$$g_{\text{tentative}} < g(v)$$

Update  $g(v)$  and parent pointer.

Then:

$$f(v) = g(v) + h(v)$$

## 9 Termination

The algorithm stops when the goal node is removed from the Open Set.

At that moment:

$$g(\text{goal}) = \text{minimum total cost}$$

## 10 Path Reconstruction

Starting from goal:

Follow parent pointers backward:

$$\text{goal} \rightarrow \text{parent}(\text{goal}) \rightarrow \dots \rightarrow \text{start}$$

Reverse sequence to obtain final path.

## 11 Multiple Start Nodes

For every room  $r_i$ :

- Start =  $r_i$
- Goals = exits  $\cup$  safe rooms
- Run A\*

This produces one optimal evacuation path per room.

## 12 System Execution Cycle

Every 5 seconds:

1. Collect sensor data.
2. Compute  $R(v)$  for all nodes.
3. Update cost structure.
4. Run A\* for each room.
5. Convert paths to LED instructions.
6. Send commands to ESP modules.

## 13 Mathematical Summary

Risk:

$$R(v) = \begin{cases} \infty, & F(v) > F_{\text{critical}} \\ \alpha S(v) + \beta \max(0, T(v) - T_{\text{safe}}) + \gamma F(v), & \text{otherwise} \end{cases}$$

Accumulated cost:

$$g(v) = g(u) + D(u, v) + R(v)$$

Heuristic:

$$h(n) = \min_{g \in G} \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2}$$

Evaluation:

$$f(n) = g(n) + h(n)$$

## 14 Conclusion

The algorithm outputs an ordered sequence of node IDs representing the safest evacuation path. It minimizes total travel distance and hazard exposure while dynamically adapting to changing fire conditions.