

Пошук шляхів у маршрутних мережах міст методом відгалужень і меж

Кузькін О. Ф., канд. техн. наук

Запорізький національний технічний університет

69063 Україна, м. Запоріжжя, вул. Жуковського, 64

Особливість задачі пошуку шляхів пересування пасажирів у маршрутних мережах міст, на відміну від звичайної задачі пошуку найкоротших шляхів у мережах, полягає у тому, що при її розв'язанні необхідно враховувати маршрутні обмеження та додаткові умови, які полягають у наступному:

1) в процесі здійснення поїздки між двома заданими зупинками у деяких випадках виникає необхідність зміни маршруту (пересадки);

2) навіть якщо між парою зупинок існує безпересадочний шлях, він може бути не найкоротшим за критерієм тривалості чи відстані;

3) тривалість пересадки залежить від просторового розміщення зупинок на вулично-дорожній мережі та інтервалу руху на маршруті, на який виконується пересадка;

4) тривалість руху між зупинками маршрутної мережі залежить від виду міського транспорту, використововуваного рухомого складу і режимів руху на маршрутах.

Ефективність розв'язання задачі раціональної маршрутизації пасажирських перевезень у містах, яка відноситься до класу NP-повних задач, залежить, серед іншого, від ефективності алгоритму пошуку найкоротших шляхів на маршрутній мережі.

Пропонується алгоритм комбінаторної задачі пошуку найкоротших шляхів з врахуванням тривалості пересадок, в основі якого лежить метод відгалужень і меж. Алгоритм, реалізація якого наведена для прикладу, коли тривалість пересадки у деякій вершини залежить тільки від її номеру, а

тривалості руху між вершинами не залежать від маршрутів, складається з семи кроків.

Крок 1. Розрахунок найкоротших відстаней між вершинами маршрутної мережі (зупинками, центрами транспортних районів) без врахування необхідності пересаджень за допомогою одного з алгоритмів пошуку найкоротших шляхів, наприклад, Дейкстри чи Форда-Воршала.

Крок 2. Визначення нижньої оцінки довжини найкоротшого шляху $h(T)$ як довжини найкоротшого шляху між початковою та кінцевою вершинами без врахування пересадок.

Крок 3. Визначення верхньої оцінки довжини найкоротшого шляху за одним з методів у порядку зменшення точності: а) за умови пересадок у кожній проміжній вершині найкоротшого шляху; б) з використанням жадібного алгоритму, який гарантує отримання шляху з найменшою кількістю пересадок; в) за допомогою пошуку найкоротшого шляху на заданій послідовності вершин з використанням алгоритму для орієнтованого ациклічного графа.

Крок 4. Розгалуження кореня дерева пошуку розв'язків на підставі можливих маршрутів руху з початкової вершини.

Крок 5. Вибір вершини-кандидата для розгалуження серед множини нерозгалужених за величиною найменшої оцінки.

Крок 6. Перевірка вершини-кандидата та її оцінки. Якщо номер вершини кандидата співпадає з номером кінцевої вершини, то оптимальне рішення знайдене, інакше переходимо до кроку 7.

Крок 7. Розгалуження вершини-кандидата. Гілки, за якими виконується розгалуження, відповідають можливим шляхам прямування з поточної вершини до суміжної їй з урахуванням необхідності зміни маршруту у останній усіма можливими маршрутами.

Крок 8. Розрахунок оцінки відгалужених вершин. Оцінки відгалужених на кроці 7 вершин розраховуються за формулою

$$b' = b^* + t_{k^*k'} + d_{k'q} - d_{k^*q} + f(k', r^*, r'), \quad (1)$$

де b^* – оцінка попередньої вершини дерева пошуку розв’язків, хв.;

$t_{k^*k'}$ – відстань між попередньою k^* та відгалуженою k' вершинами дерева пошуку розв’язків за певним маршрутом r^* , хв.;

$d_{k'q}$ – найкоротша відстань між відгалуженою вершиною k' та кінцевою вершиною q на маршрутній мережі без врахування пересадок за результатами кроку 1, хв.;

d_{k^*q} – найкоротша відстань між попередньою вершиною дерева пошуку розв’язків та кінцевою вершиною q на маршрутній мережі без врахування пересадок, хв.;

$f(k', r^*, r')$ – функція, що враховує тривалість пересадки

$$f(k', r^*, r') = \begin{cases} 0, & \text{якщо } r^* = r'; \\ \tau_{k'}, & \text{якщо } r^* \neq r'. \end{cases} \quad (2)$$

Після цього переходять до кроку 5.

Представлений алгоритм може бути легко модифікований для врахування різноманітних обмежень і додаткових умов, зазначених на початку, зокрема для пошуку найкоротшого маршруту, який містить не більше заданої кількості пересадок.

Програмна реалізація алгоритму на мові програмування Python 2.7, з використанням для зберігання множини нерозгалужених вершин у бінарній купі і пошуку верхньої оцінки довжини найкоротшого шляху за допомогою жадібного алгоритму, була протестована на змодельованій маршрутній мережі, яка налічує 110 вершин та 1008 маршрутів. Середня тривалість пошуку найкоротшого шляху між заданою парою вершин склала близько 1,32 мс машинного часу. З цього можна зробити висновок, що розроблений алгоритм є ефективним у обчислювальному плані і може бути застосований для рішення практичних задач, які потребують пошуку шляхів у маршрутних мережах міст, зокрема, задачі маршрутизації пасажирських перевезень у містах.

