

УДК 519.85:658.846

РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ ПОШУКУ ОПТИМАЛЬНИХ РІШЕНЬ З ВИКОРИСТАННЯМ ТЕОРІЇ ГРАФІВ

Войтко В.В., Денисюк А.В., Гавенко О.В., Стахов А.О.

Анотація: Запропоновано метод пошуку оптимального шляху комівояжера з використанням засобів штучного інтелекту. Розроблена автоматизована система дозволяє здійснювати пошук оптимального шляху між усіма чи заданими вершинами неорієнтованого графа.

Аннотация: Предложен метод поиска оптимального пути коммивояжера с использованием средств искусственного интеллекта. Разработанная автоматизированная система позволяет осуществлять поиск оптимального пути между всеми или заданными вершинами неориентированного графа.

Abstract: A method for finding optimal traveling salesman with the use of artificial intelligence. The developed automated system allows you to search the optimal path between all vertices, or given disorientated graph.

Ключові слова: оптимальний шлях, автоматизована система, граф, шлях комівояжера, штучний інтелект.

Вступ

Задачі пошуку оптимальних рішень класично є базовими у різних галузях людської діяльності: економіці, логіці, техніці, медицині, мережевих технологіях тощо. Розв'язок широкого кола оптимізаційних задач можна формалізувати до опису математичної моделі критеріїв оптимальності засобами теорії графів та математичної статистики. Такий підхід дозволяє розглядати питання оптимізації як пошук найкоротших шляхів між вершинами графа з урахуванням фізичного значення його вершин та дуг [1,2].

Актуальність дослідження алгоритмів пошуку оптимальних рішень

Значну кількість оптимізаційних задач можна розв'язувати за допомогою засобів теорії графів, звівши оптимізаційну задачу до пошуку найкоротших шляхів між двома чи усіма вершинами графа або пошуку шляху комівояжера. Проте такий підхід часто вимагає трудомістких та складних розрахунків. Сучасні програмні засоби аналізу графів, такі як VPathOptimizer, Graf v1.0, Графоаналізатор, MaxFlow мають спеціалізований характер, що обмежує їх широке використання. Тому актуальною є розробка нових методів та алгоритмів пошуку оптимальних шляхів між заданою кількістю вершин графа та їх реалізація в автоматизованій системі, призначеній для розв'язання широкого кола оптимізаційних задач теорії графів.

Мета дослідження

Метою дослідження є покращення якісних характеристик методів пошуку оптимальних шляхів між заданими вершинами графа та зменшення трудомісткості розв'язання оптимізаційних задач.

Під об'єктом дослідження розуміємо оптимізаційні процеси пошуку кінцевого рішення та формалізацію математичної моделі розв'язку засобами теорії графів. Предметом дослідження постають методи та алгоритми пошуку оптимального шляху між заданими вершинами графа.

Постановка задачі

Основними задачами роботи вбачаємо розробку та удосконалення методів пошуку оптимальних шляхів та реалізацію запропонованих методів в автоматизованій системі.

Часто умову задачі можна звести до опису за допомогою моделі неорієнтованого графа. Розв'язок задачі тоді полягає у пошуку оптимальних шляхів між заданою кількістю вершин графа. Формалізація даних проводиться в два етапи. На першому етапі обирається критерій оптимальності, а на другому – вхідні дані процесу пошуку оптимального рішення подаються у вигляді неорієнтованого графа, причому вершини та ребра графа характеризують структуру даних задачі, а ваги ребер – їх інформаційне значення.

Задачі пошуку оптимального рішення поділяються на:

- пошук найкоротшого шляху між двома вершинами графа;
- пошук найкоротших шляхів між усіма вершинами графа;
- пошук оптимального шляху комівояжера.

Розробка методу пошуку оптимальних рішень

У середовищі автоматизованої системи реалізовано комбінований підхід до вибору алгоритмів пошуку оптимальних рішень. Так пошук оптимального шляху між заданою кількістю вершин графа здійснюється за алгоритмами Дейкстри та Флойда, а пошук оптимального шляху комівояжера реалізовано за допомогою розробленого удосконаленого алгоритму з використанням засобів штучного інтелекту. Такий підхід дозволяє проводити аналіз графа в умовах динамічної зміни його структури, обираючи на кожному етапі дослідження потрібні алгоритми розрахунків залежно від зміни вхідних вимог.

Нехай вхідні дані задачі презентовано у вигляді неорієнтованого графа, який складається з шести вершин (рис. 1). Критерієм оптимальності є деяка характеристика p .

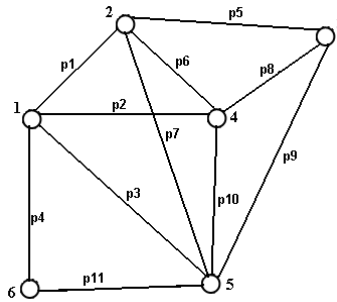


Рисунок 1 – Формалізація даних задачі у вигляді неорієнтованого графа

Вершини вхідного графа нумеруємо цілими числами від 1 до N. Позначимо довжину найкоротшого шляху з вершини i у вершину j через змінну d_{ij}^m . Якщо між вершинами i та j не існує жодного шляху, то вважаємо $d_{ij}^m = \infty$. Для будь-якої вершини i позначаємо $d_{ii}^m = 0$. На базі початкової таблиці D^m (розмірністю $N \times N$) за допомогою рекурсії визначаються елементи проміжних та кінцевої таблиць графа. Кінцева таблиця D^n містить набір найкоротших шляхів між обраними чи усіма вершинами графа [3].

Таблиця 1 містить вхідні дані (табл. D^m), які однозначно відтворюють початкові умови формування графа, зображеного на рис. 1.

Таблиця 1 – Подання графа у вигляді таблиці D^m

Номер	1	2	3	4	5	6
1	0	p1	∞	p2	p3	p4
2	p1	0	p5	p6	p7	∞
3	∞	p5	0	p8	p9	∞
4	p2	p6	p8	0	p10	∞
5	p3	p7	p9	p10	0	p11
6	p4	∞	∞	∞	p11	0

Матриця D^{n+1} розраховується у процесі циклічного аналізу елементів матриці D^n за правилом:
 $d_{ij}^{n+1} = \min\{d_{ij}^n, d_{i(n+1)}^n + d_{(n+1)j}^n\}$, де $i \neq j$; $d_{ii}^{n+1} = 0$.

Якщо задача зводиться до пошуку найкоротшого шляху між двома вершинами графа, то через d_i позначимо довжину найкоротшого шляху від початкової вершини k до вершини i , а через U_i – булеву змінну, яка визначає факт розгляду вершини i у процесі виконання алгоритму. Для будь-якої вершини $i \neq k$ вважаємо, що $d_i = \infty$, $d_k = 0$ і $U_i = 0$. Пошук вершини i відбувається за ідентифікацією найкоротшої довжини шляху до вершини k (тобто $\min(d_i)$), коли $U_i = 0$. Знайшовши таку вершину, покладаємо $U_i = 1$ і перевіряємо кожну її сусідню вершину. Якщо існує така сусідня вершина j , для якої справедливою є нерівність $(d_i > d_j + V_{ij})$, де V_{ij} – довжина ребра, що з'єднує вершини i та j , то вважаємо $(d_i = d_j + V_{ij})$. Цикл завершується після розгляду всіх вершин графа (якщо для будь-якої вершини i виконується умова $U_i = 1$) або у випадку, коли для будь-якої вершини i (такої, що $U_i = 0$) виконується умова $d_i = \infty$; d_i отримує значення довжини найкоротшого шляху з початкової вершини k до заданої вершини i [4].

Задача пошуку оптимального шляху комівояжера розглядається у просторі станів. Множиною можливих ситуацій буде перелік відвіданих вершин графа, записаних у порядку їх проходження. Початковим станом вважається та вершина, з якої починається пошук. Будь-який рядок, що розпочинається і закінчується числом 1 і містить у собі разову наявність усіх інших цілих чисел з діапазону $(1; n]$, є цільовим станом.

Зведемо запис таблиці вхідних даних графа (табл. 1) до виду табл. 2, у якій місце запису вагового значення кожного ребра p визначається номерами вершин, які воно поєднує, та умовою вибору відповідного елемента таблиці за правилом $i < j$.

Таблиця 2 – Табличне подання значимих параметрів графа

Номер вершини	1	2	3	4	5	6
1	-	p1	-	p2	p3	p4
2	-	-	p5	p6	p7	-
3	-	-	-	p8	p9	-
4	-	-	-	-	p10	-
5	-	-	-	-	-	p11
6	-	-	-	-	-	-

Операторами переходу є переходи від поточної вершини до сусідніх вершин графа:

$$S_1X \xrightarrow{Y} S_1XY,$$

де X – поточна вершина, а Y – сусідня вершина.

Оператор вважаємо забороненим, якщо він призводить до недопустимого опису стану, або коли не існує ребра між вершинами X і Y. Оціночна функція алгоритму пошуку оптимального шляху комівояжера має вигляд:

$$\hat{f}(n) = \hat{g}(n) + \hat{h}(n),$$

де $\hat{g}(n)$ - оцінка вартості шляху з початкової вершини до вершини n;

$\hat{h}(n)$ - евристична функція оцінки вартості найкоротшого шляху з вершини n до цільової вершини.

Через $g(n)$ позначимо оптимальну вартість шляху з початкової вершини до вершини n. Під $h(n)$ розуміємо оптимальну вартість шляху від вершини n до будь-якої із цільових вершин графа.

Тоді $\hat{g}(n)$ визначатиметься довжиною маршруту від початкової вершини графа до поточної вершини, де n є номером вершини дерева простору станів.

Евристична функція визначається як середнє арифметичне комплексних показників мінімального і максимального значень ребер графа :

$$\hat{h}(n) = \frac{\min(G) + \max(G)}{2},$$

де $\min(G) = \sum_{i=1}^n \min_i(G)$, а $\max(G) = \sum_{i=1}^n \max_i(G)$.

Через $\min_i(G)$ позначаємо знайдене мінімальне значення i-го рядка табл. 2, яка містить параметри вхідного графа, а через $\max_i(G)$ – максимальне значення i-го рядка цієї таблиці.

З використанням засобів штучного інтелекту пошук відбувається в найперспективнішому напрямку. Послідовно розглядаються вершини у порядку зростання їх оціночних функцій \hat{f} . Пошук припиняється за умови досягнення цільової вершини графа.

На рисунку 2б наведено приклад пошуку оптимального шляху комівояжера у просторі станів для графа, зображеного на рисунку 2а.

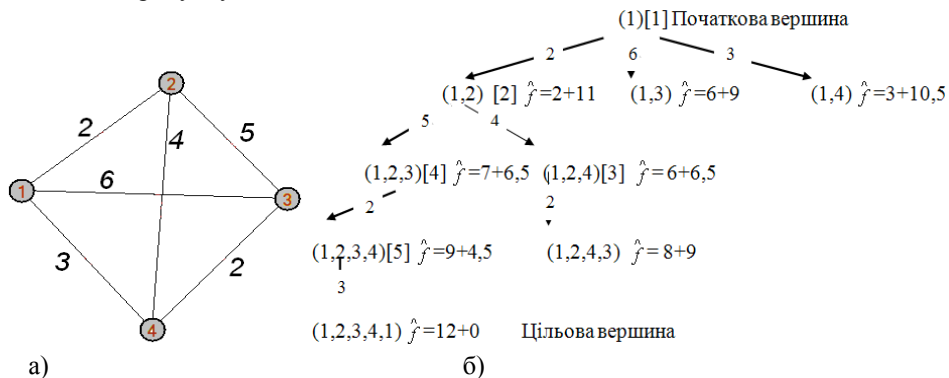


Рисунок 2 – Приклад пошуку оптимального шляху комівояжера у просторі станів: а) – граф; б) – алгоритм розрахунків

Для графа (рис. 2а) кількість відкритих вершин простору станів розробленим алгоритмом пошуку оптимального шляху комівояжера (рис. 2б) складає 9 вершин, а загальна кількість вершин простору станів складає 22 вершини. Використання засобів штучного інтелекту дозволяє збільшити ефективність процесу пошуку оптимального шляху комівояжера за рахунок зменшення кількості

варіантів перебору вершин простору станів.

Розробка автоматизованої системи пошуку оптимальних рішень

Автоматизована система пошуку оптимального рішення передбачає реалізацію множини підходів до отримання вхідних даних, вибір алгоритмів оптимізації графа, виведення результатів пошуку оптимального рішення (рис. 3).

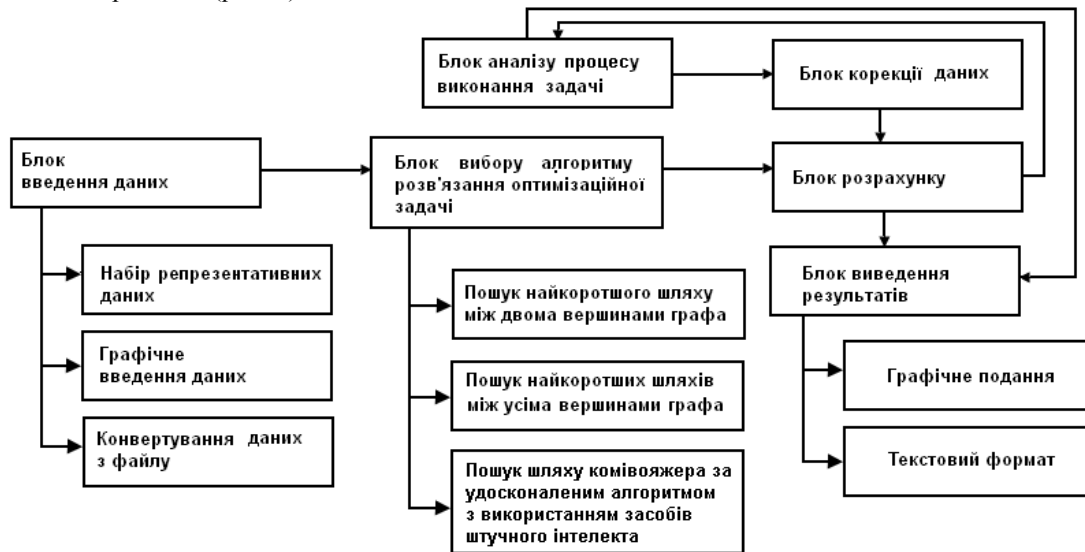


Рисунок 3 – Модель автоматизованої системи пошуку оптимальних рішень

Блок введення даних дозволяє користувачеві обрати зручний формат отримання вхідних даних.

Блок вибору алгоритму розв'язання оптимізаційної задачі передбачає можливість пошуку найкоротшого шляху між двома вершинами графа; пошуку найкоротших шляхів між усіма вершинами графа; пошуку оптимального шляху комівояжера.

Блок розрахунку виконує алгоритмізовану обробку даних, орієнтовану на пошук оптимальних рішень. Блок аналізу здійснює моніторинг процесу розв'язання задачі на предмет виконання усіх вхідних вимог. Блок корекції даних забезпечує можливість динамічної зміни структури графа шляхом корегування вхідних даних. Блок виведення результатів проводить підготовку вихідної інформації до подання в обраному форматі та здійснює відображення результатів роботи системи.

Результати роботи системи у випадку пошуку оптимального шляху комівояжера подано на рис. 4. Графічний формат відображення вихідних даних дозволяє наочне виведення знайденого оптимального маршруту.

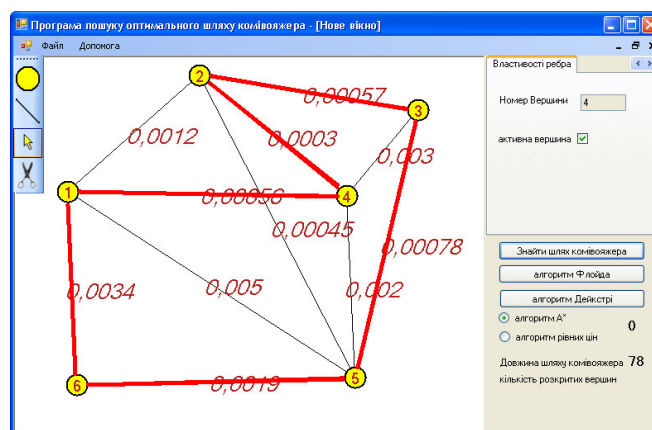


Рисунок 4 – Результат пошуку оптимального шляху комівояжера

Висновок

Розроблена автоматизована система пошуку оптимальних рішень призначена для вибору удосконалених алгоритмів теорії графів у процесі розв'язання оптимізаційних задач і орієнтована на забезпечення можливості пошуку оптимального шляху між обраними чи всіма вершинами графа. Запропонований удосконалений алгоритм пошуку шляху комівояжера базується на використанні засобів штучного інтелекту, що дозволяє підвищити ефективність процесу пошуку оптимального рішення.

Список літератури

1. Емеличев В.А. Многогранники, графы, оптимизация. / В.А. Емеличев, М.М. Ковалев. – М. : Наука, 1991. – 342с. – ISBN 5-900916-48-0.
2. Войтко В.В. Комбінований метод пошуку оптимальних рішень з використанням засобів теорії графів / В.В. Войтко, С.В. Бевз, С.М. Бурбело, О.В. Гавенко // Оптико-електронні інформаційно-енергетичні технології. – №2 (16). – 2008. – С. 67-70. – ISSN 1681-7893.
3. Оре О. Графы и их применение. / О. Оре. – М. : ЛКИ, 2001. – 350с. – ISBN 5-466-00113-9.
4. Харари Ф.О. Теория графов. / Ф.О. Харари. – Москва: Едиториал, 2006. – 300с. – ISBN 5-354-00301-6.

Відомості про авторів

Войтко Вікторія Володимирівна – к.т.н., доцент, Україна, м. Вінниця, Вінницький національний технічний університет, кафедра програмного забезпечення, vojtko@vstu.vinnica.ua.

Денисюк Алла Василівна – асистент, Україна, м. Вінниця, Вінницький національний технічний університет, кафедра програмного забезпечення.

Гавенко Олег Віталійович – Україна, м. Вінниця, Вінницький національний технічний університет, кафедра програмного забезпечення, магістрант, Gavenko2007@yandex.ru.

Стахов Артем Олександрович – Україна, м. Вінниця, Вінницький національний технічний університет, кафедра програмного забезпечення, студент, artemstahov@mail.ru.