

<https://doi.org/10.31891/2307-5732-2026-365-43>

УДК 004:92

ЛИМАН ДМИТРО

Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”

<https://orcid.org/0009-0001-1300-580X>

e-mail: liman.dimitriy@gmail.com

МАРЧЕНКО ОЛЕКСАНДР

Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”

<https://orcid.org/0000-0002-4537-3420>

e-mail: oleksandr.i.marchenko@gmail.com

МЕТОД ЗМЕНШЕННЯ ГРАФА ВИДИМОСТІ ДЛЯ ОРТОГОНАЛЬНОЇ ВІЗУАЛІЗАЦІЇ РЕБЕР ГРАФА

Метод зменшення графа видимості за допомогою графової нейронної мережі для ортогонального зображення ребер графа. У статті розглядається актуальна задача прискорення побудови ортогонального зображення ребер графа, яка є важливою складовою автоматичного зображення графів у технічних діаграмах, таких як UML-діаграми, схеми бізнес-процесів та електричні схеми. Існуючі провідні методи ортогонального зображення ребер базуються на побудові повного ортогонального графа видимості з наступним пошуком оптимального шляху, що призводить до квадратичної залежності кількості ребер графа видимості від кількості вершин діаграми та обмежує можливість інтерактивної роботи з великими графами. Проведений аналіз останніх досліджень та публікацій показав, що графові нейронні мережі вже успішно застосовуються для зменшення часу роботи методів на графах у суміжних галузях, зокрема для зменшення простору пошуку в задачах комбінаторної оптимізації, навчання оцінювальних функцій для методів пошуку найкоротшого шляху та розрідження графів перед розв'язанням задачі комівояжера, проте ці підходи досі не були застосовані до задачі ортогонального зображення ребер, де специфічна геометрична структура графа видимості та багатокритеріальна функція вартості шляху вимагають спеціалізованого підходу. Метою роботи є розробка методу зменшення ортогонального графа видимості за допомогою графової нейронної мережі, який дозволив би зменшити кількість ребер графа видимості перед виконанням пошуку оптимального шляху, зберігаючи при цьому якість зображення ребер, порівнянню з результатами існуючих методів. Запропонований метод складається з чотирьох послідовних етапів: побудова графа-кандидата без перевірки фактичної видимості між вершинами, класифікація ребер попередньо навченою графовою нейронною мережею, відсіювання ребер з низькою ймовірністю входження до шуканого шляху та пошук шляху на зменшеному графі. Принциповою відмінністю від існуючих підходів є те, що розрідження виконується на етапі побудови графа видимості, а не на етапі пошуку шляху, що дозволяє зменшити час обробки на обох етапах. Описано процес формування навчального набору даних та навчання нейронної мережі, де результат роботи існуючого методу використовується як еталон для розпізнавання ребер, що є частиною шуканих шляхів. Резервний механізм гарантує, що у випадку невдалого розрідження результат роботи запропонованого методу ніколи не буде гіршим за результат існуючих методів.

Ключові слова: ортогональне зображення ребер, граф видимості, графова нейронна мережа, розрідження графа, пошук найкоротшого шляху.

LYMAN DMYTRO, MARCHENKO OLEKSANDR

National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”

GRAPH VISIBILITY REDUCTION METHOD FOR ORTHOGONAL VISUALIZATION OF GRAPH EDGES

A Method for Orthogonal Visibility Graph Reduction Using a Graph Neural Network for Orthogonal Edge Routing. The article addresses the current challenge of accelerating orthogonal edge routing, which is an essential component of automatic graph drawing in technical diagrams such as UML diagrams, business process schemas, and electrical circuit designs. Existing state-of-the-art orthogonal edge routing methods rely on constructing a full orthogonal visibility graph followed by an optimal path search, resulting in a quadratic dependency of the number of visibility graph edges on the number of diagram vertices and limiting the feasibility of interactive work with large graphs. An analysis of recent studies and publications has shown that graph neural networks have already been successfully applied to reduce the runtime of graph-based methods in related domains, including search space reduction in combinatorial optimization problems, learning heuristic functions for shortest path search methods, and graph sparsification prior to solving the travelling salesman problem; however, these approaches have not yet been applied to the orthogonal edge routing problem, where the specific geometric structure of the visibility graph and the multi-criteria path cost function require a specialized approach. The goal of this research is to develop a method for orthogonal visibility graph reduction using a graph neural network that would reduce the number of visibility graph edges before performing the optimal path search while maintaining edge routing quality comparable to existing methods. The proposed method consists of four sequential stages: constructing a candidate graph without verifying actual visibility between vertices, classifying edges using a pre-trained graph neural network, pruning edges with a low probability of belonging to the target path, and searching for a path on the reduced graph. The fundamental distinction from existing approaches is that sparsification is performed at the visibility graph construction stage rather than at the path search stage, which allows reducing processing time at both stages. The process of training data generation and neural network training is described, where the output of the existing method is used as a ground truth for recognizing edges that are part of the target paths. A fallback mechanism guarantees that in the case of unsuccessful sparsification, the result of the proposed method is never worse than that of existing methods.

Keywords: orthogonal edge routing, visibility graph, graph neural network, graph sparsification, shortest path search.

Стаття надійшла до редакції / Received 18.03.2026

Прийнята до друку / Accepted 16.04.2026

Опубліковано / Published 28.05.2026



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© Лиман Дмитро, Марченко Олександр

Постановка проблеми у загальному вигляді

та її зв'язок із важливими науковими чи практичними завданнями

Автоматичне зображення графів є однією з важливих задач в комп'ютерних науках, що знаходить

застосування у великій кількості інструментів для різних галузей науки і техніки: від редакторів UML-діаграм та BPMN-процесів до інструментів проєктування електричних схем, візуалізації мережевих топологій та біологічних шляхів метаболізму. Якість автоматичного зображення графа безпосередньо впливає на здатність користувача сприймати та аналізувати структуру даних, що робить цю задачу практично значущою для комп'ютерної інженерії та суміжних галузей.

Однією з ключових складових автоматичного зображення графів є визначення геометричної форми та координат ребер у вигляді ліній на площині. В ортогональному стилі зображення, який є стандартом для технічних діаграм, ребра складаються виключно з горизонтальних та вертикальних сегментів. Побудова таких ребер повинна задовольняти низку вимог одночасно: уникнення перетинів з вершинами-перешкодами, мінімізація кількості згинів, зменшення перетинів між ребрами та забезпечення достатньої відстані між паралельними сегментами.

Провідні методи розв'язання цієї задачі [1–3] базуються на побудові допоміжної графової структури — ортогонального графа видимості, з наступним пошуком оптимального шляху через нього. Основна обчислювальна проблема такого підходу полягає у квадратичному зростанні кількості ребер графа видимості відносно кількості точок з'єднання для кожної з вершин графа. При цьому оптимальний шлях між будь-якою парою точок використовує лише підмножину ребер, тобто переважна більшість побудованих ребер не входить до жодного оптимального шляху. Для великих графів з сотнями вершин побудова повного графа видимості та наступний пошук на ньому потребують значного часу обробки, що обмежує можливість інтерактивної роботи, коли час виконання є найбільш важливим. Існуючі методи оптимізації [3] зменшують простір пошуку, але не усувають цю фундаментальну проблему. Водночас, у суміжних галузях вже накопичено значний досвід застосування методів машинного навчання для пришвидшення комбінаторних методів на графах, проте ці напрацювання досі не були застосовані до задачі ортогонального зображення ребер.

Таким чином, задача зменшення ортогонального графа видимості за допомогою навченої моделі для пришвидшення побудови зображення ребер графа є актуальною і пов'язана як з дослідженнями методів візуалізації графів та машинного навчання, так і з практичними потребами авторів інтерактивних редакторів графів.

Аналіз досліджень та публікацій

Провідним підходом до ортогонального зображення ребер є метод, запропонований у роботах [1–3]. Цей метод складається з трьох етапів. На першому етапі виконується побудова ортогонального графа видимості, вершинами якого є кути перешкод та точки, з яких виходять ребра графа, а ребрами графа видимості — лінії ортогональної видимості між ними. На другому етапі виконується пошук оптимального шляху через побудований граф за допомогою модифікованого методу пошуку найкоротшого шляху. На третьому етапі формується кінцеве зображення ребра у вигляді послідовності горизонтальних та вертикальних сегментів.

У роботі [3] автори запропонували два способи оптимізації для зменшення простору пошуку. Перший спосіб оптимізації полягає у використанні графа видимості з одним згином — структури, що містить ребра лише між вершинами, які можуть бути з'єднані одним ортогональним згином. Другий спосіб оптимізації базується на спостереженні, що між двома точками часто існує багато різних шляхів з однаковою вартістю, які відрізняються лише відстанню до найближчої перешкоди. Замість дослідження всіх таких шляхів, у роботі розглядається лише один представник з кожної групи — шлях, що проходить якомога ближче до перешкод. Ці способи оптимізації дозволяють зменшити кількість досліджуваних шляхів, проте не усувають квадратичну залежність кількості ребер графа видимості від кількості вершин діаграми.

Паралельно з розвитком класичних детерміністичних методів, активно досліджується застосування графових нейронних мереж для зменшення часу роботи методів на графах. У роботі [8] автори запропонували метод, у якому графова нейронна мережа визначає, які вершини графа з високою ймовірністю входять до оптимального рішення, після чого класичний метод виконується лише на підграфі з цих вершин. За твердженням авторів, такий підхід дозволяє скоротити час роботи методу щонайменше вдвічі для задач пошуку найменшого вершинного покриття, максимальної незалежної множини та максимальної кліки. У роботі [9] досліджується вплив попереднього розрідження графа перед його обробкою нейронною мережею для розв'язання задачі комівояжера. Автори роботи [9] експериментально підтвердили, що зменшення покращує якість отриманого розв'язку та водночас зменшує час обробки.

Окремий напрям досліджень присвячений навчанню оцінювальних функцій для методів пошуку найкоротшого шляху. У роботі [5] автори тренують графову нейронну мережу одночасно на двох задачах: відтворення методу пошуку найкоротшого шляху та передбачення оцінювальної функції для цього методу. Таке поєднання дозволяє знаходити найкоротший шлях, досліджуючи при цьому меншу кількість вершин. У роботі [6] запропоновано метод навчання оцінювальних функцій графового пошуку на основі імітаційного навчання, де графова нейронна мережа тренується на траєкторіях пошуку та відстанях найкоротших шляхів. Навчена оцінювальна функція працює за сталий час, незалежний від розміру графа, і дозволяє зменшити кількість досліджених вершин у середньому на 58,5% порівняно з найкращими існуючими методами. У роботі [4] автори переформулювали метод пошуку найкоротшого шляху як диференційований модуль нейронної мережі, що дозволяє виконувати наскрізне навчання на прикладах оптимальних шляхів.

Також варто зазначити роботу [11], де автори будують спрощене представлення графа у вигляді остову, на якому тренується нейронна мережа для прискорення пошуку найкоротшого шляху у великих мережах. У роботі [10] навчена функція пріоритету відбирає лише десятку частину вершин і ребер на кожній ітерації пошуку,

що дозволяє застосовувати цей метод до графів з мільйонами вершин.

Проте жодна з розглянутих робіт не досліджує застосування нейронних мереж для розрідження графів у контексті ортогонального зображення ребер. Ортогональний граф видимості має специфічну геометричну структуру: його ребра є виключно горизонтальними або вертикальними, а вершини розташовані на перетинах прямих, що випускаються з кожної вершини вхідного графа. Крім того, вартість шляху у такому графі визначається не лише довжиною, а й кількістю згинів та перетинів з іншими ребрами. Ці особливості відрізняють задачу ортогонального зображення ребер від задач, розглянутих у роботах [4–11], і вимагають розробки спеціалізованого підходу.

Формулювання цілей статті

Метою роботи є: розробка методу зменшення ортогонального графа видимості за допомогою графової нейронної мережі, який дозволив би зменшити кількість ребер графа видимості перед виконанням пошуку оптимального шляху, зберігаючи при цьому якість зображення ребер, порівнянню з результатами існуючих методів. Як зазначалося вище, існуючі методи ортогонального зображення ребер графа базуються на побудові повного графа видимості, що призводить до квадратичної залежності часу обробки від кількості вершин діаграми. Існуючі оптимізації зменшують простір пошуку, але не усувають оригінальної квадратичної складності завдання, а напрацювання у галузі машинного навчання на графах досі не були застосовані до цієї задачі. Запропонований метод має враховувати специфічну геометричну структуру ортогонального графа видимості та функцію від багатьох критеріїв вартості шляху, що включає довжину, кількість згинів та кількість перетинів між ребрами.

На наступних етапах дослідження планується виконати експериментальне порівняння запропонованого методу з існуючими підходами на графах різного розміру з метою визначення умов, за яких метод забезпечує прискорення обробки. Розробка такого методу дозволила б, з однієї сторони, зменшити час побудови ортогонального зображення ребер для великих графів, а з другої сторони, визначити можливості та обмеження застосування графових нейронних мереж для прискорення класичних методів зображення графів.

Виклад основного матеріалу

Перед описом запропонованого методу уточнимо поняття, що використовуються надалі у роботі. Вхідний граф — це граф, для якого необхідно побудувати ортогональне зображення ребер. Його вершини вже мають визначені координати на площині та зображуються у вигляді прямокутників, наприклад, класи в UML-діаграмі або блоки у схемі бізнес-процесу. При побудові зображення ребер ці прямокутники виступають перешкодами, які ребра повинні обходити. Таким чином, кожна вершина вхідного графа є одночасно і перешкодою для побудови ребер.

Як зазначалося вище, існуючі методи ортогонального зображення ребер [1–3] будують повний граф видимості, що містить квадратичну кількість ребер відносно кількості вершин вхідного графа. Основна ідея запропонованого методу полягає у тому, щоб замість побудови повного графа видимості використати графову нейронну мережу для визначення, які ребра графа видимості з високою ймовірністю входять до шуканого шляху. Це дозволяє побудувати лише підмножину ребер графа видимості та виконати пошук шляху на меншому графі. Запропонований метод складається з чотирьох послідовних етапів: побудова графа-кандидата, класифікація ребер нейронною мережею, відсіювання ребер, які з низькою ймовірністю будуть входити до найкоротшого шляху, та пошук шляху на зменшеному графі. Нижче наведено детальний опис кожного кроку.

На першому кроці формується початковий граф-кандидат, що є меншим за повний граф видимості. Вершинами цього графа є ті ж самі точки, що й у графі видимості, описаному в роботах [1–3]: кути перешкод та точки, з яких виходять ребра вхідного графа. Для кожної вершини зберігаються її координати на площині, тип та розміри перешкоди, до якої вона належить.

Ребра графа-кандидата створюються лише між просторово близькими вершинами. Для цього можуть бути використані два підходи: з'єднання кожної вершини з фіксованою кількістю найближчих сусідів або з'єднання вершин, що лежать на спільній горизонталі чи вертикалі в межах обмеженої відстані. Для кожного ребра зберігаються різниця координат по осі абсцис та ординат, відстань ребра та його напрямок (горизонтальний або вертикальний).

Принциповою відмінністю від побудови повного графа видимості є те, що на цьому етапі не виконується перевірка фактичної видимості між вершинами, тобто не перевіряється, чи перетинає ребро якусь перешкоду. Завдяки цьому побудова графа-кандидата потребує значно менше часу, ніж побудова повного графа видимості.

На другому кроці попередньо навчена графова нейронна мережа обробляє граф-кандидат і для кожного його ребра визначає ймовірність того, що це ребро входиться до шуканого шляху. Процес навчання цієї мережі описано нижче, після опису усіх кроків методу. Процес обробки відбувається наступним чином. Спочатку кожна вершина отримує числове представлення на основі своїх ознак (координати, ознака кута перешкоди чи точки з'єднання ребра, розміри перешкоди). Далі виконується декілька ітерацій обміну інформацією між сусідніми вершинами, під час яких кожна вершина збагачує своє представлення даними про оточення — розташування та кількість сусідніх перешкод, наявність поблизу точок початку та кінця шуканого ребра. Після завершення ітерацій обміну для кожного ребра обчислюється оцінка ймовірності на основі представлень його двох кінцевих вершин та ознак самого ребра. Результатом роботи мережі є значення ймовірності від 0 до 1 для кожного ребра графа-кандидата.

На третьому кроці з графа-кандидата видаляються ребра, ймовірність яких є нижчою за задане порогове

значення. Вибір порогового значення визначає баланс між ступенем розрідження графа та ризиком втрати ребер, необхідних для побудови шуканого шляху. Занизьке порогове значення залишає у графі зайві ребра, але зменшує ризик втрати потрібних. Зависоке порогове значення дає більше розрідження, але збільшує ризик того, що шуканий шлях не зможе бути побудований через відсутність необхідних ребер.

Після відсіювання, для ребер, що залишилися, виконується перевірка фактичної видимості — чи не перетинає ребро хоч одну перешкоду. Оскільки кількість ребер після відсіювання є значно меншою за кількість ребер повного графа видимості, ця перевірка виконується швидко.

На четвертому кроці виконується пошук шляху між заданими точками на зменшеному графі видимості за допомогою того ж методу пошуку найкоротшого шляху, що використовується в існуючих роботах [1–3]. Оскільки зменшений граф містить значно менше ребер, пошук виконується швидше.

У випадку, коли пошук на зменшеному графі не знаходить жодного шляху між заданими точками, передбачено резервний механізм: виконується побудова повного графа видимості за існуючим методом [1–3] і пошук повторюється на повному графі. Такий підхід гарантує, що результат роботи запропонованого методу ніколи не буде гіршим за результат існуючого методу. У найгіршому випадку додатковий час обробки обмежується лише часом роботи нейронної мережі на другому кроці.

У другому кроці запропонованого методу, є посилання на навчену графову нейронну мережу. Далі описано процес навчання цієї мережі. Для початку, необхідно сформувати навчальний набір даних. Процес формування такого набору складається з наступних дій:

1. Генерація випадкового графа із заданою кількістю прямокутних перешкод та пар точок, між якими необхідно побудувати ребра.

2. Побудова повного графа видимості та пошук шляхів для всіх пар точок за допомогою існуючого методу [1–3].

3. Позначення кожного ребра графа видимості: ребро отримує мітку «1», якщо воно входить до будь-якого зі знайдених шляхів, та «0» в іншому випадку.

4. Збереження графа-кандидата разом з мітками ребер як одного прикладу навчального набору.

Ці дії повторюються для великої кількості випадкових графів різного розміру (кількість ребер та вершин) та складності (середній степінь вершин графа). Таким чином, результат роботи існуючого методу [1–3] використовується як еталон, на основі якого нейронна мережа навчається розпізнавати ребра, що є частиною шуканих шляхів.

Навчання нейронної мережі виконується шляхом мінімізації функції втрат, що визначає різницю між передбаченими мережею ймовірностями та фактичними мітками ребер. Оскільки помилкове відсіювання потрібного ребра (ребро позначено «1», а мережа передбачила низьку ймовірність) є значно небезпечнішим за помилкове збереження зайвого ребра (ребро позначено «0», а мережа передбачила високу ймовірність), функція втрат використовує різні ваги для цих двох типів помилок. Вага помилки відсіювання потрібного ребра встановлюється значно вищою, що спонукає мережу краще залишити зайве ребро, ніж пропустити потрібне.

Якість навченої моделі оцінюється за трьома показниками. Перший показник — частка правильно збережених потрібних ребер серед усіх потрібних ребер, яка має перевищувати 99%. Другий показник — частка відсіяних ребер серед усіх ребер графа-кандидата, що визначає ступінь розрідження. Третій показник — відношення вартості шляху, знайденого на зменшеному графі, до вартості шляху на повному графі видимості, яке має бути близьким до одиниці.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

Проведене дослідження спрямоване на розробку методу зменшення ортогонального графа видимості за допомогою графової нейронної мережі для прискорення побудови зображення ребер графа. Проведений аналіз існуючих робіт [4–11] показав, що графові нейронні мережі вже успішно застосовуються для зменшення часу роботи методів на графах у суміжних наукових роботах, зокрема для зменшення простору пошуку у задачах комбінаторної оптимізації [8], навчання оцінювальних функцій для методів пошуку найкоротшого шляху [5, 6] та розрідження графів перед розв'язанням задачі комівояжера [9]. Проте жодна з розглянутих робіт не досліджує застосування цих підходів до задачі ортогонального зображення ребер, де специфічна геометрична структура графа видимості та багатокритеріальна функція вартості шляху вимагають розробки спеціалізованого підходу.

Запропонований метод складається з чотирьох послідовних кроків: побудова графа-кандидата без перевірки фактичної видимості, класифікація ребер попередньо навченою графовою нейронною мережею, відсіювання ребер з низькою ймовірністю та пошук шляху на зменшеному графі. Принциповою відмінністю від існуючих підходів є те, що розрідження виконується на етапі побудови графа видимості, а не на етапі пошуку шляху, що дозволяє зменшити час обробки на обох етапах. Резервний механізм гарантує, що у випадку невдалого розрідження результат роботи методу ніколи не буде гіршим за результат існуючого методу [1–3].

На даному етапі, автори цієї роботи бачать три напрямки розвитку для цього дослідження. По-перше, це експериментальне порівняння запропонованого методу з існуючими підходами на графах різного розміру для визначення умов, за яких метод забезпечує пришвидшення обробки, та визначення розміру графа, починаючи з якого використання нейронної мережі стає доцільним. По-друге, дослідження можливості інкрементального оновлення результатів розрідження при поступових змінах діаграми (додавання або видалення окремих вершин чи ребер), що є важливим для інтерактивних редакторів, де користувач змінює діаграму поступово. По-третє,

дослідження впливу різних архітектур графових нейронних мереж та їх параметрів (кількість ітерацій обміну інформацією, розмір числового представлення вершин) на якість розв'язання та час роботи методу.

Література

1. Wybrow M. Orthogonal connector routing / M. Wybrow, K. Marriott, P.J. Stuckey // Graph Drawing. GD 2009. Lecture Notes in Computer Science. – Springer, 2010. – Vol. 5849. – P. 219–231.
2. Wybrow M. Incremental connector routing / M. Wybrow, K. Marriott, P.J. Stuckey // Graph Drawing. GD 2005. Lecture Notes in Computer Science. – Springer, 2006. – Vol. 3843. – P. 446–457.
3. Marriott K. Seeing Around Corners: Fast Orthogonal Connector Routing / K. Marriott, P.J. Stuckey, M. Wybrow // Diagrammatic Representation and Inference. Diagrams 2014. Lecture Notes in Computer Science. – Springer, 2014. – Vol. 8578. – P. 31–37.
4. Tian H. COMBHelper: A Neural Approach to Reduce Search Space for Graph Combinatorial Problems / H. Tian, S. Medya, W. Ye // Proceedings of the AAAI Conference on Artificial Intelligence. – 2024. – Vol. 38, No. 18. – DOI: 10.1609/aaai.v38i18.30070.
5. Hougardy S. On the Importance of Sparsification for Transformers and GNNs for TSP / S. Hougardy, M. Wilde // arXiv. – 2024. – arXiv:2403.17159.
6. Numeroso D. Learning heuristics for A* / D. Numeroso, D. Bacciu, P. Veličković // arXiv. – 2022. – arXiv:2204.08938.
7. Pándy M. Learning Graph Search Heuristics / M. Pándy, W. Qiu, G. Corso, P. Veličković, R. Ying, J. Leskovec, P. Liò // Proceedings of the First Learning on Graphs Conference. – 2022. – PMLR 198.
8. Liu T. Skeleton-Guided Learning for Shortest Path Search / T. Liu, X. Li, H. Li, H. Lu, C.S. Jensen, J. Xu // arXiv. – 2025. – arXiv:2508.02270.
9. Zhu Z. A*Net: A Scalable Path-based Reasoning Approach for Knowledge Graphs / Z. Zhu, X. Yuan, M. Galkin, L.-P. Xhonneux, M. Zhang, M. Gazeau, J. Tang // Advances in Neural Information Processing Systems (NeurIPS). – 2023.
10. Yonetani R. Path Planning using Neural A* Search / R. Yonetani, T. Taniai, M. Barekatin, M. Nishimura, A. Kanezaki // Proceedings of the 38th International Conference on Machine Learning. – 2021. – PMLR 139. – P. 12029–12039.

References

1. Wybrow M. Orthogonal connector routing / M. Wybrow, K. Marriott, P.J. Stuckey // Graph Drawing. GD 2009. Lecture Notes in Computer Science. – Springer, 2010. – Vol. 5849. – P. 219–231.
2. Wybrow M. Incremental connector routing / M. Wybrow, K. Marriott, P.J. Stuckey // Graph Drawing. GD 2005. Lecture Notes in Computer Science. – Springer, 2006. – Vol. 3843. – P. 446–457.
3. Marriott K. Seeing Around Corners: Fast Orthogonal Connector Routing / K. Marriott, P.J. Stuckey, M. Wybrow // Diagrammatic Representation and Inference. Diagrams 2014. Lecture Notes in Computer Science. – Springer, 2014. – Vol. 8578. – P. 31–37.
4. Tian H. COMBHelper: A Neural Approach to Reduce Search Space for Graph Combinatorial Problems / H. Tian, S. Medya, W. Ye // Proceedings of the AAAI Conference on Artificial Intelligence. – 2024. – Vol. 38, No. 18. – DOI: 10.1609/aaai.v38i18.30070.
5. Hougardy S. On the Importance of Sparsification for Transformers and GNNs for TSP / S. Hougardy, M. Wilde // arXiv. – 2024. – arXiv:2403.17159.
6. Numeroso D. Learning heuristics for A* / D. Numeroso, D. Bacciu, P. Veličković // arXiv. – 2022. – arXiv:2204.08938.
7. Pándy M. Learning Graph Search Heuristics / M. Pándy, W. Qiu, G. Corso, P. Veličković, R. Ying, J. Leskovec, P. Liò // Proceedings of the First Learning on Graphs Conference. – 2022. – PMLR 198.
8. Liu T. Skeleton-Guided Learning for Shortest Path Search / T. Liu, X. Li, H. Li, H. Lu, C.S. Jensen, J. Xu // arXiv. – 2025. – arXiv:2508.02270.
9. Zhu Z. A*Net: A Scalable Path-based Reasoning Approach for Knowledge Graphs / Z. Zhu, X. Yuan, M. Galkin, L.-P. Xhonneux, M. Zhang, M. Gazeau, J. Tang // Advances in Neural Information Processing Systems (NeurIPS). – 2023.
10. Yonetani R. Path Planning using Neural A* Search / R. Yonetani, T. Taniai, M. Barekatin, M. Nishimura, A. Kanezaki // Proceedings of the 38th International Conference on Machine Learning. – 2021. – PMLR 139. – P. 12029–12039.