Вероніка Ушакова
студентка 2 курсу спеціальності
121 «Інженерія программного забезпечення».
Науковий керівник: **Н. Хацько**,
канд. техн. наук, доцент кафедри программної інженерії,
Харківський національний університет радіоелектроніки,
м. Харків

ДВОРІВНЕВИЙ АЛГОРИТМ ПОБУДОВИ ШЛЯХУ З ВРАХУВАННЯМ ПЕРЕШКОД НА МІСЦЕВОСТІ

В наш час існує багато традиційних алгоритмів знаходження найкоротшого шляху між об'єктами. Однак на практиці виникають випадки, коли потрібно не тільки обійти всі перешкоди на шляху і потрапити з початкової точки в кінцеву. Може скластися ситуація, коли на деякій площині потрібно у певному порядку відвідати певну кількість точок, що є обов'язковими для відвідання.

Такі задачі дуже актуальні, наприклад, для побудови траєкторії безпілотного літального апарата (БПЛА), які широко використовуються в цивільних областях, наприклад, для вирішення завдань моніторингу поверхні Землі, спостереження за об'єктами транспортної інфраструктури. З бурхливим розвитком БПЛА, здатним рухатися в умовах складного рельєфу, завдання планування маршруту, що дозволяє уникнути зіткнень з перешкодами, актуальне для різних областей застосувань [1, 2].

Загалом, методи планування шляху за характером навколишнього середовища можна розділити на методи планування в статичному навколишньому середовищі [3] та методи планування в динамічному середовищі. Також можна відрізняти методи за повнотою інформації про навколишнє середовище: методи з повною інформацією (в такому випадку говорять про глобальне плануванні шляху) і методи з неповною інформацією. У другому випадку мова йде про знання параметрів середовища в безпосередній близькості від об'єкта, та про локальне планування шляху. В цих тезах пропонується алгоритм, що дотичен саме до останнього сімейства методів.

Нехай в деякій площині D існують точки $t_i, \ i = 1..n$, кожна з яких має свій пріоритет відвідування та координати $t_i\{x_i; \ y_i\}$. На площині D є множини точок $P_j, \ j = 1..m$, які описують периметри певних m геометричних фігур. Також є точка L, що пересувається по площині та має окіл з радіусом r.

Необхідно побудувати траєкторію руху точки L від точки t_1 до t_2, від точки t_2 до t_3, ..., від точки t_{n-1} до t_n, не чипляючи множин P_j з урахуванням окілу r.

Математичну складову методу ми опустимо через обмеження розмірів публікації.

Для прикладу візьмемо частину карти міської забудівлі, що відповідає центральному району великого міста (рис. 1). На рисунку можна побачити точки A і B, що відповідають початку та кінцю маршруту.

Як видно на рис. 1, не можна з'єднати початкову і кінцеву точки маршруту через перешкоди, які перебувають на шляху. Застосуємо алгоритм.

Розіб'ємо карту на великі, різні за площею сектора. Даний фрагмент

Рис. 1 – Приклад карти об'єкту.

фрагмент карти місцевості розділено на 15 секторів. Введемо наступні позначення:

S_n – окремий сектор на карті, n – порядковий номер сектора.

Також приймемо, що k_i – вершини графа маршруту, що будується в секторі S_n; $1 \leq i \leq 4$ – кількість виділених вершин, що будуються в секторі S_n. Дві з яких обов'язково стосуються контуру секторів S_n. Виділені вершини - це вершини за допомогою, яких ми будемо проходити сектор.

Вершину виділяємо тільки, якщо клітина не є «зайнятої». «Зайнята» клітина – це клітина, яка не була виділена раніше або не є перешкодою. В нашому прикладі перешкоди – це будинки, вони позначені блакитним кольором (рис. 2 а).

а) б) в)

Рис. 2 – а) приклад «зайнятих» вершин; б) приклад визначення кількості вершин фрагменту; в) приклад фрагменту з двома вершинами.

Кількість вершин графа шляху залежить від того, скільки секторів мають спільну сторону контуру (рис. 2 б), а також від перешкод. Спільний контур виділено червоною лінією. Фрагмент складається з 4 областей, з яких лише 2 сектора є «сусідами» лівого нижнього фрагмента. Тому наш фрагмент на рис. 2 матиме тільки дві вершини, через які розглядається можливість переміщення в сусідній сектор.

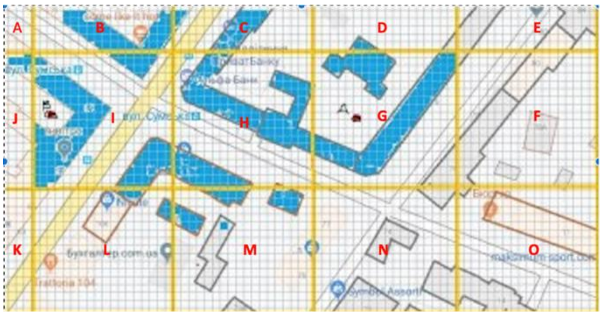
Червоними квадратами відзначені вершини фрагмента S_n, відстань між, якими мінімальна (рис. 2 в) При розставлені вершин графа, обов'язково вибираємо мінімальний по довжині шлях.

Отримали 2 вершини - даний сектор має чотирьох «сусідів», але перешкода не дає нам дістатися до трьох сторін.

Після отримання відрізків траєкторії у всіх сегментах формуємо маршрут руху цілком. Досліджуємо наш алгоритм. Позначимо сектора буквами, як показано на рис. 3 а).

Початкова точка А знаходиться в секторі G (рис. 3 б). Знаходимо вершини фрагмента G і шлях мінімальної довжини (рис. 3 б). Повторюємо ті ж дії в інших секторах і отримуємо загальний маршрут руху від початкової точки до кінцевої.

а) б)



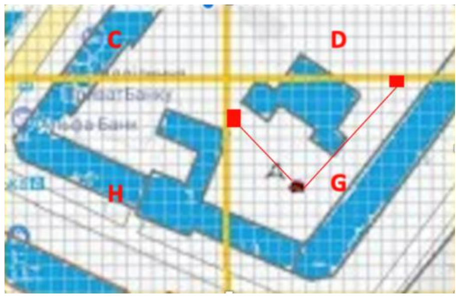


Рис. 3– а) приклад позначення секторів; б) Маршрут руху через сектор *G*.

Таким чином, представлено приклад роботи нового геометричного алгоритму, що може бути використано в програмному забезпеченні безпілотних літальних апаратів в певних умовах експлуатації.

Розроблений алгоритм вмістив у себе дві методики планування шляху, а саме алгоритм на графі та метод клітинної декомпозиції, поєднуючи їх ієрархічним підходом. Така стратегія дозволила розбити задачу побудови шляху на невеликі частини, що дозволило прискорити пошук шляху.

В якості наступного напряму роботи ми вирішили розробити новий метод або пристосувати один із існуючих методів для згладжування кінцевої траєкторії.

Список використаних джерел

1. Хацько Н.Е., Молчанов К.Д., Бойко Д.И. Разработка геометрического метода построения маршрута БПЛА для обхода препятствий. *Теоретичні та практичні дослідження молодих вчених* : тези доп. XIII Міжн. наук.-практ. конф. магістрантів та аспірантів. Харків, 2019. 19-22 листопада 2019р. С. 94-95.

2. Khatsko N. E., Ryabtseva O. O. Designing of trajectory of the movement of UAV with the avoidance of obstacles. Інформаційні технології: наука, техніка, технологія, освіта, здоров’я: зб. доповідей XXVI Міжнародної наук-практ. конф. Microcad-2018. Ч.1 Харків, 2017. 17-19 травня 2018р. С.72.

3. Евстигнеев В. А. Применение теории графов в программировании. Москва : Наука, 1985. 352с.