

A STUDY OF THE PERFORMANCE OF ANY-ANGLE THETA* ALGORITHMS ON WEIGHTED GRID MAPS FOR ROUTE PLANNING

Kis Y. – Post-graduate student of the Department of Discrete Analysis and Intelligent Systems, Ivan Franko National University of Lviv, Ukraine. ROR: <https://ror.org/01s7y5e82>. ORCID: <https://orcid.org/0009-0009-7816-237X>.

Shcherbyna Y. M. – PhD, Professor of the Department of Discrete Analysis and Intelligent Systems, Ivan Franko National University of Lviv, Ukraine. ROR: <https://ror.org/01s7y5e82>. ORCID: <https://orcid.org/0000-0002-4942-2787>.

Kunanets N. E. – Dr. Sc., Professor of the Department of Information Systems and Networks, Lviv Polytechnic National University, Lviv, Ukraine. ROR: <https://ror.org/0542q3127>. ORCID: <https://orcid.org/0000-0003-3007-2462>.

Yarymovych Y. A. – Post-graduate student of the Department of Information Systems and Networks, Lviv Polytechnic National University, Lviv, Ukraine. ROR: <https://ror.org/0542q3127>. ORCID: <https://orcid.org/0009-0006-1391-3214>.

ABSTRACT

Context. The article addresses the study of free-direction pathfinding algorithms, in particular the Theta* algorithm, and evaluates their performance on weighted grid maps in order to determine optimal routes for delivering goods to a firearms store. This research is carried out in the broader context of developing an information system for tracking and managing arms sales and logistics under complex conditions. One of the main motivations is that any-angle methods can produce more realistic and natural-looking paths compared to the classical A* algorithm.

Objective. The purpose of the study is to analyze the performance of three Theta*-based algorithms – Basic Theta*, Lazy Theta*, and Strict Theta* – on both uniform and weighted square grids, with special emphasis on execution time and path cost metrics. The work aims to generalize the applicability of these algorithms to weighted environments and to propose improvements suitable for real-world route planning scenarios.

Method. The principles of A*, the three Theta* variants, and path post-processing smoothing techniques are presented. The research describes the transition from unweighted uniform square grids to weighted grids and highlights the complexity of calculating accurate path costs when applying any-angle approaches. Visual demonstrations of algorithmic behavior were implemented using the Unity game engine. Performance metrics were measured separately for uniform and weighted grids to ensure comparative analysis.

Results. The results include comparative evaluations of Basic Theta*, Lazy Theta*, Strict Theta*, and classical A* algorithms. The analysis identifies conditions under which each algorithm performs effectively, as well as factors that limit their applicability in weighted environments. It is shown that path length and path cost may differ substantially in weighted grids, leading to new considerations for cost-based optimization. Based on the experiments, a generalization of the Basic Theta* algorithm is proposed to enhance its suitability for weighted square grids, and a potential extension of the Strict Theta* algorithm to this context is outlined.

Conclusions. The findings demonstrate that while any-angle algorithms provide smoother and more realistic routes, their effectiveness in weighted environments depends on careful adaptation of cost functions. The research highlights their value not only for simulating complex virtual environments and agent behaviors in games and robotics but also for practical applications in logistics, particularly in the development of an information system for tracking and managing firearms sales. The proposed algorithmic adaptations may contribute to improving delivery planning and supply chain efficiency, including the modeling of weapons delivery routes under wartime conditions.

KEYWORDS: pathfinding, path planning, square grid, any-angle algorithm, path cost, weighted grid, Theta*.

ABBREVIATIONS

AI – Artificial Intelligence;
A* – A-star algorithm;
API – Application Programming Interface;
GIS – Geographic Information System;
SCM – Supply Chain Management;
WGM – Weighted Grid Map;
UML – Unified Modeling Language;
ms – milliseconds.

NOMENCLATURE

s is a node corresponding to the point of departure;
 S is a set of grid cells;
 N is a number of nodes;
 W_r is a road weight coefficients;
 $R_{i,j}$ is a risk factor for each cell;
 $T_{i,j}$ is a traffic intensity;

$T(P)$ is a travel time;
 E is an efficiency and safety indicator;
 P is a logistics cost parameters;
 g is a node corresponding to the firearms store (destination);
 D is an optimal path;
 L is a path length;
 $L(s, g)$ is a set of all feasible routes between s and g that do not pass through forbidden or hazardous cells;
 V is a set of grid cells (nodes);
 $v_{i,j}$ is a set of grid cells (nodes) indexed by coordinates (i,j) ;
 $d_{i,j}$ is a road quality coefficient (road surface condition);
 $r_{i,j}$ is a risk factor (e.g., presence of checkpoints, crime level);

$t_{i,j}$ is an average traffic;
 $p_{i,j}$ is a logistics cost (fuel, duties, etc.);
 α is a weight for road surface condition;
 β is a weight for risk level;
 γ is a weight for traffic density;
 δ is a weight for surveillance or control measures;
 r_{\max} is a maximum allowable risk of a cell;
 Ω is a set of cells designated as risk zones or restricted areas;
 λ is a congestion impact coefficient;
 v_{avg} is a average vehicle speed;
 P is a set of all valid routes between s and g that do not pass through forbidden or dangerous cells;
 V is a set of nodes (grid cells) indexed by coordinates (i,j) ;
 v_k is a k -th node (vertex) in the grid representing a specific position or cell;
 v_{k+1} is a $k+1$ -th node in the path or in the visibility sequence;
 $v_{i,j}$ is a node is assigned a weight (traversal cost);
 $C(P)$ is a weighted length (cost) of a path P ;
 $C(v_k, v_{k+1})$ is a transition cost between nodes v_k and v_{k+1} ;
 $\bar{w}_{k,k+1}$ is an averaged weight of the cells crossed by the segment $[v_k, v_{k+1}]$;
 $w(v_k)$ is a weight of node v_k .

INTRODUCTION

Nowadays, pathfinding algorithms are widely used across numerous fields related to computer science, including robotics [1], logistics, navigation systems [2], routing protocols [3, 4], and video games [5]. It is evident that different application domains require different approaches to solving the problem of pathfinding – or, as some sources refer to it, path planning. For example, in logistics, pathfinding may involve traversing graph vertices representing warehouses between which goods are transported. In contrast, in video games, the virtual environment is typically discretized and represented as a square grid or navigation mesh. Another potential variation arises in scenarios involving incomplete information about the graph structure, such as in routing protocols, where routers (treated as graph nodes) operate with limited knowledge and attempt to forward data toward a destination – often prioritizing reachability over optimality.

Given the broad spectrum of use cases, pathfinding has been extensively studied. However, this does not imply that there is no room for improvement of existing algorithms. This study focuses on the exploration of any-angle pathfinding algorithms, particularly Theta*, when

© Kis Y. O., Shcherbyna Y. M., Kusanets N. E., Yarymovych Y. A., 2026
DOI 10.15588/1607-3274-2026-1-12

applied to square grid environments. These algorithms are capable of producing shorter paths in Euclidean space compared to the classical A* algorithm [6], which is constrained to move between adjacent grid nodes.

The primary emphasis of this work lies in investigating the performance of Theta*-based algorithms on weighted square grids, where their any-angle nature provides a substantial advantage over A*. Moreover, paths generated by any-angle methods often appear visually more plausible, as agents are not restricted to movement along fixed angular directions but can adjust their trajectory according to the geometry and structure of the environment.

The relevance of this topic stems from the growing demand for realistic simulations of complex virtual environments [7], as well as the need for intelligent decision-making in agent navigation. One such example might be simulating a video game character crossing a river.

The relevance of this research also lies in adapting the results of studying Theta* algorithms for planning supply routes to firearms stores, particularly under complex logistical conditions. The results of the study are integrated into an information system for managing firearms sales, including an AI module for selecting and re-selecting routes as the situation changes, simulating risk scenarios, and visualizing routes on a logistics dashboard. In addition, this research is relevant for security services, private security companies, military logistics, specialty goods stores, and the public sector, where high accuracy, flexibility, and realism of planned routes are required. All these factors can be converted into a quadratic grid and expressed as the weight of a specific cell, which indicates either the time required to traverse that cell or the probability of failure when passing through it.

A distinguishing feature of this paper is its comparative analysis of Theta* algorithms against the standard A* algorithm and a variant that incorporates path smoothing. The smoothing process partially mimics the any-angle nature of Theta*, but it has limitations – primarily because it only smooths paths generated by A*, which can negatively affect the cost of the resulting path, especially when operating on a weighted grid.

The object of research is the processes of searching for optimal routes in discrete spatial models (grids), taking into account the weighted characteristics of the environment.

The subject of research of the study is any-angle pathfinding algorithms, in particular the variants of the Theta* algorithm (Basic Theta*, Lazy Theta*, Strict Theta*) and their application to weighted square grids for route planning tasks.

The obtained results are distinguished by their **scientific novelty**, which lies in the proposal to use the algorithms lies in the fact that, for the first time, the specific features of applying Any-Angle Theta* algorithms to weighted square grids have been analyzed, with consideration of the difference between path length and path cost, which is critical for optimization tasks on maps with weighted coefficients. The methodology for quantitatively

evaluating the efficiency of Theta* algorithms has been improved by comprehensively accounting for two metrics-execution time and path cost – in environments with different types of grids (uniform and weighted). Furthermore, the generalization of the Basic Theta* algorithm for application in weighted environments has been further developed, and directions have been identified for extending the Strict Theta* algorithm to weighted maps, opening new opportunities for practical use in logistics systems and complex route management.

1 PROBLEM STATEMENT

In the context of secure logistics, the task of planning safe and efficient delivery routes for high-risk goods, particularly firearms, is becoming increasingly complex due to unstable infrastructure, dynamic security conditions, traffic restrictions, and the need for real-time adaptation. Traditional grid-based algorithms such as A* do not always ensure sufficient accuracy, flexibility, or adaptability, since they rely on fixed movement directions and do not adequately reflect weighted environmental constraints such as risks, terrain conditions, or traffic.

Input Variables: $S, N, W_r, R_{i,j}, T_{i,j}, P, \alpha, \beta, \gamma, \delta, s, g$.

Desired Outcomes (Output Variables): $D, L, C(P), T(P), E$.

Dependencies: D depends on spatial constraints S and N , L depends on road and obstacle weights W_r .

$C(P)$ is a function of road quality, risk factor, traffic, and logistics costs:

$$C(P) = \sum_{k=0}^{n-1} \frac{\text{length}(v_k, v_{k+1})}{V_{avg}} + \lambda \cdot t_{i,j}. \quad (1)$$

$T(P)$ depends on average traffic and speed constraints:

$$C(P) = \sum_{k=0}^{n-1} \frac{\text{length}(v_k, v_{k+1})}{V_{avg}} + \lambda \cdot t_{i,j}. \quad (2)$$

$\text{length}(v_k, v_{k+1})$ is the Euclidean distance between nodes v_k and v_{k+1} . E depends on minimization of cost and risk under time constraints:

$$T(P) \leq T_{\max}, r_{i,j} \leq r_{\max}. \quad (3)$$

The problem reduces to finding a path P^* such that:

$$P^* = \arg \min_{P \in L(s,g)} C(P).$$

Mathematical modeling of these parameters enables the construction of optimal, safe, and realistic routes on weighted grid maps. This justifies the need to investigate

the performance of Any-Angle Theta* algorithms (Basic, Lazy, Strict) as adaptive tools for secure logistics and route planning in high-risk environments.

2 REVIEW OF THE LITERATURE

Previous studies have explored the application of Theta* algorithms on weighted square grids. In the original paper proposing the Basic Theta* algorithm [8], a generalized version was presented and compared against A* and Field D*. In that study, the cost of the Theta* path was calculated along the line connecting two vertices, considering the cumulative weights of all intersected cells. The weighted Theta* paths were, on average, 3% shorter than those produced by A*, but only in cases where the grid contained large contiguous areas of cells with baseline weight values.

An alternative approach was proposed in which the entire map was assigned a weight factor [9], and this factor was incorporated into the heuristic distance function to the goal. Through such pre-analysis of map complexity, algorithms in the Theta* family achieved higher efficiency by reducing execution time during pathfinding.

There also exists a generalization of the Lazy Theta* algorithm adapted for pathfinding in weighted 3D environments [10]. The cost computation approach in that work was similar to that of [8], but the primary focus was on three-dimensional space.

Furthermore, a dynamically hybrid algorithm, Non-uniform-Theta*, was developed for autonomous ground vehicle navigation in environments containing both static and dynamic obstacles [11]. This algorithm enables real-world maneuvering by integrating both path planning and agent-level motion control in dynamic settings.

To date, no dedicated study has been found that evaluates the performance of the Strict Theta* algorithm on weighted square grids. In general, the most promising approach for computing accurate and optimal paths appears to be the line-based cost evaluation method, which considers all intersected cell weights along the line-of-sight between vertices. Given that the primary advantage of Theta* algorithms over A* lies in their ability to produce lower-cost paths, this cost-evaluation method was adopted in the present study.

In contrast to prior works, this paper evaluates the performance of all three Theta* algorithms (Basic, Lazy, and Strict) on a weighted square grid, and compares them against both the standard A* algorithm and A* with path smoothing.

3 MATERIALS AND METHODS

This study adopts a simulation-based experimental methodology, combining concepts from computational geometry, optimization, and artificial intelligence to analyze the performance of directional pathfinding algorithms – specifically Theta* and its modifications – on weighted grid maps. The research aims to evaluate route quality, realism, and risk-aware cost optimization in weapon delivery logistics under uncertain and dynamic conditions.

The simulation environment is built in Unity to reflect realistic terrain conditions, traffic, and risk zones. This allows reproducible and controlled experiments for testing routing algorithms under varying logistical constraints.

The routing environment is modeled as a Weighted Grid Map (WGM), where the delivery region (e.g., Lviv Oblast or border zones) is discretized into grid cells (e.g., 50×50 meters). Each cell carries a weight $w_{j,i}$, which is computed using a composite function:

$$w_{j,i} = \alpha \cdot d_{j,i} + \beta \cdot r_{j,i} + \gamma t_{j,i} + \delta p_{i,j}, \quad (4)$$

represent road conditions, security risk, traffic intensity, and surveillance density respectively, while $\alpha, \beta, \gamma, \delta$ are tunable coefficients based on scenario-specific priorities (e.g., minimizing risk in high-value cargo delivery). $\alpha, \beta, \gamma, \delta \in R^+$ are the weighting coefficients reflecting priorities.

The experimental framework evaluates three variants of the Theta* family:

- Basic Theta*: prioritizes shortest, realistic paths with arbitrary angles;
- Lazy Theta*: optimistically assumes line-of-sight and defers visibility checks;
- Strict Theta*: enforces obstacle-hugging behavior for higher realism in constrained urban settings.

Each algorithm is benchmarked across multiple scenarios, including uniform and weighted maps, short-range and long-haul delivery simulations. Over 2500 runs were conducted, with randomized source-target locations and consistent pseudorandom seeds for reproducibility.

The results of each algorithm were assessed using total route cost (sum of weights along the path), computation time (ms), path realism (visual coherence and feasibility of navigation). For dynamic adaptation, an API connection to real-time traffic data (e.g., simulated Google Traffic) and threat updates was considered in the weighted grid recalculation.

Route visualization and metric logging were performed via Unity and Python, while data analysis and comparisons used Pandas, Matplotlib, and NumPy libraries.

Inspired by techniques in AI-driven navigation and perceptual learning, the weighted environment simulates realistic logistical bottlenecks. While this research does not employ GANs or image-based synthesis, the methodology follows a multi-layer abstraction similar to preprocessing in computer vision. Grid generation, threat encoding, and cost-map construction serve as “preprocessing layers”, and path optimization mimics an inference step, outputting cost-effective, secure routes. This layered modeling allows flexible integration into AI-based decision support systems, e.g., those incorporating reinforcement learning or probabilistic planning for supply chain operations.

In the current conditions, ensuring the logistics of high-risk goods supply, particularly weapons, is becoming

an increasingly complex task due to unstable infrastructure, dynamic changes in the security environment, traffic restrictions, and the need for real-time adaptation to external factors. Traditional pathfinding algorithms, such as A*, are not always capable of providing sufficient accuracy, flexibility, and efficiency in route planning under such conditions, as they operate in a discrete grid-based environment with fixed movement directions and limited adaptability to changing weight characteristics of the route. This leads to inefficient routes that fail to account for risks, road conditions, dynamic constraints, and other critical parameters.

The problem lies in the lack of sufficiently universal, flexible, and adaptive tools capable of modeling a complex logistics environment using a weighted spatial model and finding routes that not only minimize distance but also consider critical parameters of the delivery environment.

In this context, it becomes necessary to investigate the efficiency and adaptation of free-direction pathfinding algorithms such as Theta*, particularly their operation on weighted grid maps, as a means of building realistic, safe, and optimal routes in a high-risk environment for delivering goods to firearms stores.

The task is to determine a safe and efficient delivery route for goods to a firearms store, taking into account spatial constraints, potential risk zones, and the need to minimize time and transportation cost. To model the environment, a weighted grid is used, where each cell has its own weight that reflects the complexity or danger of traversing it (for example, the presence of obstacles, surveillance cameras, checkpoints, or other risk factors).

It is necessary to find a route from a starting point (warehouse or logistics center) to a firearms store so that:

- the total cost of the route is minimal;
- the route remains within permitted areas and maximally avoids high-risk zones;
- the trajectory is as close as possible to the optimal (shortest) path, taking into account the possibility of movement not only horizontally and vertically but also at arbitrary angles.

To solve this problem, a free-direction Theta* algorithm is proposed. Unlike classical grid-based pathfinding algorithms (such as A* and D*), Theta* allows the construction of routes that are not limited to grid axes and reduces unnecessary turns, producing paths closer to the straight-line optimum.

The result is a route that simultaneously meets the criteria of safety (avoiding dangerous cells) and efficiency (minimizing both path length and total weighted cost).

Let the delivery environment be discretized as a weighted grid

$$G = (V, E). \quad (5)$$

$V = v_{i,j}$ is the set of grid cells (nodes) indexed by coordinates (i,j) ; $E \subseteq V \times V$ is the set of edges connecting adjacent nodes (in Theta*, connections in arbitrary directions are allowed if the conditions of direct visibility

are satisfied); $C:V \times V \rightarrow R+$ is the cost function for moving between nodes, which takes into account the properties of the cells through which the segment passes. Each node $v_{i,j}$ is assigned a weight (traversal cost) $w(v_{i,j}) \geq 0$, determined by the conditions of the terrain and safety. Start and goal nodes $s \in V, g \in V$. For each grid cell $w_{i,j}$ we define its weight as (4).

We define the cost function of a route. Let the path P be a sequence of nodes $P = (v_0, v_1, \dots, v_k)$, $v_0 = s_{start}$ is a start node, with the goal $v_n = v_{goal}$. The weighted length of the path is defined as:

$$C(P) = \sum_{k=0}^{k-1} (\|v_{k+1} - v_k\| \cdot \frac{w(v_k) + w(v_{k+1})}{2}) \quad (6)$$

$\|v_{k+1} - v_k\|$ is the Euclidean distance between the centers of the grid cells (for Theta*, both diagonal and arbitrary directions are taken into account). v_k, v_{k+1} is a any pair of adjacent nodes between which there is direct visibility in the grid.

Objective (optimality criterion) it is required to find a route.

And minimizes the total cost:

$$\frac{\min}{P} \sum_{k=0}^{k-1} C(v_k, v_{k+1}), \quad (7)$$

where the transition cost is

$$C(v_k, v_{k+1}) = length(v_k, v_{k+1}) \cdot \bar{w}_{k,k+1}. \quad (8)$$

It is required to find a path P^* that satisfies

$$P^* = arg \frac{\min}{P \in L(s, g)} C(P), \quad (9)$$

$L(s, g)$ for which $w(v) = \infty$. A path is considered feasible if:

$$\forall_k, visibility(v_k, v_{k+1}) = true \text{ i } r_{i,j} \leq r_{max}.$$

For any pair of consecutive nodes (v_k, v_{k+1}) , a clear line-of-sight without intersecting obstacles is required. In this case $v_k \notin \Omega, \forall v_k \in P$.

To define the delivery time constraint, let $T(P)$ denote the total travel time along the route (a function of the path length and traffic):

$$T(P) = \sum_{k=0}^{k-1} (\frac{length(v_k, v_{k+1})}{v_{avg}} + \lambda \cdot t_{i,j}). \quad (10)$$

It is required to (3). The generalized formalized problem can be expressed as follows:

$$\frac{\min}{P} \sum_{k=0}^{n-1} C(v_k, v_{k+1}) \quad (11)$$

subject to $T(P) \leq T_{max}, r_{i,j} \leq r_{max},$
 $visibility(v_k, v_{k+1}) = true.$

Thus, the problem reduces to finding a path of minimal cost, taking into account the cell weights and safety constraints, using the Theta* algorithm, which allows constructing routes with freely chosen directions.

The pathfinding problem assumes the existence of a certain graph structure over which the search process is performed. In this case, the graph is represented by a square grid, where each cell has eight neighbors (including diagonals). Grid cells can exist in one of two states: unblocked or blocked, visualized as white and gray cells, respectively, in Figure 1. Unblocked cells may also carry a numerical weight value, indicating an increased traversal cost when passing through the respective cell. A square grid in which cells contain such weight values is referred to as a weighted square grid.

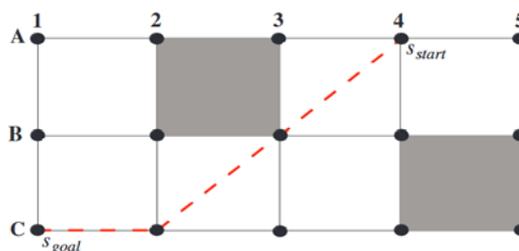


Figure 1 – An illustration of an unweighted square grid; the computed path is shown as a dashed line

In the pathfinding process, movement occurs through the corners of grid cells, which serve as the graph's vertices. The set of all vertices is denoted as S , and an individual vertex is denoted as s , where $s \in S$. The goal of the pathfinding problem is to determine an unblocked path from the starting vertex $start$ to the target vertex $send$. An example of such a path is shown as a dashed line in Figure 1. A path is considered unblocked if every vertex along the path has line-of-sight to the next vertex. Line-of-sight is defined as the condition in which a straight line connecting any two vertices does not pass through the interior of any blocked cell or between two blocked cells along a shared edge.

Key criteria for evaluating pathfinding algorithms include path optimality and computational efficiency. An optimal path is the shortest possible path, while efficiency refers to the minimization of computation time during the search. Due to their any-angle nature, Theta* algorithms are capable of finding Euclidean-optimal paths, often yielding shorter paths than those produced by the classical A* algorithm, which is limited to grid-adjacent move-

ments. The same advantage is observed when operating on weighted square grids.

Since Theta* algorithms are derived from the A* algorithm [6], the following section outlines the fundamental principles of A*, based on which the key modifications introduced by any-angle algorithms are identified.

A* is conceptually straightforward: starting from the initial vertex s_{start} , it aims to find a path to the target vertex s_{goal} that minimizes the total cost. The algorithm operates by constructing a search tree, beginning at s_{start} and incrementally expanding paths one vertex at a time until the goal is reached.

At each iteration of the main loop, the A* algorithm must determine which vertex to expand next. The vertex with the lowest numerical value of $f(s)$ is selected. This value represents an estimated cost of the shortest possible path from the s_{start} vertex s_{start} , passing through s , to the goal vertex s_{goal} . The function $f(s)$ is defined as the sum of two components: the actual cost $g(s)$, which is the known cost from s_{start} to s , and the heuristic estimate $h(s)$, which approximates the remaining distance from s to s_{goal} . The choice of heuristic depends on the specific problem domain. In this study, the heuristic function is defined as the Euclidean distance between s and s_{goal} .

The value of f is updated at each iteration during the expansion of vertex s . All neighboring vertices $[s_1, s_2, \dots, s_8]$ compute their respective $f(s_n)$ values. If the newly computed value is lower than the previously stored one (at the beginning of the algorithm, the value of f of all vertices is $+\infty$), update this value to the newly calculated one and update the parent vertex with vertex s .

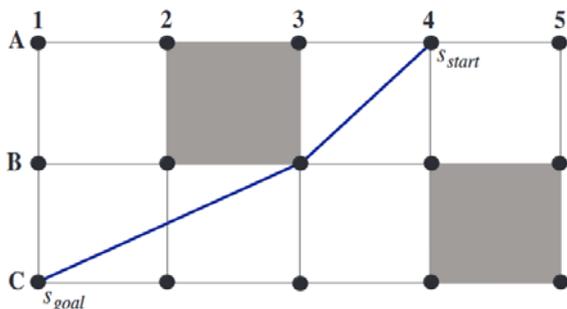


Figure 2 – The true shortest path is shown as a solid line from s_{start} to s_{goal}

This process defines the operation of the optimal path-finding algorithm A* [5]. To obtain even shorter paths, the Theta* algorithm [7] was introduced. As mentioned earlier, Theta* searches for the shortest paths in Euclidean 2D space and is not restricted to movement in just eight grid directions. This flexibility allows Theta* to approximate true shortest paths on a square grid more accurately. The any-angle Theta* algorithms discussed in this work do not guarantee perfectly optimal Euclidean paths, but they typically produce paths that are shorter than those generated by A*. An example of an optimal A* path is shown in Figure 1, while Figure 2 illustrates a true shortest path found using an any-angle Theta* algorithm.

The key distinction of Theta* algorithms lies in the fact that the parent of any vertex along the path may be any other vertex on the grid that is reachable via an unobstructed line of sight. In contrast, in the A* algorithm, a vertex can only have one of its immediate neighbors as a parent.

In practice, the only functional difference in the Basic Theta* algorithm is that, during the expansion of a vertex s and subsequent evaluation of the f -value of a neighboring vertex s' , two possible paths are considered. These paths are illustrated in Figure 3.

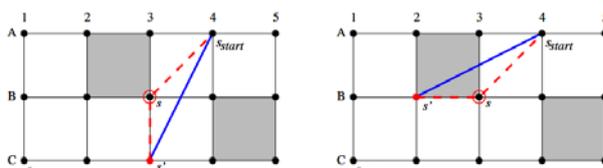


Figure 3 – Path 1 is shown as a dashed line. Path 2 is shown as a solid line

Path 1 corresponds to the standard path evaluated by A*, where the g -value is equal to the sum of the distance from $parent(s)$ to s , and from s to s' . In contrast, Path 2 checks whether a direct connection exists between $parent(s)$ and s' , bypassing vertex s .

If a straight, unobstructed line of sight exists between $parent(s)$ and s' , this path is selected instead. According to the triangle inequality, Path 2 can never be longer than Path 1, since any side of a triangle is not longer than the sum of the lengths of the other two sides.

The smoothed A* algorithm adopts a similar principle, but on a smaller scale. While Basic Theta* evaluates Path 2 during the expansion phase, the smoothing process applies the same idea after the A* algorithm has already computed a complete path.

Given a sequence of vertices from the resulting path, the smoothing process iterates through them, checking whether a later vertex in the sequence has a direct line of sight to the currently processed vertex. If such a vertex is found, the intermediate vertices can be bypassed.

This post-processing approach is efficient, as it operates on a fixed path. However, it is limited to the set of vertices discovered by A*, and cannot search for truly optimal paths. This behavior is illustrated in Figure 4.

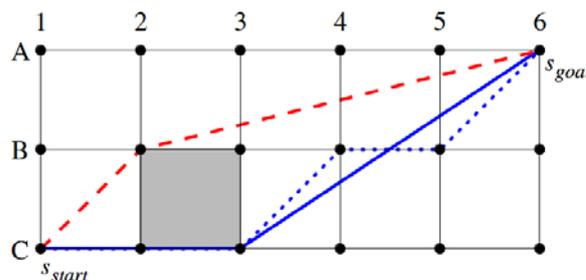


Figure 4 – The true shortest path is depicted by a dashed line, the path found by A* is represented by a dotted line, and the smoothed path is shown as a solid line

The Lazy Theta* algorithm [12] optimistically assumes that Path 2 is always available. As a result, it significantly reduces the number of computations performed during the vertex expansion phase, since it does not initially verify whether the assumed path truly provides an unobstructed line of sight. Instead, in the next iteration, it re-evaluates whether the assumption was valid and, if not, reverts to Path 1 without changing the current vertex. This strategy is effective in reducing runtime but heavily relies on the triangle inequality, which does not always hold when a square grid is transformed into a weighted grid.

The Strict Theta* algorithm [13] further introduces the concept of a tightly surrounding path. A path is considered tightly surrounding if every change in direction closely wraps around a specific obstacle. On uniform square grids, the optimal path is typically one that is tightly surrounding. However, not all tightly surrounding paths are necessarily optimal. Therefore, while Strict Theta* does not guarantee optimal pathfinding, it increases the likelihood of identifying near-optimal paths, as it leverages this additional spatial constraint – something that Basic Theta* does not account for.

Strict Theta* is implemented largely in the same manner as Basic Theta*, with the exception of an additional check to determine whether the path tightly surrounds an obstacle – this check is performed in constant time. If the path to the current vertex is not tightly surrounding, an additional penalty distance is added to the g-value after the vertex is expanded. The penalty distance is defined as $\sqrt{2} - 1$, which approximately equals 0.42.

The check to determine whether the path [parent(s), s, s'] is tightly surrounding requires examining a single grid cell. Among the four cells adjacent to vertex s, only the one that lies within the interior of the angle $\angle_{parent(s), s, s'}$ less than 180 degrees is evaluated. This segment is considered tightly surrounding if and only if the corresponding cell is blocked. This behavior is illustrated in Figure 5.

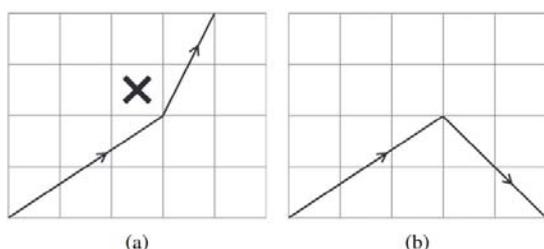


Figure 5 – (a) Since the grid cell within the angle less than 180 degrees is blocked, the path is considered tightly surrounding; (b) the path is not tightly surrounding

The following section describes the methodologies used for computing the path cost on a weighted square grid. Figure 6 provides a visual representation that serves as a reference for verifying the described path cost computation techniques.

Since A* navigates between neighboring vertices, there are three possible cases for assigning cell weights during path cost computation. In these examples, cells are

denoted by the name of their top-left vertex. For instance, the cell enclosed by vertices A1, A2, B1, and B2 is referred to as A1, and its weight is denoted as weight(A1). The distance between two vertices is represented as $c(s_1, s_2)$.

In the first case, the path proceeds from the starting vertex ($s_{start} = B2$) to the goal vertex ($s_{goal} = A1$) within the same cell, i.e., diagonally. In this scenario, the path length equals $c(B2, A1)$, while the path cost is computed as the product of the path length and the weight of the cell it passes through. Thus, the path cost is calculated according to Equation (1)

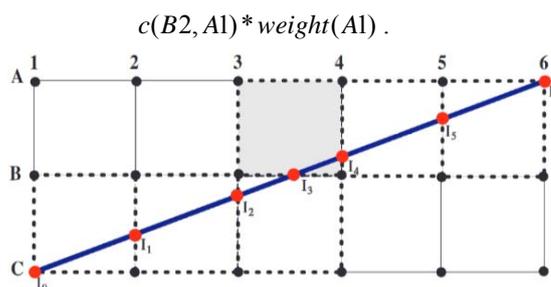


Figure 6 – The line represents the path found by the any-angle algorithm

In the second case, the movement occurs from the starting vertex ($s_{start} = B2$) to the goal vertex ($s_{goal} = A2$) along the edge shared by two unblocked cells, i.e., a horizontal or vertical move. Similarly, the path length is calculated as $c(B2, A2)$; however, in this case, it is multiplied by the average weight of the two adjacent cells. Therefore, the path cost in this scenario is computed according to Equation (2)

$$\frac{c(B2, A2) \cdot (\text{weight}(A1) + \text{weight}(A2))}{2}$$

In the third case, the movement proceeds from the starting vertex ($s_{start} = B1$) to the goal vertex ($s_{goal} = A1$) along the edge between one unblocked cell and one blocked cell, which is again a horizontal or vertical move. In this scenario, cell A0 does not exist; if it did, it would be considered blocked. The path cost is calculated as the path length multiplied by the weight of the unblocked cell, according to Equation (3)

$$c(B1, A1) \cdot \text{weight}(A1)$$

To some extent, the situations described are appropriate to apply when calculating the path cost during the operation of Theta* algorithms. At the same time, let us analyze another unique case.

In the fourth case, the movement proceeds from the starting vertex ($s_{start} = C1$) to the goal vertex ($s_{goal} = A6$) across multiple grid cells. The straight line from vertex C1 to vertex A6 is divided into several segments at the points where it intersects the cell boundaries (l_0, l_1, \dots, l_6). The length of each segment is denoted as

$c(l_n, l_{n+1})$, and the cost of each segment is calculated as the segment length multiplied by the weight of the cell through which the segment passes. The total path cost from C1 to A6 is the sum of the segment costs that make up the path. As an example, the cost of the first segment is computed as follows:

$$c(l_0, l_1) \cdot \text{weight}(B1).$$

To compute the lengths of specific path segments, the Fast Voxel Traversal Algorithm [14] is employed. Although originally designed for use in three-dimensional space, this algorithm performs equally effectively in two-dimensional environments.

The following section discusses the modifications related to the operation of Theta* algorithms on weighted grid maps. While working with uniform grid maps, the triangle inequality property held true, since the cost of Path 2 (direct diagonal) was never greater than that of Path 1 (via intermediate vertex), assuming cells had equal weights. Consequently, the implementation of the Basic Theta* algorithm was designed to prefer Path 2 without verifying whether Path 1 might be shorter.

Since the work is now being conducted on a weighted grid map, the triangle inequality no longer holds in many cases. As a result, a more deliberate decision must be made between Path 1 and Path 2. This implies that an explicit cost comparison between both paths is required, and the one that truly yields the shorter total cost should be selected.

The Lazy Theta* algorithm optimistically selects Path 2, which immediately suggests that its performance will degrade significantly, as it was originally designed under the assumption that the triangle inequality always holds.

Similarly, the Strict Theta* algorithm introduced the concept of tightly bounding paths, assuming that changes in the optimal path direction occur only at the edges of obstacles. This assumption becomes invalid with the introduction of weighted grid maps. Nevertheless, the algo-

rithm was also extended to compare the costs of Path 1 and Path 2 before making a final decision.

A weighted grid map was generated with a width of 125 cells and a height of 50 cells. The size of an individual cell was set to 0.4 meters. Every second cell was randomly assigned a weight value. The weight value was randomly selected as a fractional number from 0 to 3 inclusive, and this value was added to the base weight value of -1 . Thus, the weight values will vary from 1 to 4 in every second cell.

The evaluation was conducted by executing 2500 algorithm runs. The start and goal positions were selected randomly. A fixed seed value was set for the pseudorandom number generator to ensure that the paths being searched were identical across all algorithm executions. The final results represent the average path cost and execution time of the pathfinding process. The distance is measured in meters, and the time in milliseconds.

Initially, the execution time and path cost were measured for the algorithms on an unweighted grid map. The results are presented in Table 1

As can be seen from the obtained results, all any-angle algorithms require more time for pathfinding while providing no significant advantages in the resulting path length compared to the smoothed A* algorithm. The smoothed A* executes only 1.13% longer than the original A*, while the cost of the path it finds is only 0.32% higher than that of Strict Theta*, which produces the shortest paths. However, the time required by Strict Theta* is 21.26% greater than that of the original A*.

Although the execution times of all algorithms remain low – within the range of milliseconds – these results highlight the efficiency of the path smoothing operation on a uniform, unweighted grid.

The results of the algorithms on the weighted grid are presented in Table 2.

Table 1 – Execution time and costs [15]

	A*	A* smoothing	Basic Theta*	Lazy Theta*	Strict Theta*
Execution time (ms)	0.7499	0.7584	0.8459	0.8418	0.9093
Execution time (%)	100%	101.13%	112.8%	112.25%	121.26%
Distance (m)	37.031	35.856	35.757	35.764	35.74025
Distance (%)	100%	96.83%	96.56%	96.58%	96.51%

Table 2 – Results of the algorithms [15]

	A*	A* smoothing	Basic Theta*	Lazy Theta*	Strict Theta*
Execution time (ms)	1.3536	1.3756	2.7028	2.996	2.2826
Execution time (%)	100%	101.63%	199.67%	221.34%	168.63%
Distance (m)	50.1497	60.7711	45.92569	60.25556	46.70809
Distance (%)	100%	121.18%	91.58%	120.15%	93.14%

The execution time of all algorithms increased significantly due to the additional computational overhead required for pathfinding on a weighted grid. Furthermore, the runtime difference between Theta* algorithms and A* became more pronounced.

The smoothed A* algorithm maintained a similar relative runtime compared to the unweighted grid case, which is expected, since the transition to a weighted grid has little effect on the smoothing procedure itself. However, the cost of the resulting path increased by 21%, as the only nodes operated on by the smoothing procedure are those previously identified by the A* algorithm, which does not account for cell weights in the same manner as any-angle pathfinding.

Basic Theta* exhibits a twofold increase in pathfinding time. This is primarily due to the evaluation of the fourth path cost case, which involves applying the fast voxel traversal algorithm to calculate path cost during any-angle traversal across the grid. In return, however, the algorithm achieves an 8.5% reduction in path cost compared to A*, which is a significant improvement – especially when contrasted with the 3% improvement reported in [8]. Considering that the algorithm still operates within a millisecond timescale, this improvement may justify the use of Theta* on a weighted grid.

Lazy Theta*, on the other hand, shows a 20% increase in path cost and a 120% increase in runtime compared to baseline A*. As Lazy Theta* relies on the triangle inequality and optimistically assumes that it always holds, it fails to perform correctly on weighted grids where this assumption no longer holds true. For correct usage, adjustments described in [10] should be considered.

Strict Theta* was expected to behave similarly to Lazy Theta*, since its primary strategy involves identifying tightly wrapping paths – an approach thought to be less relevant in weighted grids, where direction changes are not necessarily constrained to obstacle corners for optimality. However, it appears that the penalty distances enforced by Strict Theta* help guide the search more effectively – allowing it to reach goals faster than Basic Theta*, especially over shorter distances. The trade-off is a slightly higher path cost than that of Basic Theta*.

After conducting additional testing exclusively over long distances (with start and end points placed at opposite edges of a grid twice the original size), the following results were obtained:

Basic Theta*: Execution time = 11.8 ms;
Path cost = 115.2 m.

Strict Theta*: Execution time = 17.3 ms;
Path cost = 117.1 m.

These results indicate that Strict Theta* can indeed find paths faster than Basic Theta* on a weighted grid – but only when the grid size is relatively small. At a certain threshold, the accumulated penalty distances in the priority queue begin to interfere with pathfinding efficiency.

A promising direction for improving Strict Theta* in the context of weighted grids lies in introducing dynamically adjusted penalty values during direction changes. To achieve this, one must define a penalty computation function tailored to the current map, and then modulate the penalty from 0 up to a predefined value based on the weight of the cell being wrapped. This approach would enable the algorithm to prune more paths that exhibit suboptimal direction changes and yield a more performant implementation – comparable to the results presented in Table 2 – while remaining applicable to grids of arbitrary size.

Figure 7 illustrates paths generated by the A*, A* with smoothing, and Basic Theta* algorithms. It is evident that A* progresses in a “staircase” fashion due to its restriction to eight movement directions between neighboring vertices. A with smoothing* generates more natural-looking paths based solely on A*'s explored vertices, yet this comes at the cost of higher overall path cost. In contrast, Basic Theta* has the flexibility to traverse in any direction between vertices on the grid, which allows it to discover shorter and more visually plausible paths. The advantage of Basic Theta* becomes even more pronounced when weights are overlaid on the grid – clearly showing how the path bypasses high-cost cells (Figure 8).

Given the low absolute execution time (on the order of milliseconds) even for relatively large grid maps, Basic Theta* proves to be a viable solution for large-scale projects. It offers a simple implementation, low-cost paths, and realistic path shapes, making it especially suitable for applications where visual plausibility and path optimality outweigh minimal time savings.

Generalization of Theta* application in weapon supply logistics involves the use of a weighted grid map as the routing space. The delivery region map (e.g., a city or an administrative area) can be divided into a square grid, where each cell is assigned a weight that reflects parameters such as road surface condition (asphalt, cobblestone, dirt), risk level (especially high risk in border zones or areas with increased criminal activity), average traffic, as well as the presence of surveillance cameras, checkpoints, or patrol units, which is particularly relevant under military or special delivery conditions. A detailed description of this procedure implies that the delivery region map can be represented as a Weighted Grid Map, implementing a spatial model of the environment in the form of a two-dimensional grid. In this grid, each cell corresponds to a specific square segment of space (e.g., 10×10 or 50×50 meters – depending on the selected scale).

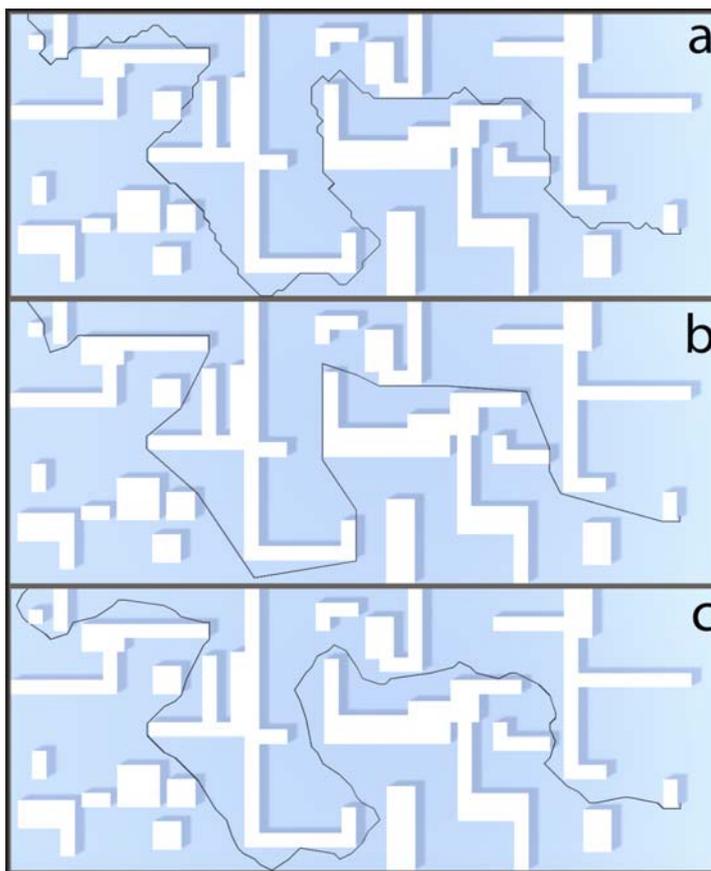


Figure 7 – Paths generated by the algorithms: a – A*, b – A* with smoothing, c – Theta*

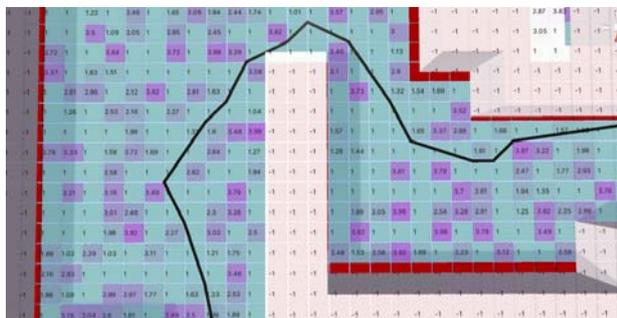


Figure 8 – The path found by the Basic Theta* algorithm is plotted on top of a square grid with weights

5 RESULTS

Each cell is assigned a numerical weight $w(i, j)$, which determines the cost of movement through that cell and directly influences the results of the Theta* pathfinding algorithm. The formula for calculating the cell's weight accounts for the selected factors and may include both road-related and logistics-security characteristics of the environment formula (4).

The weighting coefficients $\alpha, \beta, \gamma, \delta$ in the weighted grid formula are used to determine the relative significance of each factor in computing the total traversal cost of a grid cell (Figure 9). Their selection depends on the routing objectives, application context (e.g., military logistics vs. civilian delivery), and priorities such as safety, time, or cost.

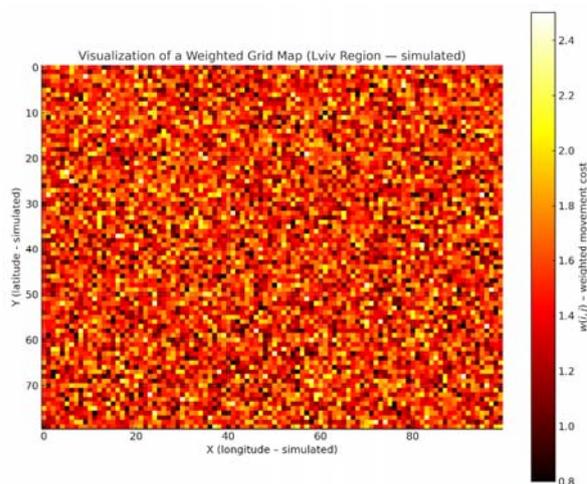


Figure 9 – Visualization of a Weighted Grid Map (Lviv Region – simulated)

The coefficients $\alpha, \beta, \gamma, \delta$ can be determined using several approaches. The first method is expert-based tuning, in which values are assigned by security specialists, logisticians, or analysts based on the specific nature of the cargo. For example, in the context of weapon delivery, security takes the highest priority, which may result in settings such as $\alpha = 0.5$, since road quality is important but not critical; $\beta = 3.0$, indicating risk is treated as a critical factor; $\gamma = 1.5$, reflecting moderate importance of traffic; and $\delta = 2.0$, emphasizing the impact of surveil-

lance infrastructure such as cameras and checkpoints on route selection.

The second approach involves normalization and weight summation, where all coefficients are scaled so that $\alpha + \beta + \gamma + \delta = 1$. This enables preservation of relative importance while remaining independent of absolute values.

The third method is the Analytic Hierarchy Process (AHP), which involves pairwise comparisons of factors by importance (e.g., security is more important than traffic, thus $\beta > \gamma$), followed by the construction of a priority matrix and computation of a weight vector.

Finally, the fourth approach is data-driven learning. Given access to historical data on routes, risks, and incidents, regression models or neural networks can be applied to derive coefficients that minimize cost, time, or delivery risk. Logistic regression or other machine learning techniques may be used for this purpose.

The coefficients α, β, γ and δ define the routing priorities. They should be adjusted depending on the target scenario: secure delivery, speed, stealth, etc. (Table 4).

The best results are achieved when combining expert analysis, normalization, and – when possible – machine learning based on historical data.

For example, the road surface condition parameter $r_{i,j}$ can be derived from satellite imagery and classified as follows:

- asphalt $r_{i,j} = 1.0$;
- cobblestone $r_{i,j} = 1.5$;
- dirt road $r_{i,j} = 2.5$;
- impassable section $r_{i,j} = \infty$ (blocked).

Table 4 – Example of Weights for Weapon Delivery (High-Risk Scenario)

Parameter	Priority Level	Weight (Expert-Based)
Road Condition r	Medium	$\alpha = 0.2$
Risk Level t	High	$\beta = 0.4$
Traffic s	Low	$\gamma = 0.1$
Surveillance v	High	$\delta = 0.3$
Road Condition r	Medium	$\alpha = 0.2$

The risk level parameter $t_{i,j}$ defines the likelihood of cargo loss or obstacles along the route. For instance:

- low risk (central areas) $t_{i,j} = 0.5$;
- medium risk (suburbs, industrial zones) $t_{i,j} = 1.0$;
- high risk (border or high-crime zones) $t_{i,j} = 2.0 - 3.0$.

The average traffic parameter $s_{i,j}$ based on Google Traffic data, reflects possible delays caused by congestion or high road occupancy:

- free-flowing traffic $s_{i,j} = 0.8$;
- moderate traffic $s_{i,j} = 1.2$;
- heavy traffic 1. $s_{i,j} = 1.8$.

The surveillance/control parameter $v_{i,j}$, derived from municipal safety maps, indicates the presence of surveillance cameras, checkpoints, or patrol units:

- no surveillance $v_{i,j} = 0.5$;
- moderate control $v_{i,j} = 1.0$;
- high surveillance (restricted access zones) $v_{i,j} = 2.5$.

The optimal delivery route is constructed based on the minimal cumulative path weight, rather than merely the geometric distance. This requires dynamic weight updates – e.g., through APIs such as Google Traffic or public service data. Such spatial modeling allows the integration of logistical risks into the routing process, which is particularly relevant for arms delivery, where security, efficiency, and route realism are critical.

The choice of algorithm depends on the specific needs of the logistics scenario. The Basic Theta* algorithm is appropriate when the priority is to generate a short route that accounts for realistic turns – especially important for delivering heavy or bulky cargo where frequent or sharp maneuvers should be avoided. It is the most versatile option and is well suited for minimizing delivery costs.

Lazy Theta* is not recommended for weighted grid maps due to its lower accuracy in path cost estimation. However, it may be used in real-time applications where rapid path generation is critical.

Strict Theta* is justified in dense urban environments, where the route must accurately bypass obstacles such as buildings, restricted zones, or barriers. This makes it effective for deliveries to central city areas or in situations with frequent movement restrictions.

An important aspect is the customization of metrics and weights: the weight of a cell may incorporate travel cost (fuel expenses, tolls, duties), delay probability (queues, weather), and the risk of loss or confiscation – especially relevant for sensitive cargo such as firearms.

Theta* has several advantages that make it suitable for such use cases: routes appear realistic for navigation; the algorithm adapts well to dynamic risk maps or traffic conditions (Figure 10); it integrates easily with AI-based systems for risk assessment or decision-making.

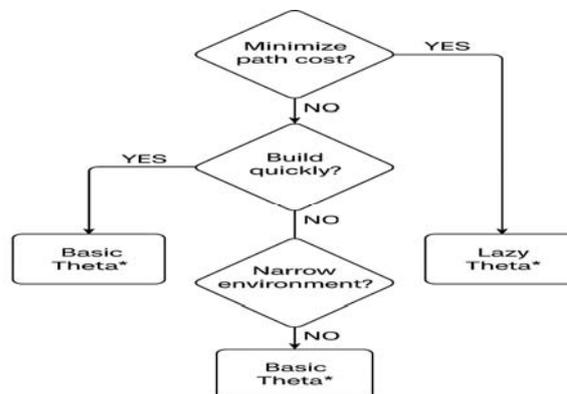


Figure 10 – Flowchart for selecting the algorithm based on operational conditions

From a technical perspective, the algorithm can be integrated into a Supply Chain Management (SCM) system, enabling: route planning with policy and constraint compliance; integration with GPS tracking; visualization of results in the logistics dashboard; use in simulation-based training for logistics personnel.

The study of Theta* algorithms, particularly their adaptation for weighted grid maps, opens new opportunities for optimizing supply routes – especially for sensitive goods such as weapons, where accuracy, reliability, security, and path realism are of critical importance. When combined with GIS, AI, and risk management systems, these algorithms become the core of modern intelligent logistics.

This UML component diagram illustrates the integration of the routing module into the firearm sales management information system. The main component, WeaponSalesManagementSystem, is responsible for managing the entire order and delivery cycle, as well as for interacting with other modules. The RoutePlanningModule is the central element for delivery route construction and includes three main subsystems (Figure 11):

WeightedGridMapEngine, which generates a weighted grid map of the area where each cell accounts for factors such as road conditions, security risks, traffic congestion, and the presence of cameras or checkpoints;

ThetaStarPathfinder, which implements the pathfinding algorithm (e.g., Basic Theta*) and enables optimal route construction based on direct line-of-sight;

RiskAnalyzer, which evaluates the safety level of the route using data from external sources and integration with the security module.

The SecurityModule verifies whether the proposed route is safe for transporting firearms by analyzing information on restricted zones, checkpoints, high-risk areas, and other constraints. Once the route is validated, it is passed to the LogisticsDashboard, where it is visualized for the logistics operator. This interface allows for comparing alternative routes, viewing satellite imagery, and manually adjusting the route if necessary.

The GISDataProvider is a separate service or API that supplies up-to-date geospatial information (e.g., from Google Maps or government sources), which is required for accurate grid generation and real-time risk evaluation.

The information system initiates a route planning request [16, 17] and sends it to the RoutePlanningModule, where all processing and analysis are performed. Then, the SecurityModule checks the safety of the route, and finally, the validated route is forwarded to the logistics dashboard. This integration ensures adaptive, secure, and efficient route planning for the delivery of firearms, accounting for real-world risks and environmental constraints.

6 DISCUSSION

The conducted experiments highlighted both the strengths and limitations of applying any-angle Theta* algorithms to weighted grid maps. A key finding is that the Basic Theta* algorithm consistently reduced overall path cost compared to the classical A*, with only a moderate increase in computation time. This confirms earlier research [7] but shows a stronger improvement in weighted environments, which more accurately reflect real-world conditions for safe and efficient route modeling.

By contrast, the Lazy Theta* algorithm proved unsuitable for weighted environments. Its assumption of universal line-of-sight, valid in uniform grids, led to inaccurate cost estimations when weights varied between cells. This caused both runtime and path cost to increase significantly. Lazy Theta* may still be applied in scenarios where rapid computation is critical, but its reliability for cost-aware route planning is limited.

Strict Theta* produced partially unexpected results. While initially thought to offer little benefit in weighted grids, its penalty mechanism improved performance in certain contexts, making its execution time competitive, especially in smaller maps. Although slightly less cost-efficient than Basic Theta*, it offers practical value in constrained or urban-like environments, where precise obstacle avoidance is important.

These findings suggest that none of the Theta* variants is universally optimal across all weighted environments. Instead, algorithm selection should depend on the operational scenario. Basic Theta* is most suitable for general tasks of route modeling where cost minimization and path realism are crucial. Strict Theta* is more relevant in dense environments, while Lazy Theta* is applicable only when rapid path generation outweighs accuracy.

Importantly, the study demonstrates how Theta* algorithms can be integrated into an information system for tracking and managing firearms sales. By representing road conditions, risk levels, and surveillance density as weighted parameters of the grid, the system can model delivery routes that are both safe and efficient. This ensures that the route planning module not only accounts for the shortest distance but also incorporates safety-related constraints – such as avoiding high-risk or restricted areas.

In summary, the presented results advance the theoretical understanding of any-angle algorithms in weighted environments and confirm their practical applicability within intelligent information systems. Specifically, the adaptation of Basic and Strict Theta* algorithms provides a foundation for developing route-planning modules in

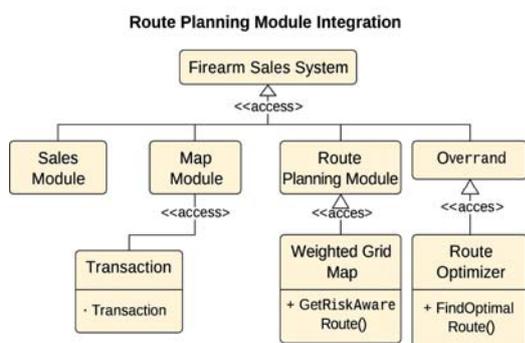


Figure 11 – Route Planning Module Integration

firearm sales management systems, where accuracy, adaptability, and safety of modeled routes are critical.

CONCLUSIONS

This study comprehensively analyzed the efficiency and adaptability of Theta* algorithms (Basic Theta*, Lazy Theta*, and Strict Theta*) when applied to path planning tasks on weighted grid maps. The insights gained not only advance the theoretical understanding of any-angle pathfinding in variable terrain environments but also open up promising avenues for practical implementation – particularly in systems that require high levels of precision, safety, and adaptability.

One of the most promising directions for applying the results of this research is in the development of intelligent route-planning modules within information systems for firearm sales and distribution. In such systems, it is critical to ensure safe, efficient, and legally compliant delivery of weapons across diverse geographic zones, including urban, rural, and high-risk areas.

The integration of the Weighted Grid Map (WGM) model into a firearm logistics information system would enable:

Spatial modeling of delivery routes with real-time adjustments to risk levels, road accessibility, and surveillance density.

Route optimization for safety, using Basic Theta* to avoid high-risk zones, congested areas, or regions with heavy surveillance (e.g., police checkpoints, border patrols).

Dynamic recalculation of routes in case of sudden geopolitical changes or traffic incidents, via integration with real-time data sources such as GIS layers or traffic APIs.

Use of customizable weights and parameters (α , β , γ , δ) for tailoring delivery priorities – e.g., prioritizing stealth in certain operations, or speed and fuel cost in others.

Seamless integration with sales records, enabling automated planning of delivery routes immediately after a weapon transaction is registered in the database.

Furthermore, the results of this study – especially the demonstrated efficiency of the Basic Theta* algorithm on weighted grids – confirm the feasibility of embedding this routing logic into an enterprise-grade information system. This includes systems developed in Java-based frameworks (such as Spring Boot) that already support modules for CRM, logistics, and compliance.

The visual validation via Unity simulations enhances trust in the system's realism and robustness, and such simulations could be used in training environments for logistics personnel, security forces, or system operators.

In conclusion, the Theta* algorithm family – particularly Basic Theta* – can serve as the core routing mechanism in intelligent firearm logistics platforms. When integrated with risk-aware mapping and real-time data, this approach can significantly enhance the security, flexibility, and effectiveness of weapon delivery opera-

tions, contributing to national security, regulatory compliance, and operational excellence.

The study highlighted that traditional pathfinding algorithms such as A* may not sufficiently address the complex risk factors and real-time constraints encountered in secure logistics, particularly for the delivery of high-risk goods such as firearms. Therefore, it is recommended that weapon supply chain management systems incorporate weighted grid maps and advanced routing algorithms like Basic Theta*. These approaches better reflect real-world road and threat conditions and provide more realistic and adaptive routing.

Given the findings, it is advisable to train logistics personnel and system developers in the application of spatial modeling, route risk evaluation, and the customization of routing parameters based on regional security and traffic data. Additionally, it is recommended to integrate route planning modules with real-time GIS and security data sources to ensure the adaptability of planned routes to changing field conditions.

Furthermore, as the visual realism and flexibility of Theta* algorithms support decision-making, it is recommended to use these methods not only in operational logistics systems but also in training simulators for logistics officers. This will promote awareness of dynamic routing factors and increase the security of firearm delivery operations.

Finally, regulatory agencies and private logistics providers should collaborate on the standardization of risk-weighted spatial models, enabling broader adoption of intelligent route planning technologies in national and regional weapon distribution systems.

DECLARATIONS

Conflict of interest: The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship, or otherwise, that could affect the research and its results presented in this paper.

Authors' contributions: Yrii Kis contributed to the conceptualization of the study, development and implementation of the Theta* algorithms, simulation experiments in the Unity environment, data collection, and primary drafting of the manuscript; Yrii Shcherbyna supervised the research methodology, contributed to the formalization of mathematical models, analysis of algorithmic performance, and critically reviewed the manuscript for theoretical consistency; Nataliia Kunanets contributed to the system-level design and integration of the routing module into the firearm sales management information system, participated in defining applied use cases, and reviewed the manuscript from the perspective of information systems and applied logistics, and assisted in manuscript editing and technical refinement; Yrii Yarymovych contributed to experimental design, data analysis and visualization, validation of results.

Data availability: The manuscript does not have associated data in a data repository.

Use of artificial intelligence tools: The authors confirm that they did not use artificial intelligence technologies in creating the submitted work.

REFERENCES

1. Van Den Berg J., Shah R., Huang A., Goldberg K. Anytime nonparametric A, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2011, Vol. 25, № 1, pp. 105–111. Mode of access: <https://doi.org/10.1609/aaai.v25i1.7819>.
2. Sturtevant N., Geisberger R. A comparison of High-Level approaches for speeding up pathfinding [Electronic resource], *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2010, Vol. 6, № 1, pp. 76–82. Mode of access: <https://doi.org/10.1609/aiide.v6i1.12400>.
3. Subbotin S. A. Construction of decision trees for the case of low-information features, *Radio Electronics, Computer Science, Control*, 2019, № 1, pp. 121–130.
4. Garcia-Luna-Aceves J., Murthy N. S. A path-finding algorithm for loop-free routing [Electronic resource], *IEEE/ACM Transactions on Networking*, 1997, Vol. 5, № 1, pp. 148–160. Mode of access: <https://doi.org/10.1109/90.554729>.
5. Lawande S. R., Jasmine G., Anbarasi J. et al. A systematic review and analysis of Intelligence-Based pathfinding algorithms in the field of video games, *Applied Sciences*, 2022, Vol. 12, № 11, P. 5499. Mode of access: <https://doi.org/10.3390/app12115499>.
6. Hart P., Nilsson N., Raphael B. A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on Systems Science and Cybernetics*, 1968, Vol. 4, № 2, pp. 100–107. Mode of access <https://doi.org/10.1109/tscc.1968.300136>.
7. Fraile-Jurado P., Llovet-Ferrer M., Roig-Munar F. X. Toward Realism: An analysis of coastal environments in Open-World Video Games, *Simulation & Gaming*, 2024, Mode of access: <https://doi.org/10.1177/10468781241287900>.
8. Daniel K., Nash A., Koenig S., Felner A. Theta: Any-Angle path planning on grids, *Journal of Artificial Intelligence Research*, 2010, Vol. 39, pp. 533–579. Mode of access: <https://doi.org/10.1613/jair.2994>.
9. Le P. T., Lee K. Weight value and map complexity in Theta* [Electronic resource], *MATEC Web of Conferences*, 2016, Vol. 54, P. 05003. Mode of access: <https://doi.org/10.1051/mateconf/20165405003>.
10. Rey R., Cobano J. A., Merino L., Caballero F. Generalized Lazy-Theta for 3D path planning considering non-uniform costs, *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022. Mode of access: <https://doi.org/10.1109/icuas54217.2022.9836069>.
11. Han S., Wang L., Y. Wang, H. He A dynamically hybrid path planning for unmanned surface vehicles based on non-uniform Theta and improved dynamic windows approach, *Ocean Engineering*, 2022, Vol. 257, P. 111655. Mode of access: <https://doi.org/10.1016/j.oceaneng.2022.111655>.
12. Nash A., Koenig S., Tovey C. Lazy Theta*: Any-Angle path planning and path length analysis in 3D [Electronic resource], *Proceedings of the AAAI Conference on Artificial Intelligence*, 2010, Vol. 24, № 1, pp. 147–154. Mode of access: <https://doi.org/10.1609/aaai.v24i1.7566>.
13. Oh S., Leong H. W. Strict Theta*: Shorter motion path planning using taut paths [[Electronic resource], *Proceedings of the International Conference on Automated Planning and Scheduling*, 2016, Vol. 26, pp. 253–257. Mode of access: <https://doi.org/10.1609/icaps.v26i1.13744>.
14. Amanatides J., Woo A. A Fast Voxel Traversal Algorithm for Ray Tracing. York University, 1987. Mode of access: <http://www.cs.yorku.ca/~amana/research/grid.pdf>
15. Kis Yu. O. Practicality and efficiency of application of weighted freely directed path finding algorithms Theta*: Qualification (master's) thesis. Lviv, 2024, 54 p.
16. Methods and characteristics of locality-preserving transformations in the problems of computational intelligence, *Radio Electronics, Computer Science, Control*, 2014, № 21, pp. 120–128.
17. Oliinyk A., Subbotin S., Lovkin V. et al. The system of criteria for feature informativeness estimation in pattern recognition, *Radio Electronics, Computer Science, Control*, 2017, № 18, pp. 85–96.

Received 14.10.2025.
Accepted 14.01.2026.
Published 27.03.2026.

УДК 004.421.2:519.6:004.89

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ АЛГОРИТМІВ ANY-ANGLE ТЕТТА НА ЗВАЖЕНИХ СІТОЧНИХ КАРТАХ ДЛЯ ПЛАНУВАННЯ МАРШРУТІВ*

Кіс Ю. – аспірант кафедри дискретного аналізу та інтелектуальних систем, Львівський національний університет імені Івана Франка, Україна. ROR: <https://ror.org/01s7y5e82>. ORCID: <https://orcid.org/0009-0009-7816-237X>.

Щербина Ю. М. – канд. техн. наук, професор кафедри дискретного аналізу та інтелектуальних систем, Львівський національний університет імені Івана Франка, Україна. ROR: <https://ror.org/01s7y5e82>. ORCID: <https://orcid.org/0000-0002-4942-2787>.

Кунанець Н. С. – д-р техн. наук, професор кафедри інформаційних систем та мереж, Національний університет «Львівська політехніка», Львів, Україна. ROR: <https://ror.org/0542q3127>. ORCID: <https://orcid.org/0000-0003-3007-2462>.

Ярмівич Ю. А. – аспірант кафедри інформаційних систем та мереж, Національний університет «Львівська політехніка», Львів, Україна. ROR: <https://ror.org/0542q3127>. ORCID: <https://orcid.org/0009-0006-1391-3214>.

АНОТАЦІЯ

Актуальність. У статті розглядається дослідження алгоритмів пошуку шляхів вільного напрямку, зокрема алгоритму Theta*, та оцінюється їхня ефективність на зважених сіточних картах з метою визначення оптимальних маршрутів для доставки товарів до магазину зброї. Це дослідження виконане в ширшому контексті розробки інформаційної системи для відстеження та управління продажем і логістикою зброї в складних умовах. Одним із головних мотивів є те, що методи будь-якого кута (any-angle) здатні генерувати більш реалістичні та природні маршрути порівняно з класичним алгоритмом A*.

Мета. Метою дослідження є аналіз ефективності трьох алгоритмів, заснованих на Theta*: Basic Theta*, Lazy Theta* і Strict Theta*, як на однорідних, так і на зважених квадратних сітках, із особливим акцентом на показниках часу виконання та вартості шляху. Робота спрямована на узагальнення застосовності цих алгоритмів до зважених середовищ і пропозицію удосконалень, придатних для реальних сценаріїв планування маршрутів.

Метод. Представлено принципи роботи алгоритму A^* , трьох варіантів Theta*, а також технік згладжування маршрутів після обчислення. У дослідженні описано перехід від незважених однорідних квадратних сіток до зважених і акцентовано увагу на складності обчислення точних вартостей маршрутів при застосуванні методів будь-якого кута. Візуалізація поведінки алгоритмів була реалізована за допомогою рушія Unity. Показники ефективності вимірювалися окремо для однорідних і зважених сіток, щоб забезпечити порівняльний аналіз.

Результати. Отримані результати включають порівняльну оцінку алгоритмів Basic Theta*, Lazy Theta*, Strict Theta* та класичного A^* . Аналіз виявив умови, за яких кожен алгоритм працює ефективно, а також фактори, що обмежують їх застосовність у зважених середовищах. Показано, що довжина маршруту та його вартість можуть суттєво відрізнятися на зважених сітках, що призводить до нових міркувань щодо оптимізації на основі вартості. На основі експериментів запропоновано узагальнення алгоритму Basic Theta* для підвищення його придатності до зважених квадратних сіток, а також окреслено можливе розширення алгоритму Strict Theta* для цього контексту.

Висновки. Результати дослідження показують, що, хоча алгоритми будь-якого кута забезпечують більш плавні та реалістичні маршрути, їх ефективність у зважених середовищах залежить від ретельної адаптації функцій вартості. Дослідження підкреслює їхню цінність не лише для моделювання складних віртуальних середовищ і поведінки агентів у іграх та робототехніці, а й для практичних застосувань у логістиці, зокрема в розробці інформаційної системи для відстеження та управління продажем зброї. Запропоновані алгоритмічні удосконалення можуть сприяти підвищенню ефективності планування доставки та управління ланцюгами постачання, у тому числі моделюванню маршрутів доставки зброї в умовах воєнного часу.

КЛЮЧОВІ СЛОВА: пошук шляху, планування маршруту, квадратна сітка, алгоритм будь-якого кута, вартість шляху, зважена сітка, Theta*.

ЛІТЕРАТУРА

1. Anytime nonparametric A^* / [J. Van Den Berg, R. Shah, A. Huang, K. Goldberg] // Proceedings of the AAAI Conference on Artificial Intelligence. – 2011. – Vol. 25, № 1. – P. 105–111. – Mode of access: <https://doi.org/10.1609/aaai.v25i1.7819>.
2. Sturtevant N. A comparison of High-Level approaches for speeding up pathfinding [Електронний ресурс] / N. Sturtevant, R. Geisberger // Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. – 2010. – Vol. 6, № 1. – P. 76–82. – Mode of access: <https://doi.org/10.1609/aiide.v6i1.12400>.
3. Subbotin S. A. Construction of decision trees for the case of low-information features / S. A. Subbotin // Radio Electronics, Computer Science, Control. – 2019. – № 1. – P. 121–130.
4. Garcia-Luna-Aceves J. A path-finding algorithm for loop-free routing [Електронний ресурс] / J. Garcia-Luna-Aceves, N. S. Murthy // IEEE/ACM Transactions on Networking. – 1997. – Vol. 5, № 1. – P. 148–160. – Mode of access: <https://doi.org/10.1109/90.554729>.
5. A systematic review and analysis of Intelligence-Based pathfinding algorithms in the field of video games / [S. R. Lawande, G. Jasmine, J. Anbarasi et al.] // Applied Sciences. – 2022. – Vol. 12, № 11. – P. 5499. – Mode of access: <https://doi.org/10.3390/app12115499>.
6. Hart P. A formal basis for the heuristic determination of minimum cost paths / P. Hart, N. Nilsson, B. Raphael // IEEE Transactions on Systems Science and Cybernetics. – 1968. – Vol. 4, № 2. – P. 100–107. – Mode of access: <https://doi.org/10.1109/tssc.1968.300136>.
7. Fraile-Jurado P. Toward Realism: An analysis of coastal environments in Open-World Video Games / P. Fraile-Jurado, M. Llovet-Ferrer, F. X. Roig-Munar // Simulation & Gaming. – 2024. – Mode of access: <https://doi.org/10.1177/10468781241287900>.
8. Theta: Any-Angle path planning on grids / [K. Daniel, A. Nash, S. Koenig, A. Felner] // Journal of Artificial Intelligence Research. – 2010. – Vol. 39. – P. 533–579. – Mode of access: <https://doi.org/10.1613/jair.2994>.
9. Le P. T. Weight value and map complexity in Theta* [Електронний ресурс] / P. T. Le, K. Lee // MATEC Web of Conferences. – 2016. – Vol. 54. – P. 05003. – Mode of access: <https://doi.org/10.1051/mateconf/20165405003>.
10. Generalized Lazy-Theta for 3D path planning considering non-uniform costs / [R. Rey, J. A. Cobano, L. Merino, F. Caballero] // 2022 International Conference on Unmanned Aircraft Systems (ICUAS). – 2022. – Mode of access: <https://doi.org/10.1109/icuas54217.2022.9836069>.
11. A dynamically hybrid path planning for unmanned surface vehicles based on non-uniform Theta and improved dynamic windows approach / [S. Han, L. Wang, Y. Wang, H. He] // Ocean Engineering. – 2022. – Vol. 257. – P. 111655. – Mode of access: <https://doi.org/10.1016/j.oceaneng.2022.111655>.
12. Nash A. Lazy Theta*: Any-Angle path planning and path length analysis in 3D [Електронний ресурс] / A. Nash, S. Koenig, C. Tovey // Proceedings of the AAAI Conference on Artificial Intelligence. – 2010. – Vol. 24, № 1. – P. 147–154. – Mode of access: <https://doi.org/10.1609/aaai.v24i1.7566>.
13. Oh S. Strict Theta*: Shorter motion path planning using taut paths [Електронний ресурс] / S. Oh, H. W. Leong // Proceedings of the International Conference on Automated Planning and Scheduling. – 2016. – Vol. 26. – P. 253–257. – Mode of access: <https://doi.org/10.1609/icaps.v26i1.13744>.
14. Amanatides J. A Fast Voxel Traversal Algorithm for Ray Tracing / J. Amanatides, A. Woo. – York University, 1987. – Mode of access: <http://www.cs.yorku.ca/~amana/research/grid.pdf>
15. Kis Yu. O. Practicality and efficiency of application of weighted freely directed path finding algorithms Theta*/ Yu. O. Kis. – Qualification (master's) thesis. – Lviv, 2024. – 54 p.
16. Subbotin S. A. Methods and characteristics of locality-preserving transformations in the problems of computational intelligence / S. A. Subbotin // Radio Electronics, Computer Science, Control. – 2014. – № 21. – P. 120–128.
17. The system of criteria for feature informativeness estimation in pattern recognition / [A. Oliinyk, S. Subbotin, V. Lovkin et al.] // Radio Electronics, Computer Science, Control. – 2017. – № 18. – P. 85–96.