

УДК 519.8

Н. Гарматій, канд.екон.наук, доц.

О. Стефанюк ст. гр. ПКс-32

Тернопільський національний технічний університет імені Івана Пулюя, Україна

## ТЕОРЕТИЧНІ ТА ПРАКТИЧНІ АСПЕКТИ МОДЕЛЮВАННЯ ТЕОРІЇ МАСОВОГО ОБСЛУГОВУВАННЯ ПРИ ІМПЛЕМЕНТАЦІЇ ТЕРМІНАЛІВ МОНОБАНК НА ОБ'ЄКТАХ

N.Harmatiy, Ph.D, Assos Prof.

O. Stefaniuk

Ternopil Ivan Puluj National Technical University, Ukraine

## THEORETICAL AND PRACTICAL ASPECTS OF MODELLING THE MASS SERVICE THEORY IN THE IMPLEMENTATION OF MONOBANK TERMINALS AT THE FACILITIES

Масове обслуговування в сучасному світі є ключовим аспектом для забезпечення ефективної роботи об'єктів, які включають термінали Monobank. Моделювання теорії масового обслуговування в контексті імплементації терміналів Monobank на об'єктах, зокрема в місті Заліщики, відкриває широкий спектр теоретичних і практичних аспектів.

За допомогою веб-сервісу Google Maps [1] вибираються 8 магазинів у місті Заліщики та створюється граф, який відображає зв'язки між ними та відділенням ПриватБанку. Магазины позначені в форматі «Об'єкт X – ОХ». Цей граф служить основою для визначення оптимального транспортного шляху (див. рисунок 1).

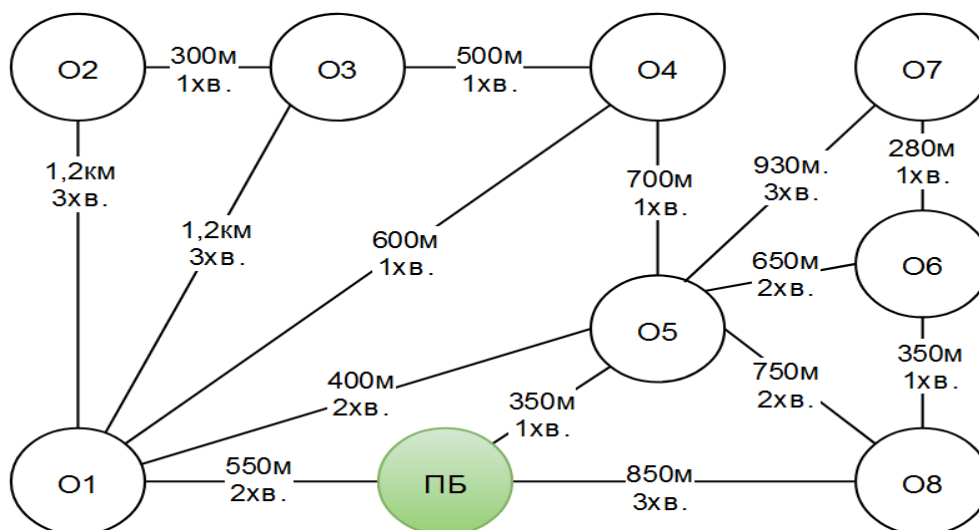


Рис.1 Граф з обраними магазинами та відділенням ПБ

Для знаходження оптимального шляху враховуючи відстані та час подорожі між вершинами в графі, використовується мова програмування Python [2]. Для цього використовувався брутфорс-підхід, який передбачає перебір всіх можливих перестановок вершин та обчислення відстані та часу подорожі для кожної перестановки. Завдяки цьому методу гарантується знаходження найоптимальнішого маршруту, але при цьому виникає експоненціальна складність, особливо для великих графів.

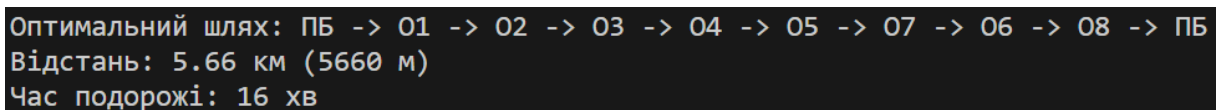
З використанням Python, створюється функція, яка призначена для визначення оптимального маршруту в графі на основі вказаних відстаней та часів подорожі між вершинами. Аналіз усіх можливих перестановок виконується шляхом використання бібліотеки Python *itertools* [3]. Код генерує всі можливі перестановки вершин, які ще не були

відвідані, і розглядає кожну з них як потенційний варіант маршруту. Для кожної варіації порядку вузлів, функція формує маршрут, розраховує загальну відстань та час подорожі використовуючи дані з графу. Якщо новий маршрут має меншу відстань і час подорожі, то він вважається оптимальним (див. лістинг 1).

Лістинг 1 – Програмний код функції для пошуку оптимального шляху

```
def find_optimal_path(graph, start_node):
    nodes_to_visit = set(graph.keys())
    nodes_to_visit.remove(start_node)
    min_distance = min_time = math.inf
    optimal_path = None
    for perm in itertools.permutations(nodes_to_visit):
        path = [start_node, *perm, start_node]
        total_distance = total_time = 0
        broken = False
        for current_node, next_node in zip(path, path[1:]):
            if next_node in graph[current_node]:
                edge_distance, edge_time =
graph[current_node][next_node]
                total_distance += edge_distance
                total_time += edge_time
            else:
                broken = True
                break
        if not broken and total_distance < min_distance:
            min_distance = total_distance
            min_time = total_time
            optimal_path = path
    return optimal_path, min_distance, min_time
```

Результатом використання створеної функції є отримання оптимального шляху з найменшою відстанню та часом проходження шляху (див. рисунок 2).



```
Оптимальний шлях: ПБ -> 01 -> 02 -> 03 -> 04 -> 05 -> 07 -> 06 -> 08 -> ПБ
Відстань: 5.66 км (5660 м)
Час подорожі: 16 хв
```

Рис. 2 – Вивід результату виконання функції для пошуку оптимального шляху

Вивчення та впровадження теорії масового обслуговування для терміналів Monobank створює нові можливості для оптимізації та удосконалення обслуговування цих терміналів. Використання програмних інструментів, таких як Python, дозволяє ефективно вирішувати задачі визначення оптимальних транспортних маршрутів, що сприяє ефективній управлінській діяльності та раціональному використанню часу та ресурсів.

### Література

1. Вебсайт картографічного веб-сервісу Google Maps URL: [maps.google.com](https://maps.google.com)
2. Офіційний вебсайт Python URL: [www.python.org](https://www.python.org)
3. Документація бібліотеки itertools URL: [docs.python.org/3/library/itertools.html](https://docs.python.org/3/library/itertools.html)