

Використання ройових алгоритмів для вирішення задачі пошуку оптимального шляху з урахуванням дорожньої обстановки

Доповідач: Нагога Анастасія Тарасівна, гр. ДА-62

Науковий керівник: Іщенко Ганна Валеріївна

ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Об'єкт: ройовий інтелект.

Предмет: побудова оптимального шляху використовуючи ройові алгоритми.

ЦІЛЬ ТА ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

Ціль: дослідження в теоретичному та практичному аспекті особливостей використання ройових алгоритмів для пошуку оптимального шляху з урахуванням дорожньої обстановки.

Задачі:

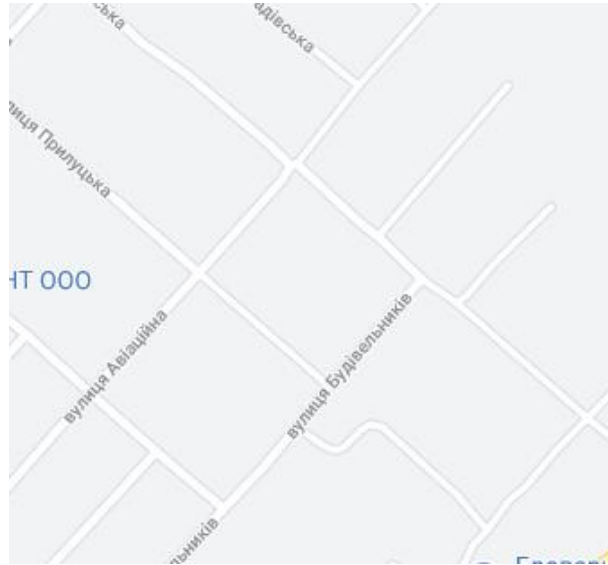
1. Дослідити предметну область задачі
2. Провести аналіз готових рішень пошуку оптимального шляху
3. Розглянути ройовий інтелект як інструмент пошуку шляху
4. Обрати ройовий алгоритм та модифікувати його для вирішення поставленої задачі пошуку шляху
5. Розробити програмну реалізацію обраного алгоритму з прив'язкою до реальної карти місцевості для побудови оптимального шляху з урахуванням дорожніх обставин

АКТУАЛЬНІСТЬ ТЕМИ



Визначення оптимального шляху для переміщення в умовах заторів є надзвичайно важливою задачею, оскільки вона допомагає запобігти екологічним, транспортним та організаційним проблемам.

ПОСТАНОВКА ЗАДАЧІ

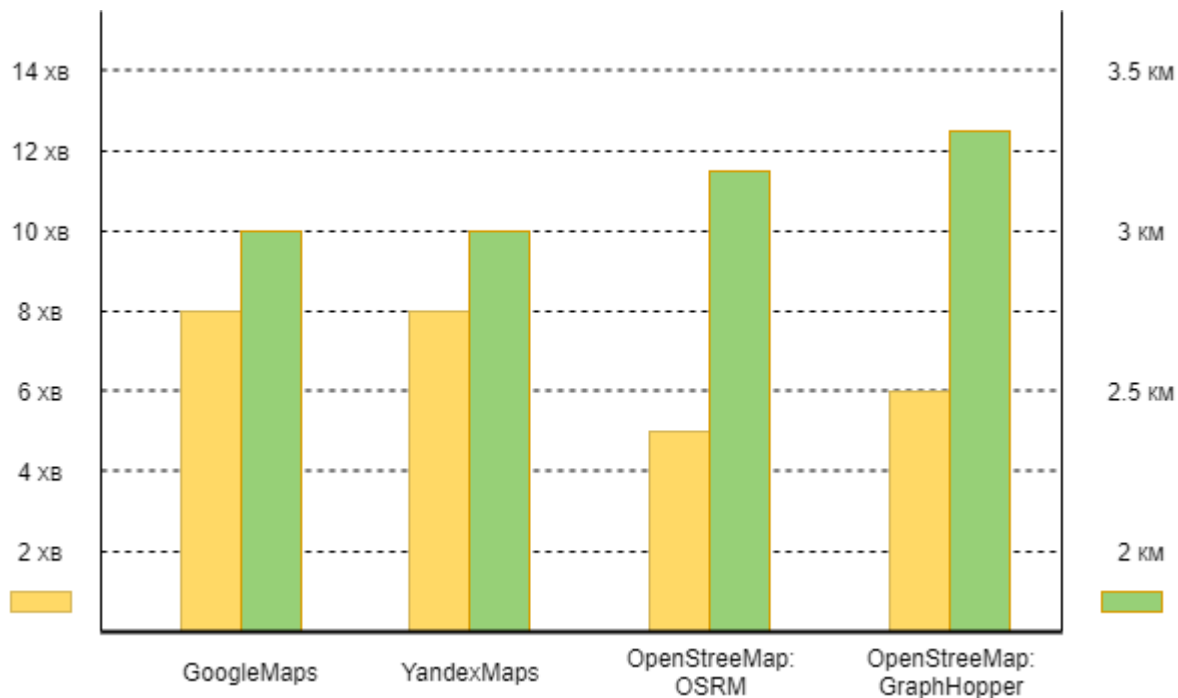


Шлях між вершиною v_1 та вершиною v_n визначається за формулою:

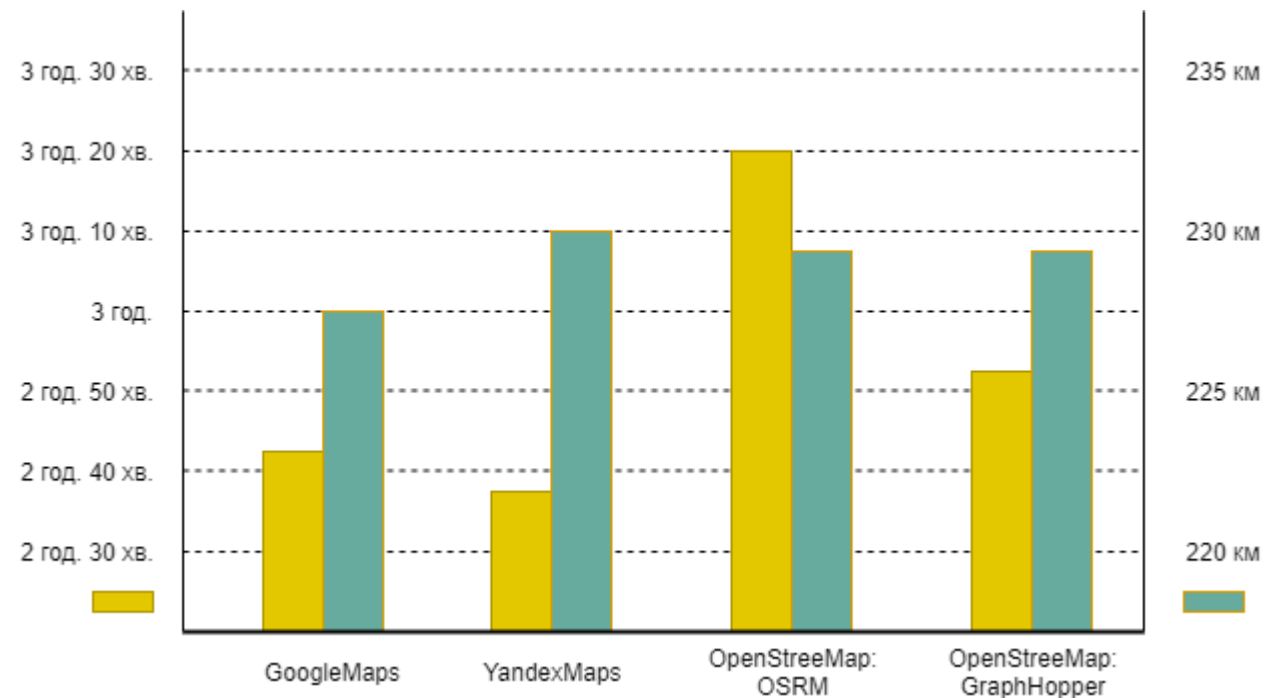
$$P = \sum_{i=1}^{n-1} g(e_{i,i+1}),$$

АНАЛІЗ КАРТОГРАФІЧНИХ СЕРВІСІВ

Пошук в межах міста

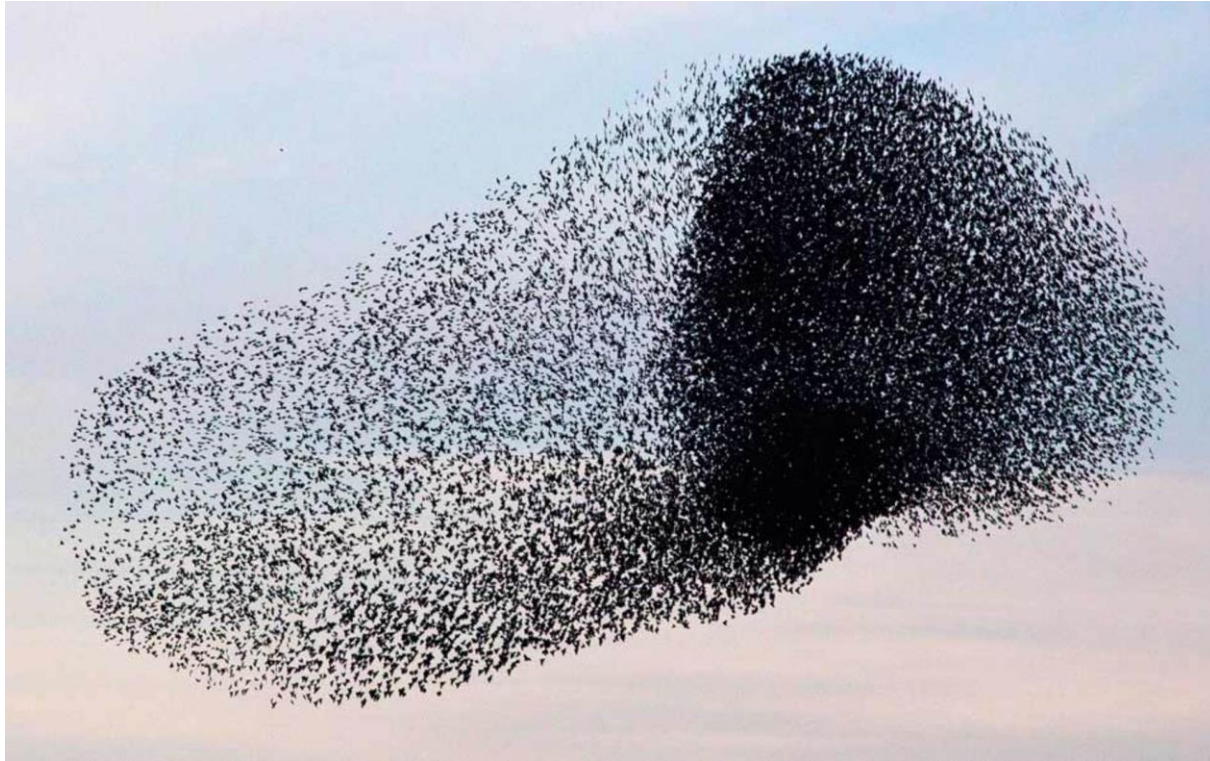


Пошук в межах країни



	Google Maps	Yandex Maps	OpenStreetMap: OSRM	OpenStreetMap: GraphHopper
Алгоритм	Dijkstra, CH	Dijkstra	CH, Multi-level Dijkstra	Dijkstra, A*, bi-directional CH
Врахування ситуації на дорозі	так	так	ні	ні

ПОНЯТТЯ РОЙОВОГО ІНТЕЛЕКТУ



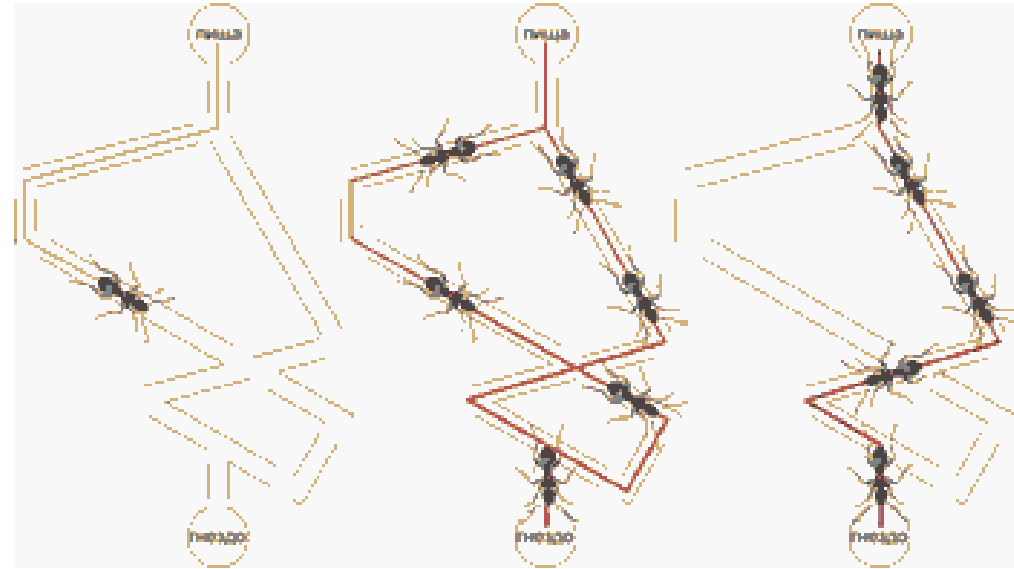
Ройовий інтелект – складна, колективна, скоординована, децентралізована та гнучка поведінка групи агентів (рою) із самоорганізацією, що дотримується простих правил.

Алгоритми:

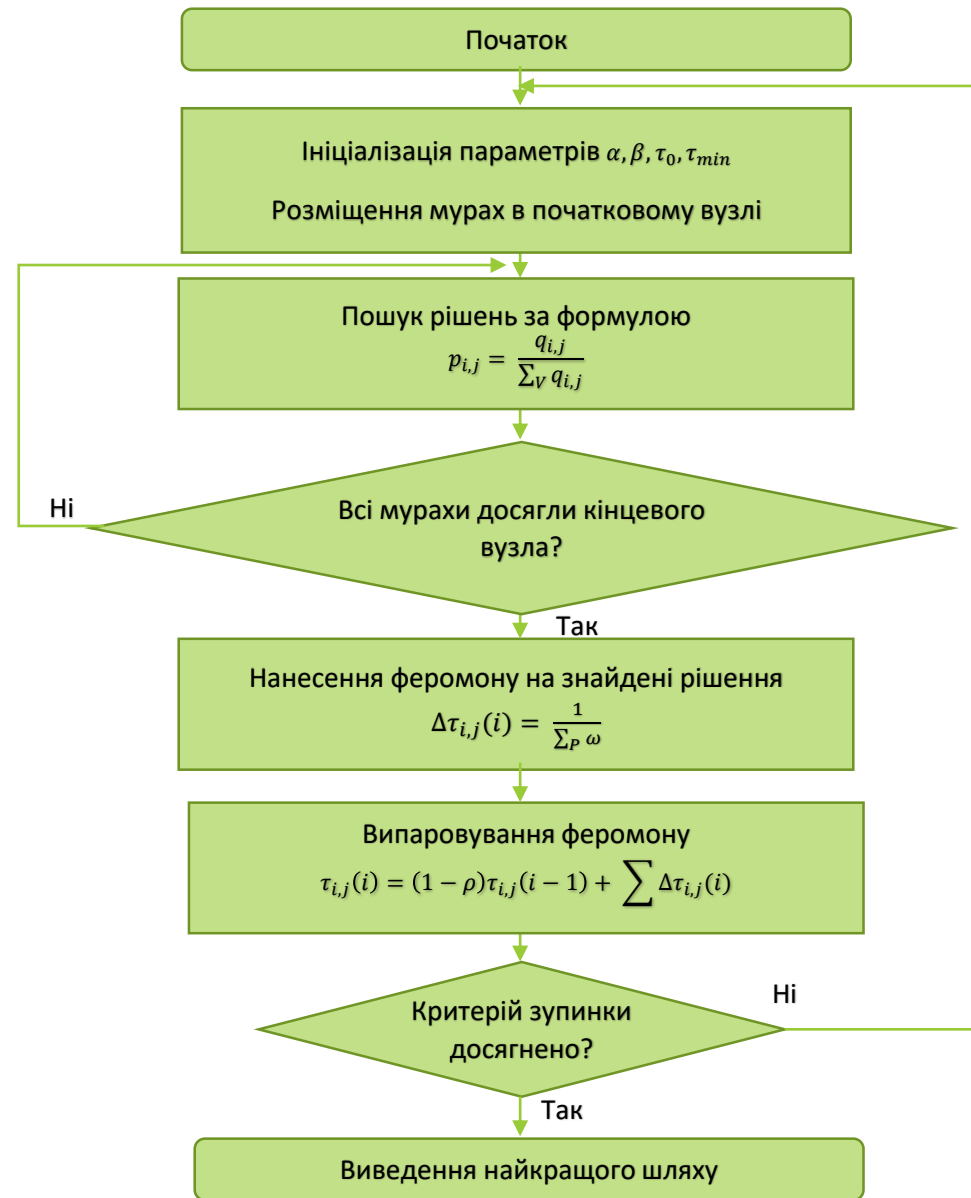
- мурашиний алгоритм,
- бджолиний алгоритм,
- алгоритм рою часток,
- алгоритм формування річок.

ОБҐРУНТУВАННЯ ВИБОРУ МУРАШИНОГО АЛГОРИТМУ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ

1. Динамічна адаптивність до змін навколишнього середовища – зміна коефіцієнту випаровуваності в залежності від завантаженості доріг.
2. Можливість вибору різного критерію зупину роботи алгоритму.
3. Представлення оптимальних шляхів в залежності від часу подолання шляху, довжини шляху, досвіду мурах.
4. Ефективність при застосуванні до великорозмірних задач.



АЛГОРИТМ МУРАШИНОЇ КОЛОНІЇ



ЗАПРОПОНОВАНА МОДИФІКАЦІЯ МУРАШИНОГО АЛГОРИТМУ

Нехай $\rho = p + q$,

де p – базова константа випаровування;

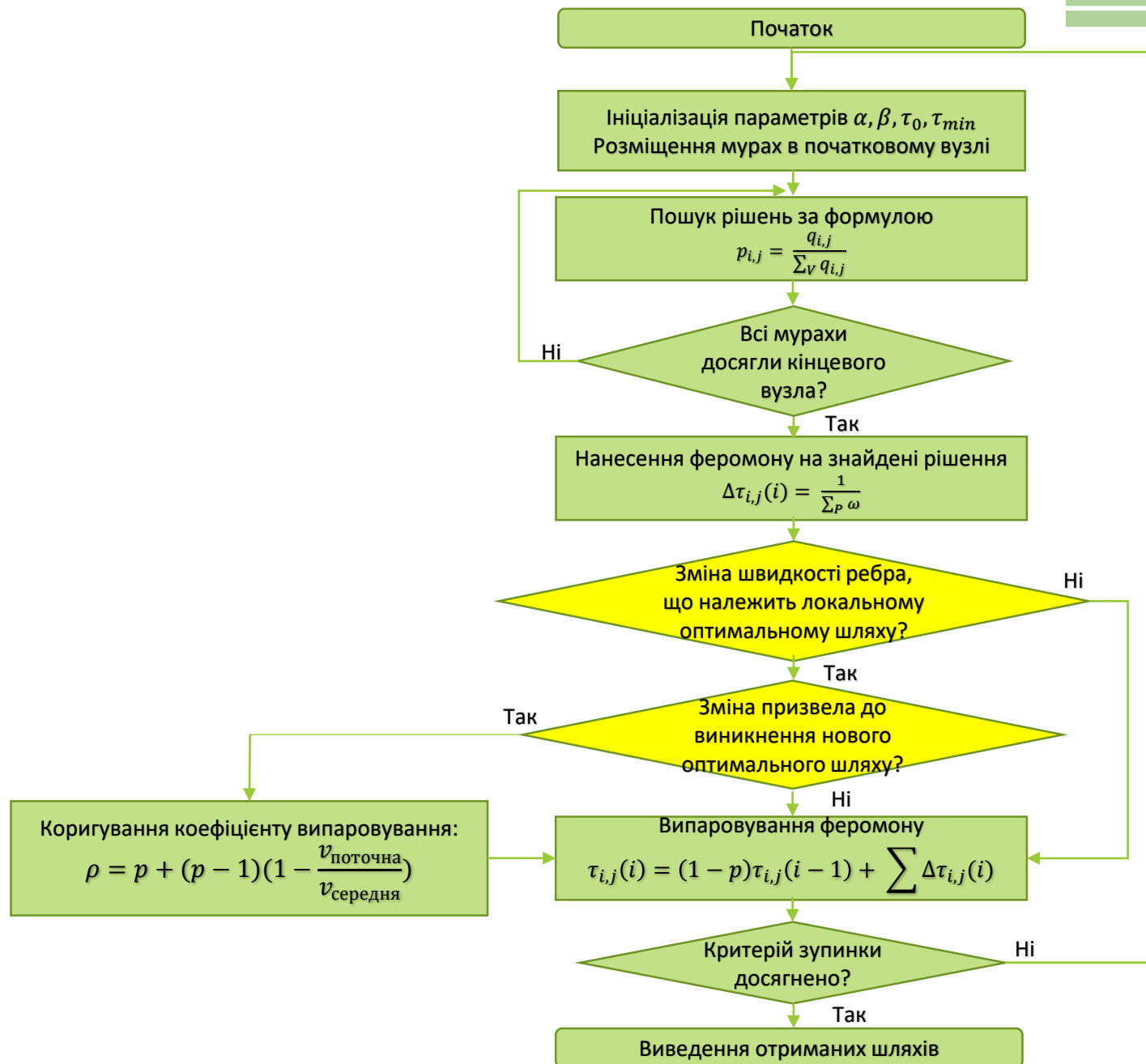
q – вплив завантаженості на коефіцієнт випаровування.

Для визначення приросту зміни швидкості я використала відношення

$$\frac{v_{\text{поточна}}}{v_{\text{середня}}}$$

Слідуючи з того, що даний приріст дає максимальне значення при малій різниці між швидкостями, коефіцієнт випаровування набуває вигляду:

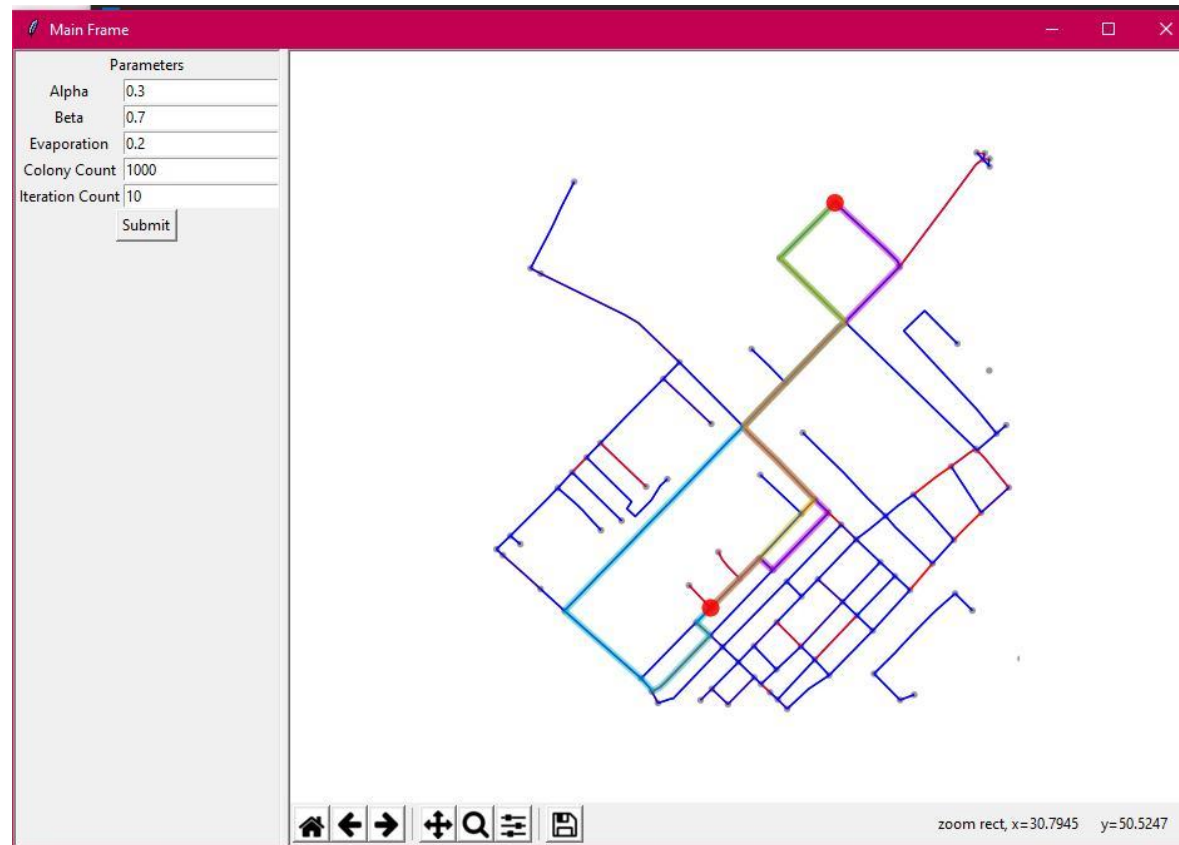
$$\rho = p + (p - 1) \left(1 - \frac{v_{\text{поточна}}}{v_{\text{середня}}}\right)$$



ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ

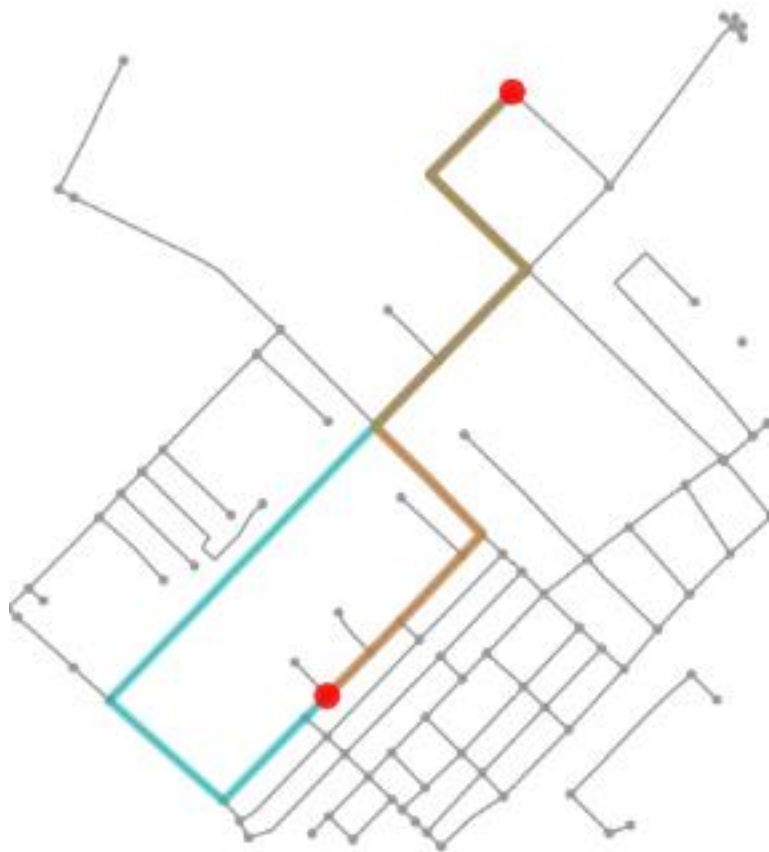
Для реалізації алгоритму використана мова програмування Python та такі пакети:

- Osmnx
- Networkx
- Tkinter
- Matplotlib
- Numpy

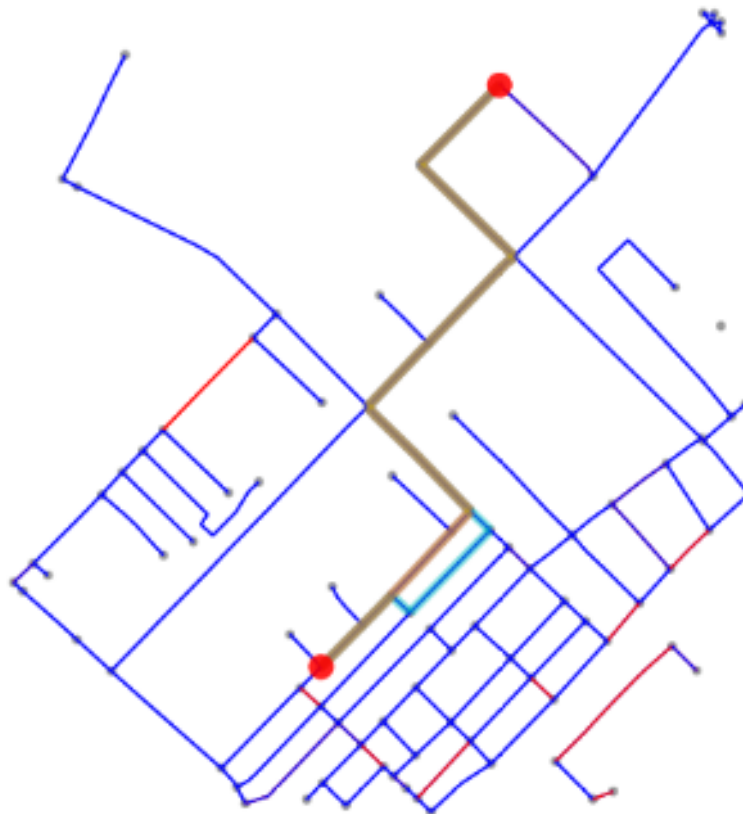


Інтерфейс користувача

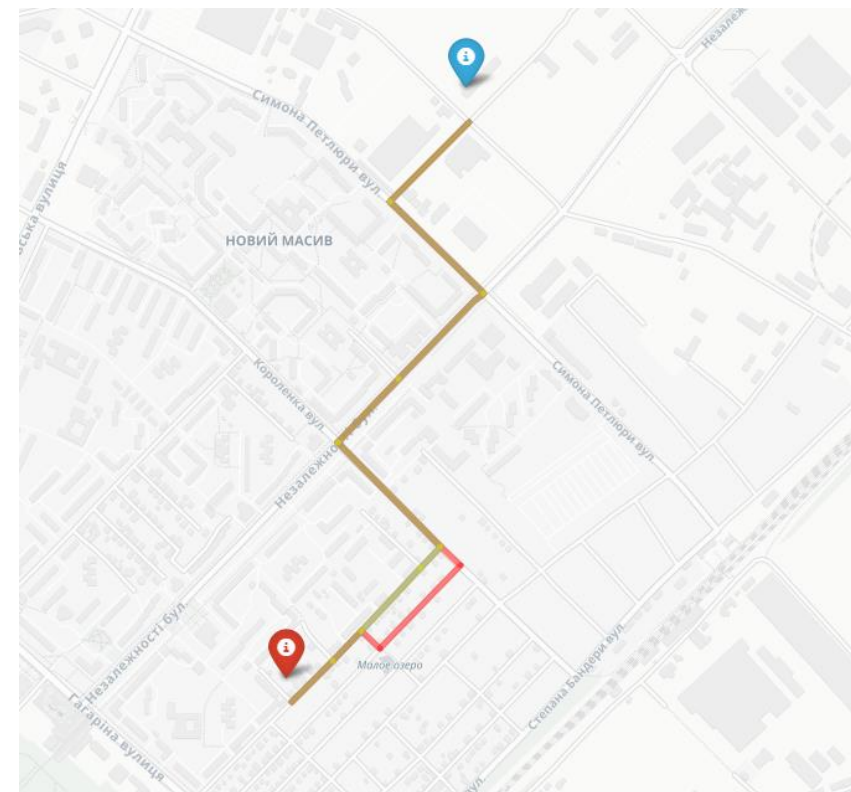
ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ



Результати роботи програми на 1 ітерації



Результати роботи програми на ітерації після утворення заторів



Результат спроектований на карту міста

РЕЗУЛЬТАТИ РОБОТИ ПРОГРАМИ НА ГРАФІ НЕВЕЛИКОГО* РОЗМІРУ

Експеримент	Alpha	Beta	Evaporation	Colony Size	min length	length
					time	min time
1	0.3	0.7	0.2	1000	1990 66	1990 66
2	0.7	0.3	0.2	1000	1987 69	1990 66
3	0.5	0.5	0.2	1000	1990 92	2117 91
4	0.3	0.7	0.5	1000	1990 100	1990 100
5	0.3	0.7	0.2	3000	1987 95	1990 92

* - кількість вершин графу до 250

РЕЗУЛЬТАТИ РОБОТИ ПРОГРАМИ НА ГРАФІ СЕРЕДНЬОГО* РОЗМІРУ

Експеримент	Alpha	Beta	Evaporation	Colony Size	min length time	length min time
1	0.3	0.7	0.2	1000	2160 94	2160 94
2	0.7	0.3	0.2	1000	2550 128	2550 128
3	0.5	0.5	0.2	1000	2556 111	2556 111
4	0.3	0.7	0.5	1000	2879 264	3806 206
5	0.3	0.7	0.2	3000	2160 191	2160 191

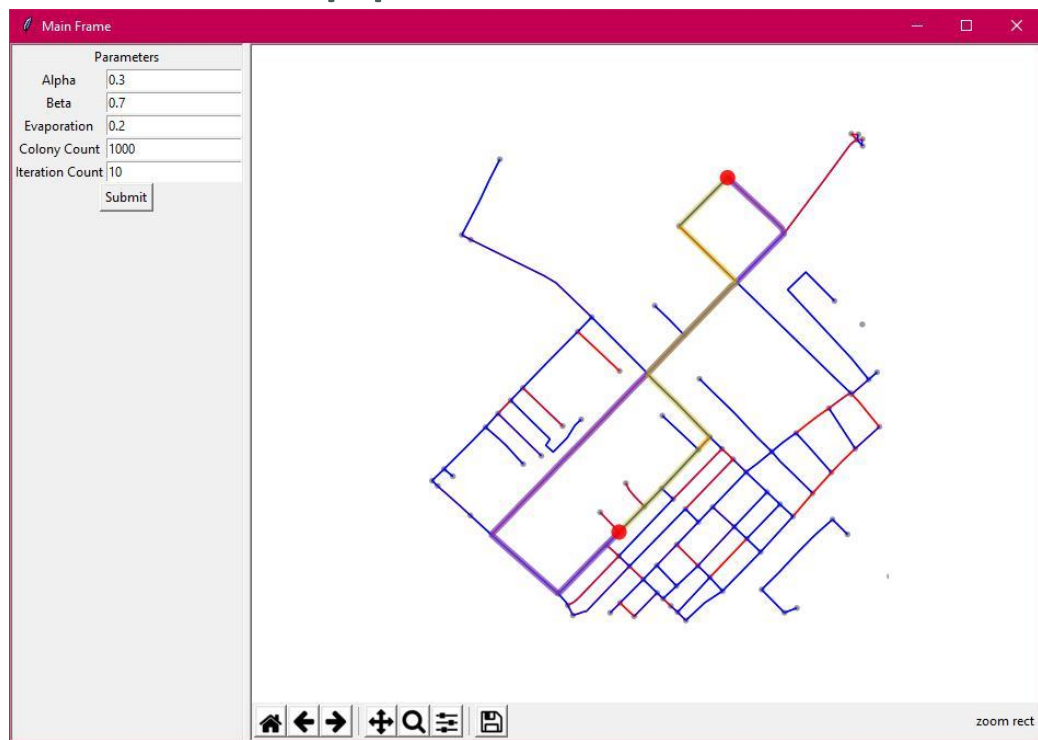
* - кількість вершин графу від 251-500

РЕЗУЛЬТАТИ РОБОТИ ПРОГРАМИ НА ГРАФІ ВЕЛИКОГО* РОЗМІРУ

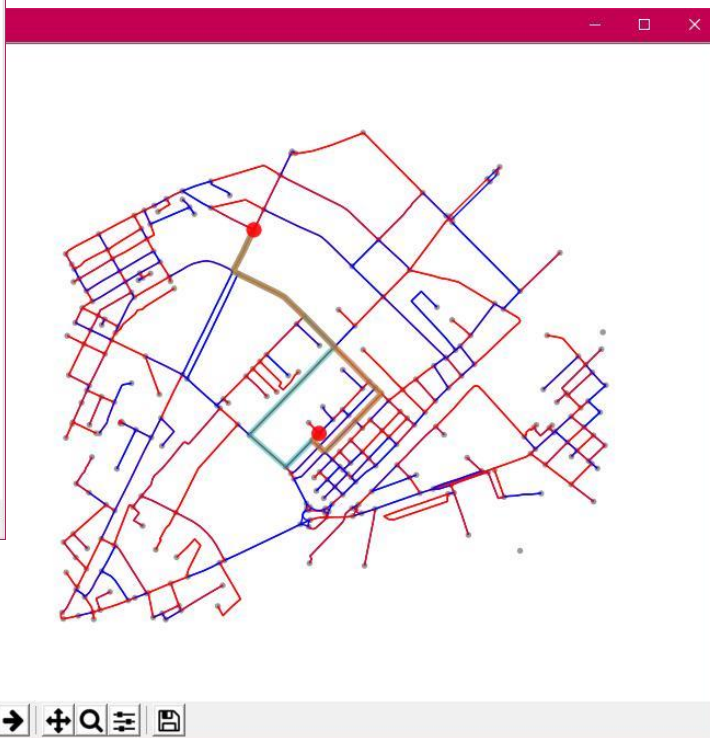
Експеримент	Alpha	Beta	Evaporation	Colony Size	min length	length
					time	min time
1	0.3	0.7	0.2	3000	5451 286	5503 279
2	0.7	0.3	0.2	3000	5282 384	5606 323
3	0.5	0.5	0.2	3000	6702 1047	7749 451
4	0.3	0.7	0.5	3000	6148 445	6148 445
5	0.3	0.7	0.2	6000	4577 564	5659 514

* - кількість вершин графу більше 501

ПРИКЛАД РОБОТИ ПРОГРАМИ НА ГРАФАХ РІЗНИХ РОЗМІРІВ



Граф невеликого розміру



Граф середнього розміру



Граф великого розміру

ВИСНОВКИ

- Науковою складовою роботи є запропонована модифікація алгоритму та її застосування до реальних даних.
- Розмір мурашиної колонії визначає якість рішення.
- На ефективність роботи алгоритму впливає коефіцієнт випаровування, а також коефіцієнти α та β .
- При великій кількості ітерації, значення оптимального шляху можна усереднити.