https://doi.org/10.31891/2307-5732-2026-363-49
УДК 004.67

КУХТА ОЛЕКСАНДР
Національний лісотехнічний університет України
https://orcid.org/0009-0000-5520-2740
e-mail: kukhta.o@nltu.lviv.ua
ПІРКО ІГОР
Національний лісотехнічний університет України
https://orcid.org/0009-0008-2378-2929
e-mail: pirko@nltu.edu.ua

ДИНАМІЧНЕ ЗВАЖУВАННЯ РИЗИКІВ У ІЄРАРХІЧНИХ СЕРЕДОВИЩАХ З РУХОМИМИ ПЕРЕШКОДАМИ

У роботі розглянуто актуальну проблему побудови безпечних і стабільних маршрутів для автономних агентів у складних динамічних середовищах. Метою дослідження є підвищення ефективності та інтелектуальності навігації шляхом інтеграції механізму динамічного зважування ризиків у дворівневу ієрархічну модель планування шляху. У межах дослідження розроблено математичну модель, яка визначає вартість проходження ієрархічних кластерів на основі коефіцієнта інтенсивності руху перешкод. Це дозволяє здійснювати оцінювання безпеки маршруту вже на етапі високорівневого планування та уникати потенційно небезпечних зон.

Методи дослідження базуються на ієрархічній декомпозиції простору, застосуванні алгоритму пошуку під довільним кутом (Theta) для забезпечення плавності траєкторій, а також на реалізації розробленої математичної моделі для динамічного коригування вагових коефіцієнтів графа в реальному часі.*

Встановлено, що практичне застосування запропонованої моделі у структурованих сценаріях дозволяє зменшити кількість зіткнень агента з перешкодами у 4,4 раза, забезпечуючи безперервність руху та високий рівень фізичної безпеки. Виявлено, що попри незначне зростання часу одиничного розрахунку маршруту (в середньому на 38–44%), загальна обчислювальна стабільність системи істотно підвищується за рахунок скорочення кількості процедур перепланування на 68%. Запропонований підхід дає змогу уникнути надлишкового виконання обчислювально витратних процедур обробки колізій, анімаційних переходів та поведінкової логіки реакції агентів, які в ігрових системах споживають значно більше обчислювальних ресурсів, ніж пошук шляху.

Практична значущість отриманих результатів підтверджена програмною реалізацією у середовищі Unity, яка демонструє здатність алгоритму формувати плавні, природні та стратегічно обґрунтовані траєкторії в умовах змінюваного середовища без використання статичного NavMesh. Результати дослідження можуть бути використані під час розроблення складних RTS-ігор, систем моделювання натовпу та систем керування автономними роботами.

Ключові слова: *пошук шляху, динамічне середовище, ієрархічна модель, динамічне зважування ризиків, Theta.*

KUKHTA OLEKSANDR, PIRKO IGOR
Ukrainian National Forestry University

DYNAMIC RISK WEIGHING IN HIERARCHICAL ENVIRONMENTS WITH MOVING OBSTACLES

The paper addresses the critical problem of constructing safe and stable routes for autonomous agents in complex dynamic environments. The aim of the study is to improve the efficiency and intelligence of navigation by integrating a dynamic risk-weighting mechanism into a two-level hierarchical path planning model. Within the scope of this research, a mathematical model was developed to formalize the computation of traversal costs for hierarchical clusters based on the object movement intensity coefficient. This enables strategic safety assessment of routes at the high-level planning stage and facilitates the avoidance of potentially hazardous areas.

The research methodology is based on hierarchical space decomposition, the application of the any-angle pathfinding algorithm (Theta) to ensure trajectory smoothness, and the implementation of the proposed mathematical model for real-time dynamic adjustment of graph weight coefficients.*

It was established that the practical application of the proposed model in structured scenarios reduces the number of agent–obstacle collisions by a factor of 4.4, ensuring continuous motion and a high level of physical safety. Despite a moderate increase in single-path computation time (on average by 38–44%), the overall computational stability of the system is significantly improved due to a 68% reduction in the number of required replanning operations. The proposed approach minimizes redundant execution of resource-intensive collision handling, animation transitions, and behavioral reaction logic, which typically consume substantially more computational resources in game systems than strategic path planning itself.

The practical significance of the obtained results is confirmed by a software implementation in the Unity environment, demonstrating the algorithm's capability to generate smooth, natural, and strategically justified trajectories in dynamic environments without relying on a static NavMesh. The results can be applied in the development of complex RTS games, crowd simulation systems, and autonomous robot control systems.

Keywords: *path finding, dynamic environment, hierarchical model, dynamic risk weighting, Theta.*

Стаття надійшла до редакції / Received 06.02.2026
Прийнята до друку / Accepted 28.02.2026
Опубліковано / Published 26.03.2026

This is an Open Access article distributed under the terms of the Creative Commons CC-BY 4.0

© Кухта Олександр, Пірко Ігор

Вступ

Автономна навігація інтелектуальних агентів у динамічних середовищах є однією з найбільш складних задач у галузі штучного інтелекту та розроблення ігрових і симуляційних систем. Класичні підходи, зокрема алгоритми A* та Dijkstra, орієнтовані на пошук оптимального шляху в статичному графі. Водночас реальні середовища, такі як комп'ютерні симуляції, сучасні відеоігри або логістичні хаби, характеризуються наявністю значної кількості об'єктів, що постійно змінюють своє положення у просторі.

Основним обмеженням класичних методів планування є їх реактивний характер. У разі виникнення динамічної перешкоди агент змушений ініціювати повний або частковий перерахунок траєкторії руху (перепланування). Такий підхід зумовлює два критичні недоліки. По-перше, багаторазові процедури перепланування для великої кількості агентів призводять до значних пікових навантажень на обчислювальні ресурси системи. По-друге, відсутність механізмів передбачення змушує агента обирати геометрично найкоротший маршрут, який часто проходить через зони з високою щільністю руху, що неминуче спричиняє часті колізії та затримки.

Крім того, у складних інтерактивних системах кожен агент виконує значний обсяг обчислювальної логіки, що включає обробку фізичних взаємодій, сенсорний аналіз середовища та прийняття рішень на основі дерев поведінки (Behavior Trees). Перебування агента в областях з високою концентрацією динамічних перешкод призводить до частішого використання сенсорних перевірок (Raycasts) і модулів уникнення зіткнень, що істотно збільшує сумарні часові витрати на один ігровий кадр.

У даній роботі запропоновано підхід, який переводить навігацію агентів із реактивної у проактивну площину. Використання ієрархічної декомпозиції простору та механізму динамічного зважування кластерів на основі активності перешкод дозволяє агенту заздалегідь ідентифікувати потенційно небезпечні ділянки середовища. Це забезпечує не лише зменшення кількості фізичних зіткнень, але й зниження загального обчислювального навантаження на систему за рахунок скорочення частоти перепланувань та спрощення локальної поведінкової логіки агентів.

Об’єкт дослідження — процес пошуку шляху автономних агентів у середовищах із рухомими перешкодами.

Предмет дослідження — алгоритми ієрархічного пошуку шляху, що використовують змінні вагові коефіцієнти для оцінювання рівня безпеки окремих ділянок середовища.

Мета роботи — підвищення стабільності та безпеки руху автономних агентів шляхом розроблення моделі, яка забезпечує попереднє виявлення та оминання зон з підвищеним ризиком зіткнень. Це дозволяє зменшити кількість процедур перепланування маршруту та оптимізувати використання обчислювальних ресурсів системи.

Для досягнення поставленої мети у роботі визначено такі основні завдання дослідження:

- розробити модель ієрархічного пошуку шляху з механізмом динамічного оцінювання ризиків на основі активності перешкод;
- реалізувати запропоновану модель програмними засобами в ієрархічному середовищі з рухомими перешкодами з використанням алгоритму A* на рівні кластерів та алгоритму Theta* на нижньому рівні планування;
- провести серію експериментів у середовищі зі сценарієм «Шосе–Болото» та у сценаріях із випадковою кількістю рухомих перешкод у кластерах;
- виконати порівняльний аналіз результатів за такими показниками, як кількість колізій, частота перепланувань і час обчислень, для конфігурацій із використанням динамічного зважування ризиків та без нього.

Матеріали та методи дослідження

Експериментальну перевірку розробленої моделі здійснено з використанням ігрового рушія Unity та мови програмування C#. Розроблений програмний комплекс забезпечує моделювання руху автономного агента в дискретному просторі з одночасною присутністю значної кількості динамічних перешкод.

Технічні параметри середовища:

- Робоча область: дискретна сітка розміром 190x190 клітинок.
- Ієрархічна структура: простір поділено на кластери розміром 15x15 клітинок.
- Параметри задачі: цільова відстань між стартовою та кінцевою точками становить 200 клітинок.
- Динаміка середовища: оновлення позицій рухомих перешкод відбувається кожні 0.3 секунди, що створює виклик для системи навігації.

Конфігурації сценаріїв тестування:

1. Сценарій «Шосе-Болото»: структуроване середовище, у якому центральна (найкоротша) зона має високу щільність перешкод — 25%, а обхідні шляхи— низьку щільність 3%.
2. Сценарій із випадковою генерацією: середовище з нерівномірним розподілом об'єктів, у якому щільність динамічних перешкод у кластерах варіюється в діапазоні від 5% до 40%.

Вимірювання проводилися за трьома ключовими метриками: час обчислення маршруту (ms), сумарна кількість колізій за час проходження дистанції та кількість викликів процедури повного перепланування шляху (replanning).

Аналіз досліджень та публікацій

Проблема пошуку оптимального шляху в динамічних середовищах є предметом активних досліджень протягом останніх десятиліть. Фундаментом для більшості сучасних рішень є алгоритм A*, запропонований Харт та ін. [1], який забезпечує пошук найкоротшого шляху на графі, проте має обмежену ефективність у великих дискретних просторах через високу обчислювальну складність.

Для вирішення проблеми масштабованості Ботеа та ін. представили концепцію *HPA (Hierarchical Path-Finding)** [2]. Автори довели, що розбиття карти на кластери та побудова абстрактного графу дозволяють значно прискорити обчислення, втрачаючи при цьому лише незначний відсоток оптимальності шляху. Однак оригінальний HPA* розроблявся для статичних середовищ і не враховував ризики, пов'язані з рухомими об'єктами.

Розвиток методів планування під довільним кутом призвів до появи алгоритму Theta* [3]. Неш та ін. продемонстрували, що за допомогою простої перевірки прямої видимості (Line-of-Sight) можна уникнути «зигзагоподібних» траєкторій, характерних для класичних сіткових методів. Це дослідження стало базовим для формування гладких шляхів у сучасних системах навігації.

Питання динамічних перешкод розглядалося у роботах, присвячених алгоритму *D Lite** [4]. Кеніг та Ліхачов розробили метод інкрементального пошуку, який дозволяє ефективно перепланувати шлях при зміні умов. Проте D* Lite зосереджений на змінах у структурі графу, а не на передбачуваному оминанні зон з високою ймовірністю колізій.

Інтеграція ієрархічних структур у динамічні умови була досліджена Харабором та Ботеа [5]. Вони запропонували методи швидкого оновлення абстрактного графу, проте акцент робився на швидкості перерахунку, а не на виборі «безпечних коридорів» на основі щільності трафіку.

Окремий напрямок досліджень стосується методів локального уникнення колізій, таких як ORCA (Optimal Reciprocal Collision Avoidance) [6]. Ці методи ефективно вирішують проблему взаємодії агентів на мікрорівні, але не здатні до глобального планування маршруту через безпечні зони, що часто призводить до потрапляння агентів у «локальні пастки» (скупчення об'єктів).

Сучасні підходи дедалі частіше використовують динамічне зважування ризиків. У роботі [7] автори пропонують модифікацію евристики A*, яка враховує ймовірнісну модель руху перешкод. Подібний підхід було розширено у дослідженні [8], де впроваджено концепцію «полів ризику», проте ієрархічна декомпозиція там не використовувалася, що обмежує застосування методу на великих картах.

Аналіз публікацій [9, 10] показує, що поєднання ієрархічного планування з адаптивним зважуванням зон на основі активності перешкод є перспективним, але недостатньо дослідженим напрямком, особливо в контексті балансу між витратами часу на пошук та зниженням навантаження на логіку агента.

Математична модель алгоритму

Запропонована модель базується на дворівневій ієрархії графів та модифікованій функції вартості, що враховує динаміку середовища.

Ключовою особливістю моделі є функція GetDynamicWeight , яка перетворює інтенсивність руху перешкод у вартість проходження графу. Для кожної клітини сітки n, що належить кластеру C, динамічна вага $w(n)$ розраховується як:

$$w(n) = \begin{cases} 1.0, & \text{якщо } DW = \text{false} \\ 1.0 + (\text{Risk}(C) \cdot \beta), & \text{якщо } DW = \text{true} \end{cases}$$

де:

- $\text{Risk}(C)$ - накопичений показник щільності перешкод у кластері C
- β - лінійний множник впливу ризику ($\beta = 2.0$).

Процес розрахунку маршруту розділений на два рівні абстракції:

1. Високорівневе планування (High-Level A*): Алгоритм знаходить послідовність кластерів $\mathcal{P}_{\text{clusters}} = \{C_1, C_2, \dots, C_k\}$ від стартової точки до фінішу. Функція вартості переходу між сусідніми кластерами C_i та C_j :

$$g(C_j) = g(C_i) + \text{dist}(C_i, C_j) \cdot (1.0 + \text{Risk}(C_j) \cdot \beta)$$

Врахування ризику що обрахований функцією GetDynamicWeight змушує алгоритм ще на етапі стратегічного планування обирати послідовність «безпечних» кластерів.

2. Побудова траєкторії під довільним кутом (Theta*).

На низькому рівні, всередині обраних кластерів, виконується Theta*-пошук. Головна відмінність від A* полягає в правилі оновлення батьківського вузла:

Якщо LineOfSight(parent(n), neighbor) = true, то:

$$g(\text{ neighbor }) = g(\text{ parent }(n)) + \text{dist}(\text{ parent }(n), \text{neighbor }) \cdot w(\text{ neighbor })$$

В іншому випадку (стандартний A*):

$$g(\text{ neighbor }) = g(n) + \text{dist}(n, \text{neighbor }) \cdot w(\text{ neighbor })$$

Також модель використовує оптимізовану перевірку цілісності шляху. Замість повної перевірки всього маршруту, агент аналізує лише обмежений горизонт подій (Look-ahead window) розміром $L = 3$ сегменти:

$$\forall s \in \{S_{\text{current}}, \dots, S_{\text{current}+L}\}: \text{Valid}(s) = \neg\text{Collision}(s) \wedge \text{LineOfSight}(s)$$

Якщо умова Valid(s) порушується, ініціюється процедура репланування.

При виникненні перешкоди алгоритм виконує локальне коригування. Нова траєкторія $\mathcal{T}_{\text{new}}$ інтегрується у вже існуючу $\mathcal{T}_{\text{old}}$ у точці поточної позиції агента P_{current} :

$$\mathcal{T}_{\text{final}} = \{P_1, \dots, P_{\text{current}}\} \cup \text{CalculatePath}(P_{\text{current}}, P_{\text{end}})$$

Результати дослідження та їх обговорення

Експериментальна перевірка запропонованої моделі базувалася на порівнянні ефективності навігації автономного агента за двох умов: із вимкненим та з увімкненим механізмом динамічного зважування ризиків. Основними критеріями оцінювання ефективності виступали часові витрати на пошук шляху, безпека руху (кількість зіткнень) та стабільність обраної траєкторії, що оцінювалася кількістю процедур перепланування. Результати 20 контрольних запусків (по 10 запусків для кожної конфігурації) узагальнено у таблиці 1.

Таблиця 1

Порівняння середніх значень показників ефективності алгоритму

Сценарій тестування	**Режим зважування**	**Середній час (ms)**	**Середня кількість колізії**	**Середня кількість репланувань**
Шосе-Болото	Без зважування	21,24 ms	213	26,3
Шосе-Болото	Динамічне зважування ризиків	29,32 ms	48	8,3
Випадкова кількість перешкод в кожному з кластерів	Без зважування	25,49 ms	150	23,5
Випадкова кількість перешкод в кожному з кластерів	Динамічне зважування ризиків	36,84 ms	175	12,5

Окрім кількісних показників, було проаналізовано просторову структуру побудованих маршрутів. На візуалізаціях межі кластерів позначено сірим кольором, портали переходу між кластерами — червоним, перешкоди — чорним, а побудований маршрут зображено у вигляді градієнтної лінії від синього до зеленого.

На рис. 1 і 2 наведено результати побудови маршрутів у сценарії «Шосе–Болото» відповідно без використання механізму динамічного зважування ризиків та з його застосуванням. Даний сценарій характеризується тим, що геометрично найкоротший маршрут проходить через центральну область з високою щільністю та інтенсивністю руху перешкод, тоді як обхідні маршрути мають значно нижчі значення цих показників.

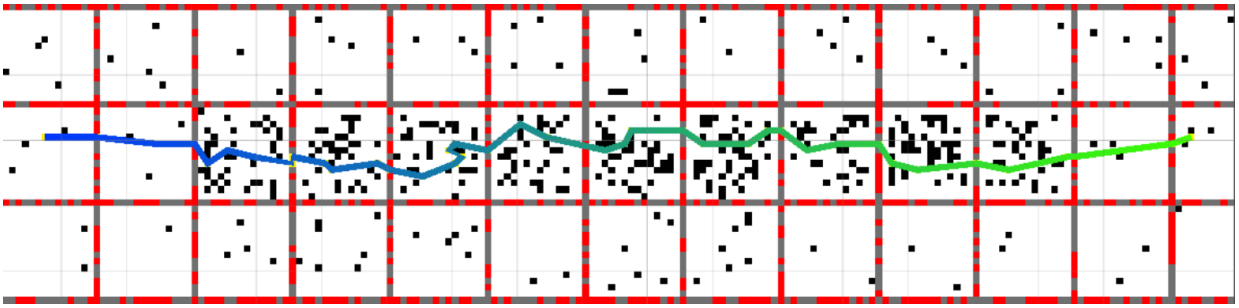

Рис. 1. Маршрут без динамічного зважування у сценарії «Шосе-Болото»

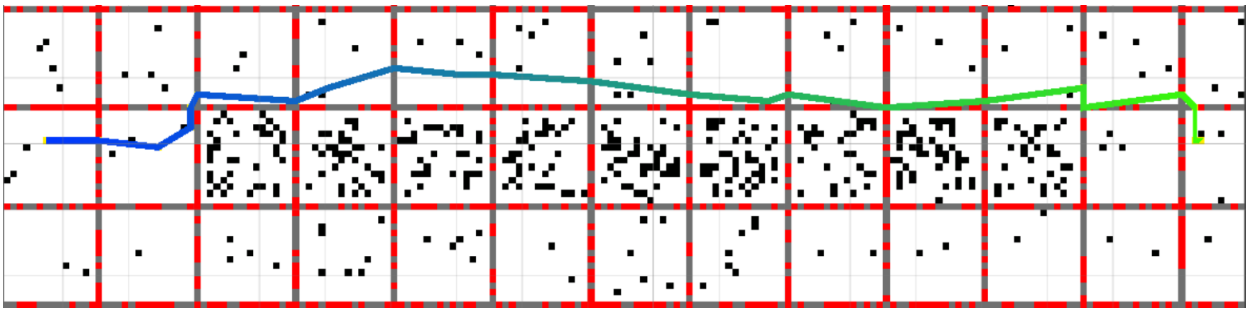

Рис. 2. Маршрут з динамічним зважування у сценарії «Шосе-Болото»

На рис. 3 і 4 представлено побудовані маршрути у сценарії з випадковим розподілом динамічних перешкод. Як видно з наведених прикладів, використання механізму динамічного зважування призводить до формування альтернативних маршрутів, орієнтованих на мінімізацію сумарного ризику, навіть за умови збільшення геометричної довжини шляху.

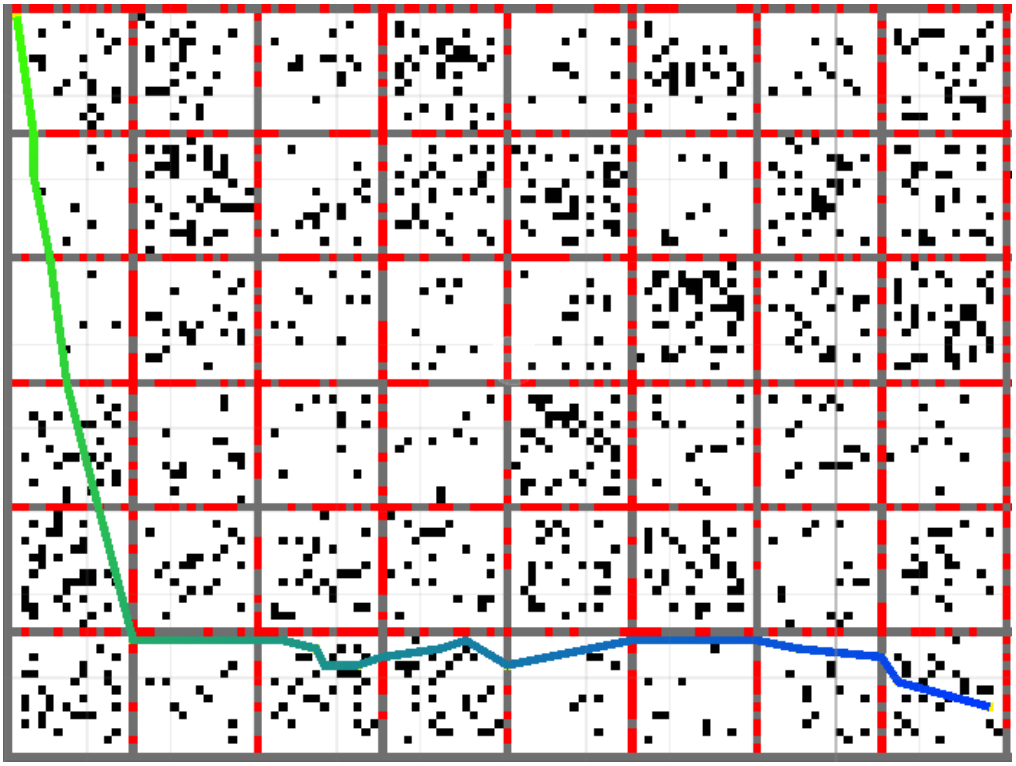

Рис. 3. Маршрут без динамічного зважування у сценарії з випадковою генерацією перешкод

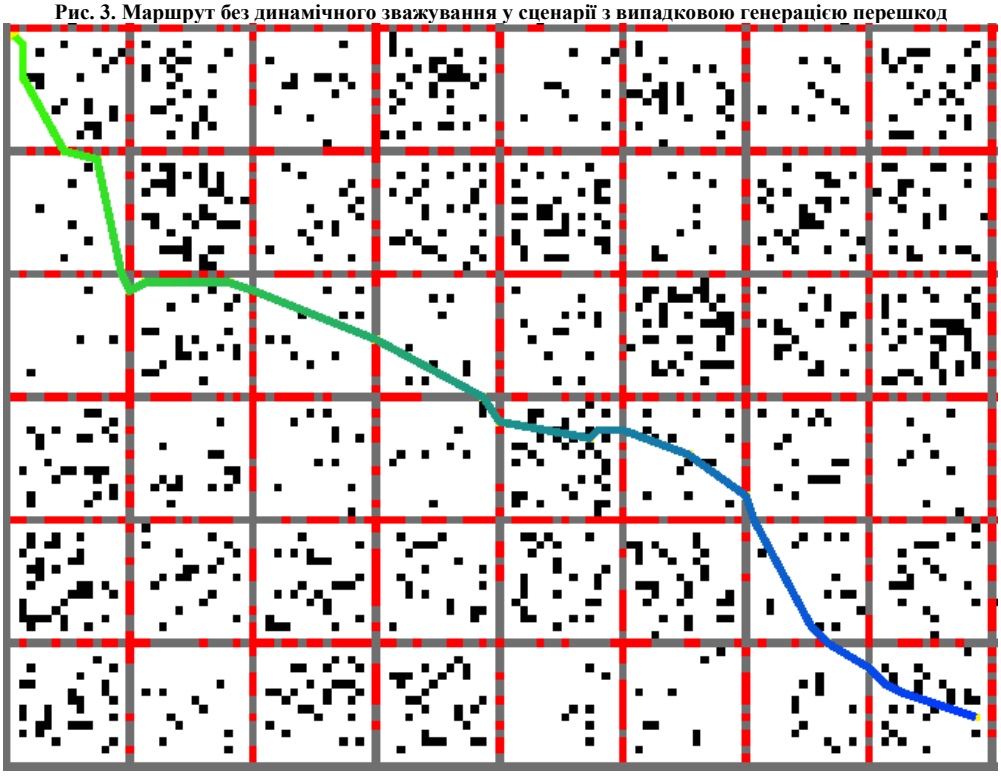

Рис. 4. Маршрут з динамічним зважуванням у сценарії з випадковою генерацією

Аналіз отриманих результатів дослідження

Аналіз статистичних даних, отриманих у процесі імітаційного моделювання, дозволяє виявити закономірності взаємодії запропонованого алгоритму з динамічним середовищем. Дослідження проведено на основі порівняння результатів у двох контрольних сценаріях: структурованому («Шосе–Болото») та хаотичному (сценарій з випадковим розподілом перешкод).

1. *Часові витрати на пошук шляху*

В обох сценаріях зафіксовано зростання часу обчислення маршруту при активації динамічного зважування. У сценарії «Шосе–Болото» середній час пошуку збільшився з 21,24 мс до 29,32 мс, що відповідає приросту на 38%. У хаотичному сценарії час обчислення зріс з 25,49 мс до 36,84 мс, тобто на 44%.

Максимальні часові витрати (до 54,15 мс у пікових значеннях) зафіксовано у хаотичному сценарії з увімкненим механізмом динамічного зважування. Це пояснюється відсутністю чітко виражених коридорів з

низькою щільністю перешкод, унаслідок чого ієрархічний планувальник змушений виконувати додаткові порівняння вартості переходів між сусідніми кластерами з близькими значеннями коефіцієнтів ризику. Водночас додаткові часові витрати в межах 8–11 мс є прийнятними з огляду на підвищення адаптивності та передбачуваності поведінки агента.

2. Безпека руху: кількість колізій

У сценарії «Шосе–Болото» механізм динамічного зважування продемонстрував найвищу ефективність. Середня кількість зіткнень зменшилася у 4,4 раза — з 213,1 до 48,3. У 30% контрольних запусків агент подолав усю дистанцію без жодного зіткнення, що свідчить про здатність моделі ефективно відокремлювати маршрут від зон з високою інтенсивністю руху.

У хаотичному сценарії значення показників для конфігурацій з увімкненим (174,5) та вимкненим (149,5) динамічним зважуванням виявилися близькими. Це зумовлено властивостями випадкового просторового розподілу перешкод, за якого алгоритм не має можливості сформувати маршрут із нульовим або близьким до нульового сумарним ризиком.

3. Стабільність траєкторії: кількість перепланувань

В обох сценаріях використання механізму динамічного зважування призвело до суттєвого зменшення кількості перепланувань. У структурованому сценарії їх кількість скоротилася на 68% (з 26,3 до 8,3), тоді як у хаотичному сценарії — на 47% (з 23,5 до 12,5).

За відсутності динамічного зважування агент обирає геометрично найкоротший маршрут, який часто перекривається рухомими об’єктами, що активує процедуру перевірки цілісності шляху (CheckPathValidity) та ініціює повторний пошук. За наявності динамічного зважування агент заздалегідь обирає ділянки з нижчим рівнем ризику, де ймовірність раптового блокування маршруту є істотно меншою.

4. Візуальний аналіз траєкторій руху

Окрім кількісних показників, виконано візуальний аналіз траєкторій руху агента, який наочно ілюструє вплив механізму динамічного зважування на геометрію маршрутів.

У сценарії «Шосе–Болото» маршрут без використання динамічного зважування проходить через центральні кластери з високою щільністю перешкод (25%) і характеризується значною кількістю локальних зламів, що є наслідком частих перепланувань. Натомість використання динамічного зважування призводить до формування альтернативного маршруту через кластери з низькою щільністю перешкод (близько 3%), що забезпечує більш плавну траєкторію, мінімальну кількість перепланувань і відсутність колізій.

У хаотичному сценарії траєкторії без динамічного зважування мають ламаний характер і часто проходять через області з високою локальною концентрацією перешкод. За умови використання механізму динамічного зважування алгоритм формує маршрут з мінімально можливим сумарним ризиком, навіть за відсутності повністю безпечних коридорів. Застосування алгоритму Theta* на нижньому рівні ієрархії забезпечує згладжування траєкторії та збереження динаміки руху при обході небезпечних ділянок.

Обговорення результатів дослідження

Результати проведених експериментальних досліджень підтверджують гіпотезу про те, що інтеграція механізму динамічного зважування ризиків у ієрархічну структуру планування шляху суттєво підвищує ефективність автономної навігації агентів у динамічних середовищах.

Ключовим аспектом обговорення є виявлений компроміс між обчислювальною складністю алгоритму та фізичною безпекою руху агентів. Хоча середній час безпосереднього пошуку маршруту зріс на 38–44%, ці витрати слід розглядати в контексті реальних ігрових і симуляційних систем. У межах даного дослідження агент представлено у вигляді спрощеної моделі, що рухається до цільової точки. Натомість у практичних застосуваннях кожен агент реалізує складну поведінкову логіку, яка включає анімаційні контролери, скрипти реакції на зіткнення та обробку фізичних взаємодій.

За використання класичних алгоритмів планування шляху без динамічного зважування ризиків кожна колізія ініціює виконання додаткових обчислювально витратних процедур, пов’язаних із відтворенням анімацій, перевіркою станів агентів і логікою ухилення. Сукупні витрати процесорного часу на обробку таких подій у реальних ігрових сценаріях значно перевищують кілька мілісекунд, необхідних для виконання більш інформованого стратегічного планування. Таким чином, запропонована модель із увімкненим механізмом динамічного зважування ризиків дозволяє оптимізувати загальний часовий бюджет кадру, зменшуючи кількість позаштатних ситуацій, пов’язаних із фізичними зіткненнями.

Отже, за результатами виконаної роботи можна сформулювати такі наукову новизну та практичну значущість результатів дослідження.

Наукова новизна отриманих результатів дослідження – отримала удосконалення ієрархічна модель планування шляху шляхом інтеграції функції динамічного зважування ризиків на рівні кластерів, що дозволяє враховувати щільність та інтенсивність руху перешкод ще до початку детальної генерації траєкторії.

У роботі вперше встановлено математичну залежність між показником локального ризику кластера та глобальною вартістю переходу в ієрархічному графі для алгоритмів класу Any-Angle (зокрема Theta*), що забезпечує побудову гладких і фізично безпечних маршрутів у динамічних середовищах.

Практична значущість результатів дослідження – визначається можливістю суттєвої оптимізації використання системних ресурсів при розробці складних симуляцій або ігрових проектів. Оскільки в реальних умовах кожен агент володіє набором складних скриптів поведінки та анімаційних контролерів, зниження

кількості колізій у 4.4 раза дозволяє уникнути надлишкових обчислювальних витрат на обробку фізичних зіткнень та реакцій персонажів, що значно економить бюджет часу кадру. Зменшення кількості репланувань на 68% забезпечує високу обчислювальну стабільність системи під навантаженням, а розроблений програмний модуль на мові C# для середовища Unity є універсальним рішенням, готовим до інтеграції в існуючі проекти для покращення навігації в умовах змінюваного середовища без необхідності перебудови NavMesh.

Висновки

У ході проведеного дослідження було розроблено та експериментально апробовано ієрархічну модель навігації автономних агентів у динамічному середовищі на основі поєднання алгоритму Theta* з механізмом динамічного зважування ризиків. На підставі отриманих результатів сформульовано такі висновки:

1. Встановлено, що введення вагових коефіцієнтів на рівні кластерів забезпечує предиктивне уникнення зон з високою щільністю динамічних перешкод. У структурованому сценарії «Шосе–Болото» це призвело до зменшення кількості колізій у 4,4 раза порівняно з класичними методами планування шляху.

2. Показано, що попри зростання часу одиничного пошуку маршруту на 38–44%, загальна обчислювальна стабільність системи суттєво підвищується завдяки скороченню кількості процедур перепланування на 68%, що є критично важливим для реальних інтерактивних систем.

3. Доведено, що використання алгоритму Theta* в ієрархічній структурі планування забезпечує формування плавних траєкторій під довільним кутом, що дозволяє компенсувати збільшення геометричної довжини маршруту під час обходу небезпечних зон.

4. Виявлено, що запропонований підхід демонструє найвищу ефективність у середовищах зі структурованими динамічними потоками. Реалізований програмний модуль у середовищі Unity може бути використаний для покращення навігації автономних агентів без необхідності застосування статичних навігаційних сіток.

Література

1. Hart P.E. A Formal Basis for the Heuristic Determination of Minimum Cost Paths / P.E. Hart, N.J. Nilsson, B. Raphael // *IEEE Transactions on Systems Science and Cybernetics*. – 1968. – Vol. 4, № 2. – P. 100–107. DOI: https://ieeexplore.ieee.org/document/4082128/

2. Botea A. Near Optimal Hierarchical Path-Finding (HPA*) / A. Botea, M. Müller, J. Schaeffer // *Journal of Game Development*. – 2004. – Vol. 1, № 1. – P. 7–28. Rezhym dostupu : https://www.researchgate.net/publication/228785110_Near_optimal_hierarchical_path-finding_HPA

3. Nash A. Theta*: Any-Angle Path Planning on Grids / A. Nash, K. Daniel, S. Koenig, A. Felner // *Proceedings of the AAAI Conference on Artificial Intelligence*. – 2007. – P. 1177–1183. DOI: https://doi.org/10.1613/jair.2994

4. Koenig S. D* Lite / S. Koenig, M. Likhachev // *Proceedings of the AAAI Conference on Artificial Intelligence*. – 2002. – P. 476–483.

5. Harabor D. Hierarchical Pathfinding for Multi-Agent Systems in Dynamic Environments / D. Harabor, A. Botea // *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. – 2008. – P. 1–8. DOI: https://doi.org/10.1109/CIG.2008.5035648

6. Van den Berg J. Reciprocal Collision Avoidance for Real-Time Multi-Agent Simulation (ORCA) / J. Van den Berg, M.C. Lin, D. Manocha // *IEEE Transactions on Robotics*. – 2011. – Vol. 27, № 1. – P. 111–122.

7. Ferguson D. Using Interpolation to Improve Path Planning: Field D* / D. Ferguson, A. Stentz // *Journal of Field Robotics*. – 2005. – Vol. 23, № 2. – P. 79–101. DOI: https://doi.org/10.1002/rob.20109

8. Khatib O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots / O. Khatib // *The International Journal of Robotics Research*. – 1986. – Vol. 5, № 1. – P. 90–98. DOI: https://doi.org/10.1177/027836498600500106

9. Sud A. Real-Time Path Planning in Dynamic Virtual Environments Using Multi-Agent Navigation Graphs / A. Sud, R. Gayle, E. Andersen, S. Guy, M. Lin, D. Manocha // *IEEE Transactions on Visualization and Computer Graphics*. – 2008. – Vol. 14, № 3. – P. 526–538. DOI: https://doi.org/10.1109/TVCG.2008.27

10. Sturtevant N.R. Memory-Efficient Abstractions for Pathfinding / N.R. Sturtevant // *Proceedings of the AAAI Conference on Artificial Intelligence*. – 2007. – P. 123–128. DOI: https://doi.org/10.1609/aiide.v3i1.18778