

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н.Каразіна

Факультет математики і інформатики

Кафедра теоретичної та прикладної інформатики

Кваліфікаційна робота

бакалавр

на тему “Використання методів "Машинного навчання з підкріпленням" для пошуку оптимального маршруту в транспортній мережі.”

Виконала: студентка 4 курсу,

групи МФ-41

спеціальність 122

«Комп’ютерні науки»

освітньо-професійна програма «Інформатика»

Білецька А.Р.

Керівник

Панченко А.С.

Харків – 2024

Вступ.....	3
РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	6
1.1. Залізничне грузоперевезення: Недоліки та шляхи покращення.....	6
1.2. Аналіз підходів до розв’язання проблеми визначення оптимальних маршрутів.....	6
1.2.1. Найкоротший шлях.....	7
1.2.2. Підходи з урахуванням ЕТА.....	8
1.2.3. Підходи до роутингу.....	10
1.2.4. Недоліки зазначених підходів та шляхи до покращення.....	13
Постановка завдання.....	16
РОЗДІЛ 2. ВИБІР ОПТИМАЛЬНИХ ІНСТРУМЕНТІВ ДЛЯ РОЗВ’ЯЗАННЯ ПРОБЛЕМИ ДОСЛІДЖЕННЯ.....	17
2.1. Моделювання задачі за допомогою графових алгоритмів.....	17
2.2. Використання networkx та matplotlib для візуалізації графа.....	18
NetworkX.....	18
Matplotlib.....	19
РОЗДІЛ 3. ОПИС ПРИЙНЯТИХ РІШЕНЬ.....	19
Використання Q-навчання для оптимізації залізничного грузоперевезення.....	19
Опис методу Q-навчання.....	19
Модель Q-навчання для залізничного грузоперевезення.....	20
Дейкстра у якості бенчмарку.....	21
Використання Reinforcement Learning підходу.....	22
Порівняння результатів.....	23
ВИСНОВКИ.....	24
ДОДАТОК А.....	25
ДОДАТОК Б.....	29
ДОДАТОК В.....	29
ДОДАТОК Г.....	30
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	31

Вступ

Штучний інтелект (ШІ) швидко стає невід'ємною частиною повсякденного життя і знаходить широке застосування у багатьох галузях.

Розглянемо декілька причин, чому ШІ є актуальним зараз:

- **Збільшення обчислювальної потужності:** завдяки постійному розвитку обчислювальної техніки і технологій обробки даних, ми маємо можливість створювати складніші моделі ШІ, які можуть аналізувати великі обсяги інформації та навчатися на її основі.
- **Зростання доступності даних:** завдяки цифровим технологіям великі обсяги даних стають все більш доступними. Це дозволяє моделям ШІ навчатися на більш типових наборах даних і робити більш точні прогнози та рекомендації.
- **Потреба в автоматизації та оптимізації:** у бізнесі, науці, медицині та інших галузях є постійна потреба в автоматизації повсякденних процесів, вирішенні складних проблем та оптимізації рішень. ШІ може допомогти забезпечити ці цілі, шляхом впровадження автономних систем, аналізу даних та прийняття рішень на основі підказок ШІ.
- **Розвиток нових галузей:** ШІ відкриває нові можливості для розвитку галузей, які раніше могли бути недосяжними або обмеженими. Наприклад, в сфері медицини ШІ може допомогти у виявленні рідкісних захворювань або розробці нових методів лікування.
- **Конкурентні переваги:** Компанії, які успішно впроваджують та використовують ШІ, можуть отримати значні конкурентні переваги на ринку. ШІ дозволяє підвищити продуктивність, знизити витрати, поліпшити якість продукту та послуги, що робить їх привабливими для споживачів.

Проблеми впровадження ШІ на виробництві:

Штучний інтелект обіцяє трансформувати виробничі процеси та поліпшити ефективність бізнесу. Проте, разом із цим приходять і складнощі, особливо коли мова йде про стратегічне використання ШІ на виробництві. Тож розглянемо недоліки ШІ:

- **Відсутність контролю над процесом:** оскільки деякі системи ШІ можуть бути автономними або використовувати складні моделі машинного навчання, не завжди можна зрозуміти чому було прийнято дане рішення.
- **Непрозорість алгоритмів:** складність розуміння та інтерпретації алгоритмів, які використовуються в системах штучного інтелекту стає на заваді використання ШІ. Рішення, які надає ШІ, можуть бути неперевіреними або незрозумілими для фахівців у галузі. Це може призвести до недовіри до результатів та ускладнити прийняття стратегічних рішень.
- **Потреба в експертному використанні:** розгляд результатів, надані ШІ, потребує додаткового аналізу та інтерпретації від фахівців у відповідних галузях. Це може призвести до збільшення часу та ресурсів, необхідних для прийняття стратегічних рішень.
- **Ризик неправильного рішення:** наслідком невдалого розуміння результатів ШІ є можливість прийняття неправильних стратегічних рішень. Недоречне використання алгоритмів та неправильне їхнє інтерпретування можуть призвести до втрати часу, ресурсів та можливостей для підприємства.

Проблематика статичних алгоритмів з комплексними критеріями:

Прийняття рішень є невід'ємною частиною багатьох сфер діяльності, розуміння та вирішення проблем, пов'язаних з комплексними критеріями, стає нагальною задачею. Статичні алгоритми, які використовують фіксовані правила та параметри, можуть стикатися з рядом проблем, коли мають справу зі складними завданнями, що вимагають зважування різних критеріїв. Давайте розглянемо ці проблеми та можливі шляхи їх розв'язання.

- **Одноманітність алгоритмів:** статичні алгоритми мають тенденцію до застосування одного набору правил для всіх ситуацій, не враховуючи індивідуальних особливостей конкретної задачі або змін в середовищі.
- **Обмежена гнучкість:** статичні алгоритми зазвичай не мають можливості адаптуватися до змін у вхідних даних або умовах вирішення задачі. Вони фіксуються на певних правилах та параметрах, що може ускладнити їх застосування в змінних середовищах.
- **Відсутність самонавчання:** статичні алгоритми не мають здатності навчатися на основі власного досвіду та взаємодії з середовищем. Вони

залишаються незмінними навіть у випадку, коли отримують нову інформацію або стикаються з новими умовами.

Можливі Шляхи Розв'язання:

- **Використання машинного навчання з підкріпленням:** методи машинного навчання з підкріпленням, можуть навчатися на основі власного досвіду та адаптуватися до змін у вхідних даних та умовах.
- **Розробка гнучких алгоритмів:** алгоритми, які можуть динамічно адаптуватися до змін у вхідних умовах та вимогах завдання, є ключовим для вирішення проблем з комплексними критеріями.
- **Врахування взаємозв'язків між критеріями:** алгоритми, які можуть враховувати не тільки окремі критерії, але й їх взаємозв'язки та важливість, допоможе покращити якість рішень.

Мета:

Метою даної роботи є розробка та порівняння двох механізмів керування рухом вантажних вагонів у мережі перевезень: перший механізм базується на використанні алгоритму Дейкстри, а другий - на застосуванні підходів з reinforcement learning (RL).

Актуальність теми:

Швидке та ефективне управління рухом товарних вагонів має велике значення для підтримання ефективної роботи транспортної системи. Розвиток нових підходів, таких як RL, може сприяти покращенню продуктивності та зниженню затримок у русі вагонів.

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Залізничне грузоперевезення: Недоліки та шляхи покращення

Залізничне грузоперевезення включає в себе ряд складних процесів, які є важкими в управлінні навіть у відносно стабільних умовах. Давайте розглянемо принципи процесу руху товарних потягів по залізничній мережі:

- **Планування маршруту та призначення шляхів:** При завантаженні товарів на вагони, планується маршрут, який буде найбільш ефективним відстані. Якщо найкоротший шлях вільний, то товарний потяг може бути направлений цим маршрутом.
- **Пріоритети в русі:** Пасажирські потяги мають пріоритет у русі перед товарними у випадку переповненості мережі.

Планування маршрутів та призначення шляхів є ключовою складовою цієї системи. Однак, цей підхід має свої недоліки.

Перш за все, недостатня гнучкість системи може призвести до проблем при адаптації до змінних умов. Наприклад, неможливість швидко перепрограмувати маршрути під час непередбачених обставин може спричинити затримки та інші проблеми в доставці товарів. Крім того, постійне використання найкоротшого шляху може призвести до його швидкого зношення, що вимагатиме частіше обслуговування та ремонту.

Також, неефективне використання інфраструктури може бути серйозною проблемою. Чекаючи на вільний найкоротший шлях, інші шляхи можуть залишатися порожніми, що призведе до не оптимізованого використання ресурсів та збільшення витрат на утримання.

1.2. Аналіз підходів до розв'язання проблеми визначення оптимальних маршрутів

Визначення оптимального маршруту

В рамках цього завдання оптимальним будемо вважати маршрут, який мінімізує загальний час, необхідний для досягнення кінцевої точки,

враховуючи всі поточні та змінні умови. Також зменшує трафік та уникає затори.

1.2.1. Найкоротший шлях

Для оптимізації грузоперевезення за допомогою алгоритмів пошуку найкоротшого шляху можна використати такі підходи:

1. Адаптація алгоритму Дейкстри:

Адаптувавши алгоритм Дейкстри до задачі грузоперевезень, де вагою ребер буде пропускна спроможність та довжина між станціями, а вершинами будуть різні станції чи пункти завантаження/розвантаження. Буде забезпечено знаходження найкоротшого шляху між станціями та подальша відправка точки до кінцевої вершини.

Основні кроки робочого процесу:

1. Ініціалізація: початкова вершина визначається як початок маршруту, а всі інші вершини помічаються як недосяжні з початкової вершини. Також ініціалізуються ваги ребер із початкової вершини до всіх інших вершин.
2. Обхід вершин: починаючи з початкової вершини, алгоритм Дейкстри переходить до сусідніх вершин, оновлюючи їх ваги відповідно до найкоротшого відомого шляху.
3. Вибір наступної вершини: знаходиться вершина з найменшою вагою, яка ще не була оброблена, і вона визначається як наступна поточна вершина.
4. Оновлення ваг вершин: для кожної сусідньої вершини, алгоритм перевіряє, чи можна скоротити шлях до неї, проходячи через поточну вершину. Якщо так, то вага цієї вершини оновлюється.
5. Повторення: ці кроки повторюються, поки всі вершини не будуть оброблені або доки не буде знайдений найкоротший шлях до кінцевої вершини.
6. Прийняття рішень: після обраного маршруту точка вирушає по ньому, враховуючи поточні ваги ребер. При потребі чи недостатній кількості ресурсів, такої, як пропускна спроможність точка очікує на відправку в останній досяжній вершині.

2. Використання алгоритму A* для маршрутизації грузів:

Алгоритм A* працює, обираючи вершини з найменшою сумою ваги і евристичної оцінки, оновлюючи ваги сусідніх вершин та продовжуючи пошук, доки не буде знайдений найкоротший маршрут до кінцевої вершини або не буде досягнуто максимальну кількість ітерацій.

Основні кроки робочого процесу:

1. Ініціалізація: початкова вершина визначається як початок маршруту, а всі інші вершини помічаються як недосяжні з початкової вершини. Також ініціалізуються евристичні оцінки для кожної вершини.
2. Обхід вершин: починаючи з початкової вершини, алгоритм A* переходить до сусідніх вершин, обираючи ту, яка має найменшу суму ваги і евристичної оцінки.
3. Вибір наступної вершини: обирається вершина з найменшою сумою ваги і евристичної оцінки.
4. Оновлення ваг вершин: вага вершини оновлюється, якщо виявляється, що можна скоротити шлях до неї, проходячи через поточну вершину.
5. Повторення: ці кроки повторюються до досягнення кінцевої вершини або знаходження оптимального маршруту.
6. Прийняття рішень: після обраного маршруту точка вирушає по ньому, враховуючи поточні ваги ребер. При потребі чи недостатній кількості ресурсів, такої, як пропускна спроможність точка очікує на відправку в останній досяжній вершині.

1.2.2. Підходи з урахуванням ЕТА

При розв'язанні проблеми перевезення грузу з використанням підходів з урахуванням ЕТА(estimated time of arrival) розглядаються методи, що базуються на передбаченні часу прибуття товарного вагону до пункту призначення.

1. Статистичні методи прогнозування:

Ці методи використовують статистичні моделі для передбачення часу прибуття грузоперевезень на основі історичних даних, таких як швидкість пересування транспортних засобів, маршрути, часи доставки тощо.

Основні кроки робочого процесу:

1. Обробка даних: збираються і аналізуються дані про маршрути, трафік, погодні умови, часи доставки та інші фактори, які можуть впливати на час прибуття грузу.
2. Побудова моделі: на основі оброблених даних розробляється статистична модель, яка передбачає час прибуття грузу. Ця модель може використовувати різні статистичні методи, такі як регресійний аналіз, часові ряди або методи машинного навчання, що базуються на статистичних принципах.
3. Тестування та валідація: після побудови моделі вона тестується на тестових даних для оцінки її точності та ефективності. Використання різних метрик, таких як середньоквадратична помилка чи коефіцієнт детермінації, дозволяє оцінити якість моделі.
4. Оновлення моделі: При необхідності модель може бути вдосконалена за рахунок внесення змін у вихідні дані або модифікації самої моделі для покращення прогнозування часу прибуття грузу.

2. Машинне навчання:

Методи машинного навчання, такі як нейронні мережі, дерева рішень, навчання з підкріпленням, можуть бути використані для прогнозування часу прибуття грузоперевезень на основі великої кількості даних.

Основні кроки робочого процесу можуть включати:

1. Обробка та підготовка даних: підготовка великих обсягів даних для навчання моделі, включаючи їх очищення, вибір функцій та нормалізацію.
2. Вибір моделі: обираються відповідні моделі машинного навчання, такі як нейронні мережі, дерева рішень, навчання з підкріпленням, залежно від характеристик даних та поставленої задачі.

3. Тренування моделі: модель навчається на тренувальних даних для виявлення залежностей між вхідними змінними та цільовими значеннями, такими як час прибуття грузу.
4. Тестування та валідація: після тренування модель перевіряється на тестових даних для оцінки її точності та генералізації на нові дані.
5. Оптимізація та налаштування: модель може бути оптимізована шляхом вибору оптимальних параметрів або застосування інших методів для підвищення її ефективності.

3. Оптимізаційні алгоритми:

Опис: Ці методи використовують математичні моделі для оптимізації маршрутів з урахуванням часу прибуття. Наприклад, методи лінійного програмування, динамічного програмування тощо.

Основні кроки робочого процесу можуть включати:

1. Постановка задачі: Описується математична модель задачі, включаючи обмеження та цільову функцію, яка враховує час прибуття грузу.
2. Вибір алгоритму: Обирається відповідний оптимізаційний алгоритм, такий як лінійне або нелінійне програмування, динамічне програмування, алгоритми градієнтного спуску тощо, в залежності від характеристик задачі.
3. Розв'язання задачі: Виконується розв'язання задачі оптимізації для знаходження оптимального рішення, яке враховує час прибуття грузу.
4. Тестування та оптимізація: Отримане рішення перевіряється на тестових даних, і в разі потреби алгоритм може бути оптимізований для підвищення його ефективності та швидкості розв'язання.

1.2.3. Підходи до роутингу

1. Методи динамічного програмування:

Методи динамічного програмування працюють шляхом розбиття складної задачі на менші підзадачі, що їх можна розв'язати окремо. Розв'язок кожної

підзадачі зберігається, щоб уникнути повторних обчислень, і використовується для отримання розв'язку більшої задачі.

Основні кроки робочого процесу можуть включати:

1. Розбиття задачі: Спочатку складна задача визначення оптимальних маршрутів перевезення вантажу розбивається на менші підзадачі. Використовуючи алгоритм Флойда-Уоршелла можна провести розрахунок найкоротших шляхів між усіма парами вузлів у мережі.
2. Збереження проміжних результатів: Розв'язок кожної підзадачі зберігається для подальшого використання, що дозволяє уникнути повторних обчислень та оптимізувати час роботи алгоритму.
3. Оновлення таблиць або матриць: Алгоритм починає ітеративно оновлювати таблиці або матриці, де кожен елемент представляє собою певну характеристику (наприклад, відстань або час перевезення) між парами вершин. На кожній ітерації алгоритм розглядає всі можливі шляхи через інші вершини та оновлює значення у відповідних елементах таблиць.
4. Пошук оптимального рішення: Процес оновлення таблиць або матриць продовжується до тих пір, поки не будуть знайдені найкоротші шляхи між усіма парами вершин. Після завершення алгоритму отримується оптимальне рішення задачі перевезення вантажу, яке враховує найкоротші шляхи між усіма пунктами в мережі вантажоперевезень.

2. Метаевристичні методи:

Метаевристичні методи вирішують оптимізаційні проблеми, використовуючи евристичні стратегії для наближеного знаходження рішень. Вони оперують на принципі пошуку у просторі можливих рішень з метою знаходження найкращого або прийняттого рішення.

Наприклад, метод симуляційного огорожі може бути використаний для оптимізації маршрутів вантажоперевезень у нашій системі. Цей метод імітує реальний рух поїздів у реальному часі, а потім виконує ітеративні покращення маршрутів на основі зібраної інформації про рух.

Основні кроки робочого процесу можуть включати:

1. Початковий розрахунок маршрутів: Спочатку виконується пошук початкових маршрутів для кожного вантажу на основі доступної інформації про розклади руху поїздів і обмежень мережі.
2. Симуляція руху поїздів: За допомогою методу симуляційного огорожі робиться спроба відтворити реальний рух поїздів у мережі, враховуючи різні фактори, такі як потоки вантажів, обмеження мережі, швидкість поїздів тощо.
3. Оптимізація маршрутів: На основі результатів симуляції алгоритм аналізує робочі маршрути та вносить зміни для поліпшення ефективності. Це може включати зміни в розкладах руху поїздів, вибір альтернативних маршрутів або реорганізацію розподілу вантажів.

3. Інтерактивні методи маршрутизації:

Інтерактивні методи маршрутизації оперують у реальному часі, використовуючи актуальні дані для прийняття рішень щодо маршрутів. Вони зазвичай враховують затори, погодні умови, обмеження мережі та інші фактори, які можуть впливати на ефективність маршрутів.

Наприклад, у системі грузоперевезень використання GPS-технологій та даних про рух поїздів у реальному часі дозволяє інтерактивно адаптувати маршрути відповідно до змін у русі та умовах на маршруті.

Основні кроки робочого процесу можуть включати:

1. Збір даних: Система збирає дані про рух поїздів, стан мережі, обмеження, а також актуальні погодні умови.
2. Аналіз у реальному часі: За допомогою отриманих даних система оцінює поточну ситуацію на маршрутах та ідентифікує можливі проблеми, такі як затори або небезпечні умови.
3. Прийняття рішень: На основі аналізу система приймає рішення щодо оптимальних маршрутів для поїздів, враховуючи актуальні обмеження та умови на шляху.

4. Оновлення маршрутів: У режимі реального часу система оновлює маршрути відповідно до змін у ситуації на маршруті, забезпечуючи ефективну доставку вантажів та уникнення затримок.

1.2.4. Недоліки зазначених підходів та шляхи до покращення

Адаптація алгоритму Дейкстри:

- Недоліки:
 - Обмежена функціональність: Алгоритм Дейкстри працює лише з неорієнтованими графами та не може обробляти негативні ваги ребер.
 - Швидкість: У великих графах з великою кількістю вершин та ребер алгоритм може стати неефективним через необхідність перевірки всіх ребер.
- Шляхи до покращення:
 - Використання покращених структур даних: Застосування пріоритетних черг або бінарних куп, що може полегшити пошук найменшої вершини та знизити час роботи алгоритму.
 - Розробка варіацій: Розгляд можливості адаптації алгоритму Дейкстри для роботи з орієнтованими графами або додавання підтримки негативних ваг ребер.

Використання алгоритму A* для маршрутизації грузів:

- Недоліки:
 - Велика обчислювальна складність: Алгоритм A* може вимагати значних обчислювальних ресурсів, особливо у великих графах або при складних евристичних оцінках.
 - Необхідність відбору правильних евристик: Недостатньо точна або неправильно обрана евристика може призвести до неоптимальних маршрутів.
- Шляхи до покращення:
 - Оптимізація евристик: Покращення точності та ефективності евристик може значно покращити результати роботи алгоритму.

- Розробка паралельних реалізацій: Використання паралельних обчислень для розділення завдань та прискорення роботи алгоритму.

Статистичні методи прогнозування:

- Недоліки:
 - Залежність від якості даних: Точність передбачень статистичних моделей може значно залежати від якості та обсягу історичних даних.
 - Стійкість до змін: Моделі можуть бути нестійкими до змін в умовах, таких як зміни маршрутів або погодних умов.
- Шляхи до покращення:
 - Використання адаптивних моделей: Розробка моделей, які можуть адаптуватися до змінних умов та автоматично оновлюватися на основі нових даних.
 - Використання ансамблевих методів: Поєднання декількох моделей для отримання більш точних передбачень та зменшення чутливості до шуму в даних.

Машинне навчання:

- Недоліки:
 - Потреба в великій кількості даних: Деякі методи машинного навчання, особливо глибоке навчання, можуть вимагати значної кількості даних для ефективного навчання.
 - Інтерпретованість: Деякі моделі, такі як нейронні мережі, можуть бути складними для інтерпретації та розуміння причин передбачень.
- Шляхи до покращення:
 - Використання переднього аналізу даних: Аналіз структури та властивостей даних перед вибором моделей може допомогти у виборі найбільш підходящих методів.
 - Застосування технік зменшення розмірності: Використання технік зменшення розмірності даних, таких як відбір ознак або зменшення розмірності, може покращити роботу моделей та зменшити вимоги до обсягу даних.

Оптимізаційні алгоритми:

- Недоліки:
 - Час виконання: Деякі оптимізаційні алгоритми можуть вимагати значного часу для пошуку оптимального рішення, особливо в складних задачах.
 - Чутливість до початкових умов: Деякі алгоритми можуть бути чутливими до початкових умов або параметрів, що може вплинути на стабільність та точність рішень.
- Шляхи до покращення:
 - Використання паралельних обчислень: Використання паралельних обчислень для прискорення виконання алгоритмів та обробки великих обсягів даних.
 - Оптимізація параметрів: Автоматична або напівавтоматична настройка параметрів алгоритмів для досягнення кращої продуктивності та ефективності.

Методи динамічного програмування:

- *Недоліки:*
 - Велика обчислювальна складність: Деякі задачі можуть мати велику обчислювальну складність через велику кількість підзадач або велику кількість даних.
 - Чутливість до розміру задачі: При збільшенні розміру задачі може зростати час обчислень, що ускладнює розв'язання.
- *Шляхи до покращення:*
 - Використання апроксимаційних методів: Застосування апроксимаційних методів для скорочення обчислювального часу при збереженні прийнятної точності рішення.
 - Паралельні обчислення: Використання паралельних обчислень для розподілу обчислювального навантаження та прискорення розв'язання задачі.

Метаевристичні методи:

- *Недоліки:*
 - Ризик застрягання в локальних мінімумах або максимумах: Метаевристичні методи можуть не забезпечити гарантовано

оптимального рішення, оскільки можуть застрягати в локальних мінімумах або максимумах.

- Час виконання: Деякі метаевристичні методи можуть вимагати значного часу для знаходження рішення, особливо в складних задачах.
- *Шляхи до покращення:*
 - Використання гібридних методів: Комбінація різних метаевристичних методів або їх поєднання з іншими методами для покращення ефективності та зниження ризику застрягання в локальних мінімумах або максимумах.
 - Оптимізація параметрів: Налаштування параметрів метаевристичних методів для забезпечення кращої збіжності та швидкості знаходження рішення.

Постановка завдання

Формалізація проблеми:

Для обох механізмів руху вагончиків мережа перевезень представляє собою граф, де вершини - це різні пункти призначення, а ребра - шляхи між цими пунктами. Кожному ребру призначається вага, що включає в себе пропускну спроможність та відстань між точками.

Обмеження:

1. Генерація:
 - Система повинна кожні 5 секунд генеруватися нові вагони
 - Початкова та кінцева точки маршруту вагончика повинні бути обрані випадковим чином з мережі перевезень.
2. Рух:
 - Вантажні вагони мають рухатися від початкової точки до кінцевої точки за допомогою алгоритму Дейкстри та на застосуванні підходів з reinforcement learning.
 - При русі враховується пропускну спроможність доріг та відстань між точками.
 - Початковий колір вагона, який рухається по ребру, є зеленим.

- Якщо пропускна спроможність ребра, по якому рухається вантажний вагон, дорівнює нулю, то вагон змінює свій колір на червоний та стає в чергу.
3. Керування швидкістю:
- Кожний вантажний вагон повинен мати випадково задану швидкість, що враховується при русі по мережі.
4. Кінець гри:
- Точка досягла кінцевої вершини шляху.
5. Визначення винагорода:
- Якщо точка змогла виконати переміщення, винагорода збільшується на 10.
 - Якщо точка перебуває у черзі для руху, винагорода зменшується на 5.
 - Якщо шлях точки складається з більшої кількості вершин, ніж загальна кількість вершин, винагорода зменшується на 10.
 - Якщо точка досягла кінцевої вершини, винагорода збільшується на 100.
 - Якщо в шляху точки виявлено дублікати, винагорода зменшується на 50.

Результат:

Результатом роботи буде порівняльний аналіз двох підходів до керування рухом вагонів у мережі перевезень. Очікується, що результати дослідження дадуть змогу визначити ефективність та переваги кожного з механізмів у вирішенні поставленої проблеми.

РОЗДІЛ 2. ВИБІР ОПТИМАЛЬНИХ ІНСТРУМЕНТІВ ДЛЯ РОЗВ'ЯЗАННЯ ПРОБЛЕМИ ДОСЛІДЖЕННЯ

2.1. Моделювання задачі за допомогою графових алгоритмів

У вантажних перевезеннях важливо ефективно планувати маршрути та розподіл вантажів між ними з метою мінімізації витрат часу та ресурсів. Можна моделювати цю проблему як графовий алгоритм, де вершини

представляють пункти завантаження та розвантаження, а ребра - шляхи між ними.

Формалізація:

Нехай $G = (V, E)$ - розріджений граф, де V - множина вершин, E - множина ребер.

Кожна вершина $v \in V$ відповідає пункту завантаження або розвантаження.

Кожне ребро $(u, v) \in E$ відповідає шляху між пунктами завантаження/розвантаження.

Вага w_{uv} кожного ребра визначає вартість перевезення вантажу по цьому шляху.

Задача полягає у знаходженні найкоротшого шляху від початкової вершини s до кінцевої вершини t у графі G .

2.2. Використання networkx та matplotlib для візуалізації графа

NetworkX

NetworkX - це пакет для Python, який призначений для створення, маніпулювання та вивчення структури, динаміки та функцій складних мереж.

Основні можливості NetworkX включають:

- Створення, зберігання та обробка графів, орієнтованих графів та мультиграфів.
- Велика кількість стандартних алгоритмів для роботи з графами.
- Визначення та аналіз структур та мережевих мір.
- Генератори для створення класичних графів, випадкових графів та синтетичних мереж.

- Гнучкість у взаємодії з вузлами та ребрами полягає в тому, що вузли можуть представляти будь-який тип даних, а ребра можуть містити різноманітні дані, включаючи, але не обмежуючись вагами та часовими рядами.

Matplotlib

Matplotlib - це комплексна бібліотека для створення статичних, анімованих та інтерактивних візуалізацій в Python. Matplotlib робить прості речі легкими і складні речі можливими.

Основні можливості Matplotlib включають:

- Створення графіків високої якості для публікацій.
- Створення інтерактивних фігур з можливістю масштабування, панорамування та оновлення.
- Налаштування візуального стилю та макету.
- Експорт до багатьох форматів файлів.
- Вбудовання в JupyterLab та графічні користувацькі інтерфейси.
- Використання широкого спектра сторонніх пакетів, побудованих на основі Matplotlib.

РОЗДІЛ 3. ОПИС ПРИЙНЯТИХ РІШЕНЬ

Використання Q-навчання для оптимізації залізничного грузоперевезення

У цьому розділі ми розглянемо використання методу навчання з підкріпленням, зокрема Q-навчання, для розв'язання проблеми оптимізації залізничного грузоперевезення.

Опис методу Q-навчання

Q-навчання є ключовим методом в галузі машинного навчання, який використовується для навчання агентів приймати рішення в невизначеному середовищі з метою максимізації нагороди. Цей метод базується на ідеї оцінки цінності дій в кожному стані середовища і визначає, яка дія є найкращою в кожному стані з урахуванням майбутніх нагород.

Основною концепцією за Q-навчання є Q-функція, яка визначає очікувану суму нагород, яку може отримати агент, вибравши певну дію у певному стані та діючи оптимально після цього. Формально, Q-функція визначається як:

$$Q(s, a) = E [R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | s, a]$$

Де s - стан середовища, a - дія, R_t - нагорода у момент часу t , γ - дисконтний коефіцієнт, який визначає вагомість майбутніх нагород. Очікування береться з урахуванням всіх майбутніх кроків.

Під час навчання агент намагається максимізувати значення Q-функції, щоб здійснювати оптимальні дії в кожному стані. Для цього використовується процес навчання, який включає в себе оновлення значень Q-функції на кожному кроці взаємодії з середовищем на основі отриманих нагород і нових станів.

Цей алгоритм використовує методи пониження помилки для оновлення значень Q-функції на кожному кроці, щоб навчити агента приймати оптимальні рішення. Під час навчання агент взаємодіє з середовищем, вибираючи дії і спостерігаючи нагороди, і оновлює значення Q-функції згідно з правилами Q-learning.

Модель Q-навчання для залізничного грузоперевезення

Для реалізації Q-навчання можна використати нейронну мережу, яка буде навчатися оцінювати цінність різних дій в різних станах системи залізничного грузоперевезення. У нашій моделі ми використовуємо лінійну нейронну мережу з декількома шарами для апроксимації Q-функції.

```
class Linear_QNet(nn.Module):  
  
    def __init__(self, input_size, hidden_size, output_size):  
  
        super().__init__()  
  
        self.linear1 = nn.Linear(input_size, hidden_size)
```

```

self.linear2 = nn.Linear(hidden_size, output_size)

def forward(self, x):

    x = F.relu(self.linear1(x))

    x = self.linear2(x)

    return x

class QTrainer:

    def __init__(self, model, lr, gamma):

        self.lr = lr

        self.gamma = gamma

        self.model = model

        self.optimizer = optim.Adam(model.parameters(), lr=self.lr)

        self.criterion = nn.MSELoss()

```

Дейкстра у якості бенчмарку

В епоху, коли штучний інтелект та машинне навчання стають все більш популярними для вирішення складних задач, класичні алгоритми, такі як алгоритм Дейкстри, залишаються важливим інструментом та відправною точкою для порівняння нових методів.

Побудова маршруту:

Дейкстра завжди знаходить найкоротший шлях, що дозволяє порівнювати результати нових алгоритмів з найкращим можливим рішенням.

Детермінованість:

Алгоритм завжди повертає однаковий результат для одних і тих же вхідних даних, що забезпечує консистентність тестів.

Простота реалізації:

Алгоритм Дейкстри відносно простий у реалізації та розумінні, що дозволяє швидко налаштувати тестове середовище.

Використання Reinforcement Learning підходу

Використання підходу з Reinforcement Learning для планування маршрутів у вантажних перевезеннях може мати значний вплив на транспортну систему. Хоча цей підхід може не завжди забезпечувати найкоротший шлях, він може призвести до ряду переваг:

Навантаження на дороги:

Однією з переваг використання підходу з RL є те, що він може допомогти розподілити навантаження на дорогах більш рівномірно. Оскільки RL може враховувати стан доріг та уникати перевантаження певних маршрутів, це дозволяє зменшити зношування доріг та покращити загальний стан транспортної інфраструктури.

Прогнозування часу доставки:

Ще однією важливою перевагою є можливість прогнозування часу доставки товарів. Оскільки агенти в RL можуть взаємодіяти з довгостроковими та короткостроковими факторами, такими як трафік, попит на перевезення та інші, вони можуть навчитися прогнозувати час доставки з великою точністю. Це дозволяє зменшити час очікування клієнтів та покращити задоволеність клієнтів від обслуговування.

Адаптивність до змін:

Оскільки RL агенти можуть навчитися адаптуватися до змінних умов, таких як зміна маршрутів, об'ємів перевезень, або навіть стану доріг, вони можуть ефективно працювати в реальному часі і вирішувати нагальні проблеми, такі як тимчасові перешкоди на дорозі або зміни в транспортних потоках.

Порівняння результатів

У сучасних задачах оптимізації маршрутів алгоритм Дейкстри є класичним підходом для знаходження найкоротших шляхів у графах. З іншого боку, методи Reinforcement Learning (RL) пропонують адаптивні стратегії, здатні враховувати динамічні фактори та довгострокові наслідки. На прикладі даних з **ДОДАТКУ А** розглянемо ефективність використання алгоритму Дейкстри та RL агентів для планування маршрутів.

Алгоритм Дейкстри демонструє менший час виконання в випадках, коли кількість вагонів є невеликою, у порівнянні з RL агентами. Але зі збільшенням кількості вагонів, час необхідний для проходження маршруту стає більшим через очікування на вершинах. Це пов'язано з тим, що в алгоритмі Дейкстри кожен вагон проходить по одному і тому ж маршруту, що може створювати затори на певних ділянках. RL агенти, в свою чергу, можуть враховувати цю особливість і розподіляти вагони по різних маршрутах, що зменшує загальний час проходження.

Переваги алгоритму Дейкстри

- Простота реалізації та використання.

Переваги Reinforcement Learning

- Здатність адаптуватися до змінних умов у реальному часі.
- Розподілення ресурсів для уникнення заторів.
- Можливість прогнозувати час доставки.

Недоліки алгоритму Дейкстри

- Не враховує динамічні зміни у середовищі.
- Відсутня можливість прогнозувати час доставки.

Недоліки Reinforcement Learning

- Складність навчання та потреба у великій кількості даних для ефективного навчання.
- Необхідність більш точного налаштування параметрів моделі.

ВИСНОВКИ

В рамках виконання даної дипломної роботи було досягнуто таких результатів:

- Проведено детальний аналіз залізничного грузоперевезення, виявлено основні недоліки та визначено шляхи їх покращення.
- Вивчено та проаналізовано сучасні підходи до розв'язання проблеми визначення оптимальних маршрутів, включаючи найкоротші шляхи, методи з урахуванням часу прибуття (ETA), та підходи до роутингу.
- Виконано моделювання задачі за допомогою графових алгоритмів.
- Використано бібліотеки NetworkX та Matplotlib для візуалізації графа, що дозволило наочно продемонструвати маршрути та їх оптимізацію.
- Використано метод Q-навчання для оптимізації залізничного грузоперевезення.
- Розроблено модель Q-навчання, яка дозволяє адаптувати маршрути в реальному часі, враховуючи змінні умови та підвищуючи ефективність системи.
- Алгоритм Дейкстри було використано як бенчмарк для порівняння з підходами Reinforcement Learning.
- Порівняно результати роботи RL та алгоритму Дейкстри, що дозволило оцінити переваги та недоліки кожного методу в контексті реальних умов.

Висновки щодо ефективності методів:

Використання методу Reinforcement Learning (RL) в плануванні маршрутів демонструє значні переваги порівняно з класичним алгоритмом Дейкстри. RL дозволяє агенту адаптуватися до змінних умов, що є критично важливим у сучасних динамічних транспортних системах. Завдяки можливості враховувати більше факторів, таких як поточні дорожні умови та трафік, RL забезпечує вибір оптимального маршруту, який мінімізує загальні витрати і час.

Один із ключових аспектів RL – здатність агента навчатися на основі взаємодії з середовищем, що дозволяє системі постійно вдосконалюватися та адаптуватися до нових умов. Це підвищує ефективність та гнучкість системи в реальному часі, роблячи її здатною швидко реагувати на непередбачувані ситуації та зміни.

ДОДАТОК А

Index	Speed	RLReward	RLTime	RLPath	Dijkstra'sTime	DijkstraPath
0	10	55	110	[5, 4, 3, 6]	96,90	[5, 0, 1, 3, 6]
1	15	120	39,7	[5, 0, 1]	38,90	[5, 0, 1]
2	10	110	80,2	[1, 2, 6]	37,20	[1, 3, 6]
3	18	95	62,6	[5, 4, 0, 1]	44,10	[5, 0, 1]
4	13	85	58,7	[5, 0, 1, 3]	62,50	[5, 0, 1, 3]
5	14	95	44,1	[1, 3, 4]	43,50	[1, 3, 4]
6	10	85	36,3	[0, 1, 3]	35,80	[0, 1, 3]
7	18	110	23,6	[4, 5]	23,30	[4, 5]
8	7	55	125,6	[5, 4, 3]	106,80	[5, 0, 1, 3]
9	13	115	69,9	[5, 4, 0]	30,80	[5, 0]
10	13	95	19,5	[7, 2]	19,20	[7, 2]

11	11	120	31,9	[0, 1, 3]	31,90	[0, 1, 3]
12	14	45	28	[5, 0]	28,90	[5, 0]
13	18	120	58,3	[5, 4, 3, 1]	46,70	[5, 0, 1]
14	10	90	94,7	[7, 2, 1, 0]	95,10	[7, 2, 1, 0]
15	18	70	79	[7, 2, 1, 3]	71,90	[7, 2, 3]
16	18	110	45	[1, 2, 6]	21,60	[1, 3, 6]
17	9	105	40,1	[0, 1, 3]	40,30	[0, 1, 3]
18	9	120	130,5	[7, 2, 1, 0]	138,60	[7, 2, 1, 0]
19	10	65	36,3	[0, 1, 3]	36,40	[0, 1, 3]
20	7	110	32,4	[6, 3]	32,40	[6, 3]
21	9	110	34,2	[2, 6]	34,30	[2, 6]
22	18	110	49,9	[5, 4, 3]	43,00	[5, 0, 1, 3]
23	14	120	42,9	[1, 0, 5]	42,90	[1, 0, 5]
24	7	120	60,1	[4, 5]	60,90	[4, 5]

25	12	120	25,3	[2, 6]	25,40	[2, 6]
26	16	100	55,2	[5, 4, 3]	48,30	[5, 0, 1, 3]
27	9	120	67	[1, 0, 5]	90,50	[1, 0, 5]
28	16	120	62,6	[5, 0, 1, 3, 6]	63,10	[5, 0, 1, 3, 6]
29	16	95	66,7	[5, 4, 0, 1]	43,40	[5, 0, 1]
30	10	120	43,1	[5, 4]	42,30	[5, 4]
31	13	120	29,4	[1, 3, 6]	29,40	[1, 3, 6]
32	11	115	45,2	[2, 1]	45,00	[2, 1]
33	11	120	32,4	[0, 1, 3]	32,20	[0, 1, 3]
34	18	120	12,8	[6, 3]	12,70	[6, 3]
35	13	100	106,3	[7, 2, 1, 3, 4]	95,00	[7, 2, 3, 4]
36	16	120	39,4	[1, 3, 4]	38,90	[1, 3, 4]
37	18	95	50,3	[3, 4, 5]	43,00	[3, 1, 0, 5]

38	16	120	22,6	[0, 1, 3]	22,60	[0, 1, 3]
39	7	120	100,3	[2, 1, 0]	100,40	[2, 1, 0]
40	15	120	42,3	[1, 3, 4]	42,20	[1, 3, 4]
41	8	120	153,1	[7, 2, 3, 4]	175,10	[7, 2, 3, 4]
42	14	95	73,6	[1, 0, 5, 4]	64,90	[1, 3, 4]
43	15	120	51,2	[3, 1, 0, 5]	51,20	[3, 1, 0, 5]
44	8	120	99,2	[7, 2, 6, 3]	149,60	[7, 2, 3]
45	18	120	35,4	[1, 3, 4]	35,30	[1, 3, 4]
46	8	120	45,1	[0, 1, 3]	45,60	[0, 1, 3]
47	15	120	42,1	[1, 3, 4]	49,70	[1, 3, 4]
48	14	120	18,5	[7, 2]	19,10	[7, 2]
49	7	120	51	[2, 7]	63,50	[2, 7]

ДОДАТОК Б

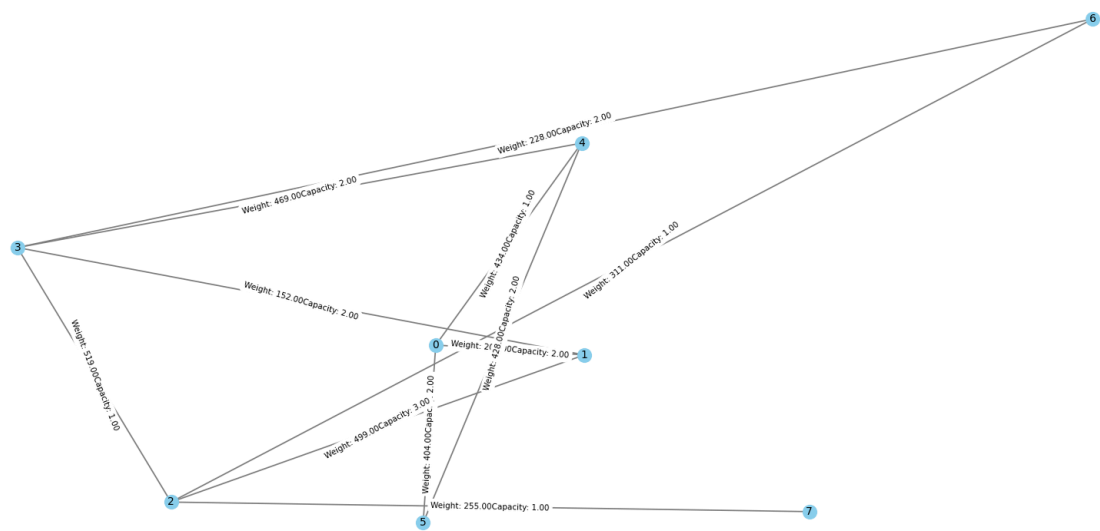
Weight	0	1	2	3	4	5	6	7
0	-	207	-	-	434	404	-	-
1	207	-	499	152	-	-	-	-
2	-	499	-	519	-	-	311	255
3	-	152	519	-	469	-	228	-
4	434	-	-	469	-	428	-	-
5	404	-	-	-	428	-	-	-
6	-	-	311	228	-	-	-	-
7	-	-	255	-	-	-	-	-

ДОДАТОК В

Capacity	0	1	2	3	4	5	6	7
----------	---	---	---	---	---	---	---	---

0	-	2	-	-	1	2	-	-
1	2	-	3	2	-	-	-	-
2	-	3	-	1	-	-	1	1
3	-	2	1	-	2	-	2	-
4	1	-	-	2	-	2	-	-
5	2	-	-	-	2	-	-	-
6	-	-	1	2	-	-	-	-
7	-	-	1	-	-	-	-	-

ДОДАТОК Г



СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

<https://matplotlib.org/>

<https://networkx.org/>

[All the Benefits Of Artificial Intelligence \(wgu.edu\)](#)

https://www.researchgate.net/publication/220696313_Algorithms_for_Reinforcement_Learning

https://www.researchgate.net/publication/2500611_On-Line_Q-Learning_Using_Connectionist_Systems

<https://hal.science/hal-03171590/file/dijkstra1959groupefmr.pdf>

<https://archive.org/details/linearnetworkopt0000bert/page/n3/mode/1up>