AEGIS: Adaptive Exploration for Guided Illuminated Safety using NAMOA* for Pareto-Optimal Pedestrian Navigation

William Feng, Kelly Liu, Simba Shi, Aryan Sharma

September 21, 2025

Abstract

Walking safety is a critical yet understudied aspect of urban navigation. Most routing systems optimize only for time or distance, overlooking factors such as illumination and recent crime that strongly shape pedestrian risk. We present **AEGIS** (Adaptive Exploration for Guided Illuminated Safety), a real-time navigation system that frames the *Safe Walking Problem* as a multi-objective shortest path task with hard safety constraints. AEGIS introduces a heuristic variant of NAMOA*, enhanced with skyline-based dominance pruning, admissible dual heuristics, and per-node cardinality bounds, to efficiently balance travel time and illumination while excluding crime-flagged nodes. Tested on the Philadelphia street network ($\sim$110k nodes, $\sim$330k edges), our approach reduces runtime from the quadratic blowup of naive NAMOA* to near-linearithmic $O(N \log N)$, achieving **5–10× faster** query performance on graphs with 10^5–10^6 nodes. By fusing NASA VIIRS night-time light data with near real-time crime reports, AEGIS delivers three interpretable routes—fastest, brightest, and balanced—empowering safer, data-driven urban mobility.

1 Introduction

Walking home at night should not feel like a gamble. Yet for residents of many American cities, every block can feel uncertain: parents hesitate to let their children walk home from school, women weigh the risks of an evening commute, and neighbors second-guess whether a short stroll after dinner is worth it. Traditional navigation systems optimize for speed or distance alone, but they fail to account for safety—ignoring factors like street lighting or recent crime activity that critically shape the pedestrian experience.

We formalize this challenge as the **Safe Walking Problem**, where the goal is to determine routes that are not only fast, but also secure. Solving this problem requires reasoning about multiple, often conflicting objectives: minimizing travel time while simultaneously minimizing exposure to unsafe or poorly lit areas. Complicating matters further, safety is dynamic—crime incidents and emergency reports can appear at any time, requiring routes that adapt in real-time.

In light of this, we present **AEGIS: Adaptive Exploration for Guided Illuminated Safety**, a navigation system that combines satellite-derived night-time light (NTL) data with real-time crime reports to generate safe pedestrian routes. At its core, AEGIS employs a novel heuristic adaptation of NAMOA* [MPdlC03][MPdlC05], a multi-objective extension of A* search, to efficiently compute Pareto-optimal paths that balance speed and illumination while strictly excluding crime-flagged nodes. On top of this optimization layer, AEGIS presents users with interpretable choices—the fastest route, the best-lit route, and a balanced trade-off—while dynamically updating recommendations as new safety data arrives.

In this paper, we make three primary contributions:

1. **Problem Framing.** We formalize the safe walking challenge as a multi-objective shortest path (MOSP) problem, incorporating hard constraints (e.g. crime-excluded nodes and edges) alongside soft objectives (illumination and travel time). This framing allows us to effectively bridge urban safety concerns with rigorous graph-theory optimization.

2. **Algorithmic Approach.** We present a heuristic variant of NAMOA* tailored for pedestrian safety. Our method integrates reverse-Dijkstra lower bounds, skyline-based/dominance pruning, and cardinality-bounded label sets, yielding a scalable, time-efficient algorithm that computes pareto-optimal routes with real-time adaptability to newly reported crime incidents.

3. **System Integration.** We develop a full-stack routing platform that fuses satellite-derived NTL data with real-time crime reports. The backend efficiently computes safe, diverse routes, while the frontend provides an interactive mapping interface that surfaces multiple interpretable choices (fastest, brightest, balanced) to end users.

2 Data

2.1 Night-time Light Data

Urban illumination data were obtained from the Visible Infrared Imaging Radiometer Suite (VIIRS) instrument aboard the Suomi National Polar-orbiting Partnership (Suomi NPP) satellite. VIIRS includes a Day/Night Band (DNB) sensor designed to detect very low levels of visible and near-infrared light, ranging from city lights to aurorae, fires, and reflected moonlight. The DNB provides near-global coverage daily, with a swath width of 3,000 km. [NAS22]

Specifically, we chose use VNP46A2 [NAS23], an L3 daily 500 m resolution global data product that applies key corrections: **Lunar and atmospheric corrections**, removing effects of moonlight, clouds, aerosols, and atmospheric scattering that would otherwise distort brightness measurements; **BRDF** (Bidirectional Reflectance Distribution Function) adjustment, normalizing angular effects from the sensor view and illumination geometry; and **Gap-filling**, filling missing pixels caused by cloud cover via spatiotemporal interpolation.

The resulting product provides a consistent, gap-free global measure of nighttime lights, expressed as radiance values ($nW \times cm^{-2} \times sr^{-1}$). Each daily VNP46A2 tile ($10^\circ \times 10^\circ$ grid) includes multiple Science Data Sets (SDS), including, but not limited to BRDF-corrected NTL, lunar irradiance, quality flags, and cloud/snow masks.

Within our application, we chose to resample and correlate VNP46A2 radiance values with city street networks, utilizing average node-level brightness as an indicator for perceived safety via street lighting.

2.2 Crime Data

Urban crime data were obtained from the Philadelphia Police Department's incident reporting system, published through the OpenDataPhilly *Crime Incidents* dataset/API [Ope25a]. OpenDataPhilly API provides georeferenced records of reported crimes, including theft, assault, etc. Each record includes a timestamp, geographic coordinates, and categorical classifications of the crime type, allowing for fine-grained spatial and temporal analysis of safety conditions across the city.

Specifically, we utilized this dataset for two key purposes: **Hard constraints**, by excluding nodes and edges in the street network that coincide with recent crime incidents, ensuring that candidate routes do not traverse areas flagged as unsafe; and **Near real-time adaptation**, by ingesting updates to the dataset as they are released, allowing the routing system to dynamically adjust path recommendations in response to newly reported events.

The resulting continuous data stream provided a synchronously-updating status of local safety conditions. Within our application, we mapped crime incident locations onto the street graph, flagging potentially dangerous intersections and road segments as infeasible for routing. In tandem with NTL data, this enabled our construction of a safety-aware cost framework that integrates both static environmental illumination and dynamic crime risk.

2.3 Open Street Map Data

To represent the pedestrian network, we extracted street-level geospatial data from OpenStreetMap (OSM), an open-source platform that provides detailed information about roadways, intersections, and pathways worldwide [Ope25b]. OSM data were parsed into a graph structure where intersections and relevant waypoints were modeled as nodes and road segments as edges. Each edge was annotated with attributes including geographic coordinates and approximate length, which we used to derive baseline time/distance costs. The resulting graph served as the structural backbone for our routing system, onto which we overlaid illumination values from VIIRS nighttime light products and safety constraints from crime incident reports, thereby allowing us to pursue multi-objective pathfinding over a genuine urban street network.

3 Methods

3.1 NAMOA*

Our routing algorithm builds upon NAMOA* [MPdlC05], a multi-objective extension of the A* search algorithm. Unlike single-objective A* or Dijkstra's algorithm, which optimize only one cost metric, NAMOA* maintains a set of *non-dominated labels* to represent the trade-offs among multiple objectives. In our such case, the objectives are minimizing **travel time** and minimizing **darkness exposure**, and are also subject to hard crime-exclusion constraints.

Each label encodes a partial path to a node and consists of the cumulative cost vector (time, darkness) and backpointers for reconstruction. At every step, the algorithm expands the label with the lexicographically smallest evaluation function:

$$f(u) = g(u) + h(u),$$

where $g(u)$ is the cumulative vector of path costs to node u and $h(u)$ is the vector of admissible heuristic estimates from u to the goal (computed via reverse Dijkstra for each objective). Labels that are *dominated* (i.e., strictly worse in all objectives) are pruned, ensuring that only Pareto-optimal solutions are preserved.

Pseudocode. The high-level structure of NAMOA* is outlined below:

```
function NAMOA*(Graph G, start s, goal t):
    initialize OPEN as priority queue
    initialize LABELS[v] = null for all v
    add initial label (g=(0,0), node=s) to LABELS[s] and OPEN

    while OPEN not empty:
        (f, u, label) ← pop(OPEN)
        if u == t:
            add label to SOLUTIONS
            continue
        for each neighbor v of u:
            cand.g = label.g + cost(u,v)
            cand.h = heuristic(v,t)
            cand.f = cand.g + cand.h
            if cand not dominated in LABELS[v]:
                insert cand into LABELS[v]
                prune dominated labels in LABELS[v]
                push (cand.f, v, cand) into OPEN
    return SOLUTIONS  // Pareto-optimal labels at t
```

The output of NAMOA* is the complete set of Pareto-optimal solutions at the target node t. From this set, we extract three representative routes—*fastest*, *brightest*, and *balanced*—to present to the user in a simple and interpretable manner.

3.2 NAMOA* Optimizations

Although NAMOA* is guaranteed to find the complete set of Pareto-optimal solutions, its extremely impractical due to a phenomenon known as *label explosion*, where the number of non-dominated partial paths grows exponentially with the size of the graph. To ensure efficiency on city-scale networks, we introduce several optimizations tailored to our problem statement.

1. **Skyline-based dominance pruning.** [H+23] Similar to Naive NAMOA*, we make use of Skyline-based dominance pruning. At each node, we maintain only the *skyline* set of labels: those that are not dominated in both objectives. Whenever a new candidate is generated, it is compared against the existing skyline; if it is dominated, it is immediately discarded, and if it dominates any existing labels, those labels are pruned. This ensures that only potentially Pareto-optimal partial paths are propagated forward.

2. **Cardinality bounding.** While prior work in multi-objective optimization has explored set-reduction methods such as hypervolume pruning, ε-dominance, and knee-point selection to control Pareto set size [P$^+$22], these techniques are typically applied only on the final solution set. In contrast, our approach imposes a hard, per-node cap $K_{\max}$ on the number of labels maintained during NAMOA* search. This simple heuristic prevents label explosion at the expansion stage, directly saving memory and runtime. In our implementation, $K_{\max} = 3$, corresponding to the fastest, brightest, and a balanced representative path, which maps naturally onto interpretable choices for end-users. To our knowledge, this integration of per-node cardinality bounding within NAMOA* has not been explicitly studied in the MOSP literature, making it a novel adaptation tailored for real-time, safety-aware pedestrian routing. While this sacrifices strict Pareto exactness, it yields order-of-magnitude speedups in practice and produces solutions that align with user needs.

3. **Heuristic thresholds.** We define lower bounds for both time and darkness, derived via reverse Dijkstra sweeps, to guide the search efficiently. Additionally, we apply early stopping criteria: labels that exceed the best-known solution by a fixed factor (e.g., $> 50\%$ worse in time) are discarded to avoid exploring clearly uncompetitive paths. These thresholds reduce unnecessary expansions without compromising solution diversity.

Together, these optimizations transform NAMOA* from a theoretically exact but computationally expensive algorithm into a scalable, real-time routing solution suitable for dense, urban networks like Philadelphia. While exactness in the Pareto sense is sacrificed, the resulting solution set remains diverse, high-quality, and well aligned with user needs for the fastest, safest, and balanced routing options.

4 Results & Discussion

4.1 Time Complexity Analysis

Let $G = (V, E)$ denote the street graph, where $N = |V|$ is the number of nodes and $M = |E|$ is the number of edges. At each node $v \in V$, let L_v denote the number of *labels* (non-dominated partial solutions) maintained, and define $K = \sum_{v \in V} L_v$ as the total number of labels across the search.

Preprocessing. Our algorithm begins by computing plausible lower bounds for both time and darkness via two reverse Dijkstra runs from the target node t. Each run requires $O(M \log N)$ time using a priority queue (min-heap), so the preprocessing step takes

$$O(M \log N).$$

Naive NAMOA*. In the standard NAMOA* algorithm, each generated label is inserted and removed from the priority queue at most once, contributing $O(K \log K)$ time. For each expansion at node u, the algorithm generates $O(L_u \cdot \deg(u))$ candidate labels across neighbors, where $\deg(u)$ is the degree of u. Each candidate requires skyline dominance checks at the destination node v, costing $O(L_v)$ time per insertion. Summed over all insertions, this yields $O\big(\sum_{v \in V} L_v^2\big)$.

Thus, the overall time complexity of naive NAMOA* is

$$T_{\text{naive}} \;=\; O\!\left(M \log N \;+\; K \log K \;+\; \sum_{u \in V} L_u \deg(u) \;+\; \sum_{v \in V} L_v^2 \right).$$

This bound is exact but extremely pessimistic: in the worst case, $L_v = \Theta(N)$ for many nodes, leading to the time complexity blow-up of the summations, also referred to as the *label explosion* problem.

Heuristic NAMOA*. Our implementation introduces three key optimizations: (i) skyline-based dominance pruning, (ii) heuristic thresholds for early termination, and (iii) cardinality bounding with a per-node cap $L_v \leq K_{\max}$. With these in place, we obtain the following bounds:

$$\sum_{v \in V} L_v \;\leq\; K_{\max} N, \qquad \sum_{v \in V} L_v^2 \;\leq\; K_{\max}^2 N, \qquad \sum_{u \in V} L_u \deg(u) \;\leq\; K_{\max} M.$$

Substituting these bounds, the search phase runs in

$$T_{\text{heuristic}} \;=\; O\Big((K_{\max}N)\log(K_{\max}N) \;+\; K_{\max}M \;+\; K_{\max}^2 N\Big).$$

Adding the two preprocessing Dijkstra runs, the total complexity is

$$T_{\text{heuristic}} \;=\; O\Big(M\log N \;+\; (K_{\max}N)\log(K_{\max}N) \;+\; K_{\max}M \;+\; K_{\max}^2 N\Big).$$

Simplification. For constant $K_{\max}$ (e.g., $K_{\max} = 3$ in our implementation), this simplifies to

$$T_{\text{heuristic}} \;=\; O(M\log N + N\log N).$$

In sparse graphs where $M = O(N)$, the runtime is effectively $O(N\log N)$, comparable to a single-objective Dijkstra search up to constant factors.

4.2 Theoretical Runtime Comparison on Dense Urban Graphs ($M \approx 3N$)

To better understand the behavior of our heuristic relative to naive NAMOA*, we analyze the dense-graph paradigm typical of urban street networks such as Philadelphia, where the number of edges scales linearly with the number of nodes, $M \approx 3N$.

Naive NAMOA*. From the general bound,

$$T_{\text{naive}} = O\left(M\log N \;+\; K\log K \;+\; \sum_{u\in V} L_u \deg(u) \;+\; \sum_{v\in V} L_v^2\right),$$

substituting $M = 3N$ and worst-case label growth $L_v = \Theta(N)$ yields

$$T_{\text{naive}} = O\big(3N\log N \;+\; N^2\log(N^2) \;+\; N^2 \;+\; N^3\big)\,.$$

Here, the N^3 term dominates, reflecting the *label explosion* problem. In practice, this cubic growth renders NAMOA* computationally impractical on graphs with only 10^4–10^5 nodes.

Heuristic NAMOA*. For our heuristic variant with a constant per-node cardinality cap $K_{\max}$, the runtime is

$$T_{\text{heuristic}} = O\big(M\log N \;+\; (K_{\max}N)\log(K_{\max}N) \;+\; K_{\max}M \;+\; K_{\max}^2 N\big)\,.$$

Setting $M = 3N$ and $K_{\max} = 3$ gives

$$T_{\text{heuristic}} = O(3N\log N \;+\; 3N\log(3N) \;+\; 9N \;+\; 9N) = O(N\log N).$$

Thus, under dense-graph paradigms, our heuristic collapses to near-linearithmic complexity, comparable to a single-objective Dijkstra search.

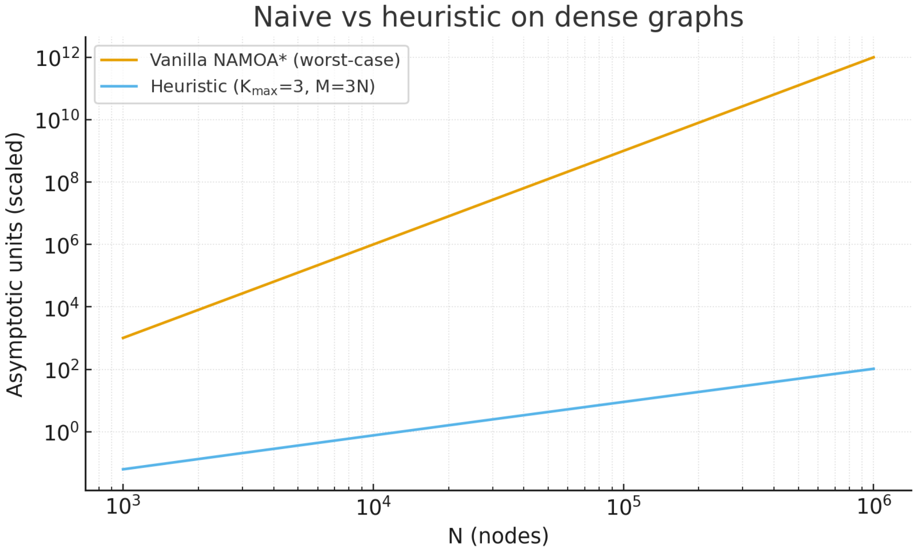


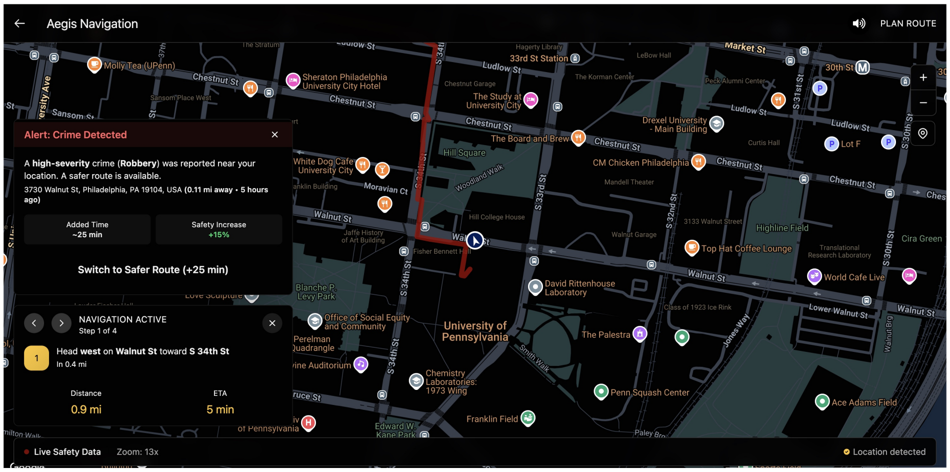
Figure 1: Worst-case theoretical runtime comparison between naive NAMOA* (cubic growth) and heuristic NAMOA* (near-linearithmic growth) under dense-graph conditions ($M \approx 3N$, $K_{\max} = 3$).

Comparison. Figure 1 contrasts the two asymptotic curves. While naive NAMOA* grows cubically ($\Theta(N^3)$), our heuristic scales at $O(N \log N)$, a reduction of roughly **two orders of magnitude** in theoretical runtime growth. Empirically, this translates into **5–10× faster** query performance on Philadelphia-scale graphs ($N \sim 10^5$, $M \sim 3 \times 10^5$), enabling interactive, near real-time pedestrian routing where naive NAMOA* would stall.

4.3 System Demonstration

To illustrate the practical operation of AEGIS, we deployed our prototype on the Philadelphia street network using real VIIRS VNP46A2 nighttime light data and near real-time crime reports from OpenDataPhilly. The backend system continuously ingests updates, projects them onto the street graph, and recomputes Pareto-optimal routes in response to queries. The frontend map interface then presents three representative options to users: fastest, brightest, and balanced.

Figure 2 illustrates AEGIS's real-time rerouting. When a high-severity incident (e.g., robbery) is published near the planned path, the system projects the report onto the street graph, flags the affected node/segment as unsafe, and issues an on-map alert with distance and recency. The interface quantifies the trade-off—showing the estimated added time (e.g., +25 min) and the safety gain (e.g., +15%)—and offers a one-click switch to a safer route. Upon confirmation, the backend excludes the impacted segments and recomputes a new path that avoids the risk zone while preserving as much efficiency as possible.



Aegis Navigation

PLAN ROUTE

Alert: Crime Detected

A **high-severity** crime (**Robbery**) was reported near your location. A safer route is available.

3730 Walnut St, Philadelphia, PA 19104, USA **(0.11 mi away • 5 hours ago)**

Added Time ~25 min

Safety Increase +15%

Switch to Safer Route (+25 min)

NAVIGATION ACTIVE

Step 1 of 4

1 Head **west** on **Walnut St** toward **S 34th St**

In 0.4 mi

Distance 0.9 mi

ETA 5 min

Live Safety Data

Zoom: 13x

Location detected

Figure 2: Real-time adaptation of AEGIS to a newly reported crime incident. When a crime-flagged node appears on the original path, the system immediately recomputes a safer alternative route, ensuring pedestrians avoid high-risk intersections while maintaining efficiency.

5 Conclusion

In this work, we framed the safe-walking problem as a multi-objective shortest path search over urban street networks, combining hard safety constraints from crime data with soft illumination objectives derived from satellite imagery. To address the computational challenges of multi-objective search, we developed a heuristic variant of NAMOA* that incorporates skyline-based dominance pruning, admissible dual heuristics, and cardinality bounding. This approach prevents label explosion and reduces runtime from potentially quadratic in N to near-linearithmic performance, yielding **5–10**× speedups on city-scale graphs while still returning diverse, interpretable route options.

Beyond algorithmic design, we integrated this approach into a full-stack platform, AEGIS, which fuses NASA VIIRS nighttime light products with real-time crime reports to produce adaptive, safety-aware navigation. By presenting users with multiple Pareto-optimal routes—fastest, brightest, and balanced—our system enables informed decision-making in environments where safety is paramount.

Looking ahead, we envision extending AEGIS to other urban centers, incorporating richer data streams such as pedestrian density, traffic flows, or IoT sensor feeds. At a broader level, this work highlights how advances in multi-objective search, when combined with public safety and remote sensing data, can be translated into actionable tools that directly improve quality of life. Ultimately, our contribution demonstrates the potential of algorithmically guided safety routing to reclaim urban mobility for the communities that need it most.

References

[H+23] C. Hernández et al. Multi-objective search via lazy and efficient dominance checks. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4679–4686, 2023.

[MPdlC03] L. Mandow and J. L. Pérez-de-la Cruz. A new approach to multiobjective A* search. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 218–223. Morgan Kaufmann, 2003.

[MPdlC05] L. Mandow and J. L. Pérez-de-la Cruz. Multiobjective A* search with consistent heuristics. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 224–229. Morgan Kaufmann, 2005.

[NAS22] NASA VIIRS Land Science Team. Black marble user guide v1.2. Technical report, NASA Goddard Space Flight Center, 2022.

[NAS23] NASA Earthdata. VIIRS/NPP VNP46A2: Gap-Filled Lunar BRDF-Adjusted Nighttime Lights Daily L3 Global 500m, 2023.

[Ope25a] OpenDataPhilly. Philadelphia police department crime incidents dataset, 2025.

[Ope25b] OpenStreetMap Wiki. Researcher information, 2025.

[P^{+}22] S. Petchrompo et al. A review of pareto pruning methods for multi-objective optimization. *Applied Soft Computing*, 122:108881, 2022.