

## ОЦІНКА ЗАБРУДНЕННЯ ПОВІТРЯ З ВИКОРИСТАННЯМ БЕЗПІЛОТНОГО ЛІТАЛЬНОГО АПАРАТУ ЗА ДОПОМОГОЮ ГЕНЕТИЧНИХ АЛГОРИТМІВ

Вінницький національний технічний університет

### *Анотація*

У даній статті проведено аналіз методу використання безпілотних літальних апаратів (дронів) за допомогою генетичних алгоритмів у випадку оцінки забруднення повітря. Досліджено актуальність даної розробки відповідно до проблем забруднення атмосферного повітря. Проаналізовано переваги та недоліки даного методу у порівнянні з альтернативними методами. Проведено аналіз компонентів апаратної частини, обрано найбільш оптимальні для вирішення поставленої задачі. Наведено схему конструкції та керування безпілотного літального апарату. Обрано та наведено аргументацію вибору алгоритму для керування БпЛА. Поетапно описано алгоритм його виконання, наведено приклад, що реалізований за допомогою мови Java. Графічно зображено виконання генетичного алгоритму та наведено графіки швидкості виконання.

### **Ключові слова:**

Генетичні алгоритми, БпЛА, дрони, машинне навчання, екологічні проблеми, забруднення повітря, аналітична оцінка.

### *Abstract*

This article analyzes the method of using unmanned aerial vehicles (drones) using genetic algorithms in the case of air pollution assessment. The relevance of this development in accordance with the problems of atmospheric air pollution was investigated. The advantages and disadvantages of this method in comparison with alternative methods are analyzed. An analysis of the components of the hardware part was carried out, and the most optimal ones were chosen for solving the given task. The design and control scheme of the unmanned aerial vehicle is presented. The rationale for choosing an algorithm for UAV control is chosen and given. The algorithm for its implementation is described step by step, an example implemented using the Java language is given. The execution of the genetic algorithm is graphically depicted and the execution speed graphics are given.

### **Keywords:**

Genetic algorithms, UAV, drones, machine learning, ecological problems, air pollution, analytical estimation.

### **Вступ**

Забруднення повітря – головна екологічна причина передчасних смертей в Європі. Вона викликає ряд захворювань, включаючи серцево-судинні хвороби, легеневі та інші. Забруднення навколишнього атмосферного повітря стає причиною передчасної смерті 600 000 чоловік в регіоні за даними ЄЕК ООН щорічно. Ця проблема з кожним роком загострюється, оскільки посилюється антропогенний вплив на навколишнє середовище, і тому потрібно здійснювати постійний моніторинг за станом довкілля та давати йому екологічну оцінку. Визначення точного внеску окремих факторів у розвиток захворювання нерідко є досить важким завданням, яке ускладнюється значною кількістю обумовлених ними ефектів, багато з яких, до того ж, можуть зустрічатися серед населення і без впливу цих факторів.

Для вирішення поставленої задачі використовується велика кількість підходів. Зокрема, в Україні є дуже мало достовірної інформації про якість атмосферного повітря. Організація та методологія моніторингу забруднення не відповідають стандартам ЄС, так як не враховує концентрацію твердих часточок діаметром менше 2,5 та 10 мкм. Основним методом визначення концентрацій забрудників є відбір проб повітря на стаціонарних постах спостереження. Кількість постів визначається розміром міста і особливостями структури промисловості. Вона може коливатись від одного поста для міст з населенням, меншим за 50 тисяч мешканців, до двадцяти постів для міст-мільйонників. Є обов'язковими спостереження за концентраціями пилу, діоксиду сірки, оксиду вуглецю, діоксиду азоту, свинцю та його неорганічних сполук, формальдегіду та радіоактивних речовин. Цей метод має ряд недоліків, тому що він не передбачає моніторинг ситуації у реальному часі. Також покривати всю територію пунктами відбору проб нераціонально. У ЄС для оцінки якості повітря використовуються різні методи: фіксовані вимірювання з відбором проб, індикативні вимірювання та моделювання. Фіксовані вимірювання дають найбільш точне уявлення про вміст забруднюючих речовин в атмосфері, оскільки передбачають безпосередній аналіз проб повітря. Такі вимірювання не є достатньо точними. Однак, основною їхньою перевагою є дешевизна та можливість отримати результати по всій території, а не в окремих точках [1].

Враховуючи недоліки вищеперерахованих методів на допомогу приходять сучасні технології та алгоритми машинного навчання які допомагають швидко, ефективно та точно оцінити проблему. Останні дослідження все більше спираються на технології машинного навчання та доводять їх ефективність у вирішенні наведеної проблеми. Зокрема, було досліджено та спроектовано БпЛА на базі алгоритму ройового інтелекту на основі живої природи [2]. Описана конструкція апарату є подібною до згаданої у даній статті, проте компоненти наведені у цій роботі є сучаснішими. Алгоритм ройового інтелекту розглянуто та виконано порівняння із пропонуваним генетичним. Також, важливим дослідженням є робота що описує створення інформаційної системи моніторингу забруднення атмосферного повітря міста на основі технології «Інтернет речей» [3]. Ця робота пропонує програмно-технічне рішення базуючись на загальній концепції мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані давачі, а також програмного забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом і комп'ютерними системами в автоматичному режимі, за допомогою використання стандартних протоколів зв'язку. Стаття надає наочне розуміння способу роботи такої системи.

Застосовуючи підхід машинного навчання, у цій статті пропонується використання БпЛА (дронів), оснащених стандартними датчиками, для виконання завдань моніторингу забруднення повітря.

### **Вибір методу для моніторингу якості повітря**

Галузь штучного інтелекту одна із найбільш популярних та розвинених на сьогоднішній день. Алгоритми ШІ стрімко розвиваються, неодноразово доводячи свою ефективність, що дозволяє впроваджувати їх майже у будь яку сферу застосування – від медицини до промисловості. Для точного оцінювання якості атмосферного середовища буде доцільним звернутись до сучасних технологій, безпосередньо до тих, які базуються на алгоритмах машинного навчання.

Прийнято вважати, що найбільший вплив на якість повітря впливає промисловість, центри якої зазвичай знаходяться у міській місцевості. Проте, моніторинг якості повітря актуальний не лише для людей, які проживають у містах, але й тому, що він безпосередньо впливає на сільськогосподарські культури та різних тварин у сільській місцевості. Таким чином, для таких середовищ слід шукати різні рішення для вимірювання якості повітря. Більшість методів, які використовуються для відстеження забруднення повітря у великих містах, покладаються на стаціонарні станції моніторингу. Проте, дані станції перевершуються використанням підходу краудсенсингу - це метод, при якому велика група людей має мобільні пристрої, здатні розпізнавати та обчислювати, спільно обмінюватися даними та отримувати інформацію для вимірювання, зіставлення, аналізу, оцінки або виведення (прогнозування) будь-яких процесів. Цей метод працює у випадку густонаселеної місцевості, тому що вимагає залучення великої кількості девайсів.

Як оптимальну та ефективну альтернативу можна розглядати використання мобільних дронів. Мобільні датчики здатні покривати територію, яка рівна території моніторингу стаціонарних станцій, при цьому не потребує великої кількості технічних засобів для їх використання, як у випадку з краудсенсингом. Для доведення універсальності методу потрібно враховувати особливості територій, на яких буде здійснюватись моніторинг якості повітря:

1. Міська густонаселена територія, з великою кількістю транспорту, промисловою інфраструктурою, велика кількість будинків та споруд.

2. Сільська територія із низькою щільністю населення, аграрними ділянками та комплексами сільськогосподарської промисловості.
3. Територія з специфічним ландшафтом, важкодоступними ділянками, нестабільним ґрунтом (наприклад гірська місцевість).
4. Водойми.
5. Небезпечні для життя та здоров'я людини місця: певні місця на промислових об'єктах, закриті території (наприклад Чорнобильська зона відчуження), території, що потерпіли від природних катаклізмів.

Враховуючи усі вищеперечислені фактори, буде універсальним та зручним методом використання безпілотних літальних апаратів (дронів), так як їх робота не залежить від певного ландшафту, завдяки пересуванню у повітрі, і також надає безпечний доступ до ділянок, які можуть бути небезпечними для життя та здоров'я людини.

### **Апаратне забезпечення БПЛА**

Для конструкції такого дрона використовуються одноплатні комп'ютери, так як система є дуже компактною. Для прикладу можна розглянути Jetson Nano або Raspberry Pi4 – доволі популярне обладнання, яке має достатньо ресурсів для вирішення поставленої задачі, невелике за розміром та адекватне за ціною. Щоб вибрати комп'ютер для даного проекту потрібно розуміти розбіжності між вказаними вище технологіями.

Raspberry Pi вважається більш класичним варіантом та використовується у школах та університетах. Характеристики Raspberry Pi4[4]:

1. Система на чіпі Broadcom BCM2711 і працює на 4-ядерному 64-розрядному процесорі ARM Cortex-A72 з тактовою частотою 1,5 ГГц.
2. Має чотири версії оперативної пам'яті: 1 ГБ, 2 ГБ, 4 ГБ і 8 ГБ LPDDR4-2400 SDRAM.
3. Завдяки двом портам micro-HDMI Raspberry Pi 4 пропонує 4K 60FPS із функцією двох екранів.
4. Оснащено двома портами USB 3.0, двома портами USB 2.0, портом USB C для живлення, 3,5-мм аналоговим аудіо-відеороз'ємом, двома портами micro-HDMI, послідовним інтерфейсом камери (CSI) і послідовним інтерфейсом дисплею (DSI).
5. Підтримує Gigabit Ethernet, Wi-Fi і Bluetooth.
6. ОС Raspberry Pi (раніше відома як Raspbian), Ubuntu, OSMC, RetroPie.

Jetson Nano – одноплатний комп'ютер розроблений компанією NVIDIA. Характеристики Jetson Nano[5]:

1. Працює на 4-ядерному процесорі ARM Cortex-A57 64-bit @ 1,42 ГГц.
2. Поставляється у версії LPDDR4 4 ГБ або альтернативі 2 ГБ.
3. Jetson Nano 4 ГБ підтримує HDMI 2.0 і DisplayPort (eDP 1.4). Версія 2 ГБ підтримує лише HDMI 2.0.
4. Має чотири порти USB 3.0, один порт USB 2.0 Micro-B, дві лінії MIPI CSI-2 DPHY, порт HDMI 2.0 і DisplayPort. Модель 2 ГБ має один порт USB 3.0, два порти USB 2.0, один порт USB 2.0 Micro-B, лінію MIPI CSI-2 D-PHY і порт HDMI 2.0.
5. Підтримує Gigabit Ethernet і M.2 Key E для підтримки Wi-Fi, а версія 2 ГБ включає бездротовий адаптер USB 802.11ac.
6. Офіційною операційною системою для Jetson Nano є Linux4Tegra на базі Ubuntu 18.04. ОС доступна через образ SD-карти, що входить у комплект, який призначений для роботи обладнання NVIDIA.

Також є відмінності у ціні – Raspberry Pi дешевша за Jetson Nano. Принципова різниця відчувається у можливостях графічної обробки. Порівняно з Raspberry Pi4 Jetson Nano має набагато потужніший графічний процесор. NVIDIA Jetson Nano має 128-ядерний графічний процесор Maxwell із частотою 921 МГц. Це робить його більш придатним для програм штучного інтелекту та машинного навчання, що може бути особливою перевагою, залежно від галузі використання. В загальному по можливостях для конструкції дрону, що буде проводити моніторинг якості повітря підходять обидва комп'ютери[6].

У даній статті буде розглянуто конструкцію дрона на основі Raspberry Pi4, так як його можливостей достатньо для вирішення заданої задачі і він має значну перевагу у ціні. Нижче наведено схему динамічного керування БПЛА шляхом підключення модуля керування БПЛА до Raspberry Pi і підключення його до набору датчиків забруднення через аналоговий перетворювач (Рисунок 1).

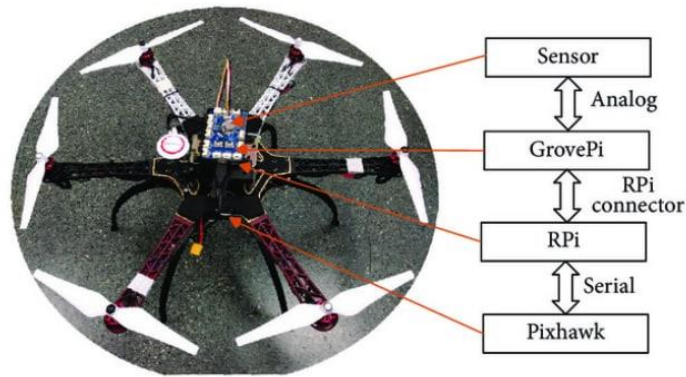


Рисунок 1 – Схема конструкції БПЛА

БПЛА керується за допомогою автопілота Pixhawk, який контролює його фізичне функціонування. Raspberry Pi монтується на шасі БПЛА та підключається до Pixhawk через послідовний порт. Датчики підключаються до Raspberry Pi за допомогою Grove Raspberry Hat (GrovePi), що дозволяє легко підключати різні типи датчиків COTS. Зокрема, на рисунку 1 зображені такі компоненти[7]:

1. Pixhawk Autopilot: високоефективний модуль керування польотом, який підходить для кількох типів автономних транспортних засобів, включаючи мультиротори, гелікоптери, автомобілі, човни та літальні апарати з нерухомим крилом. Програмне забезпечення для автопілота, яке включає в себе операційну систему реального часу (RTOS) із середовищем у стилі POSIX для керування дроном.
2. Одноплатний комп'ютер Raspberry Pi4, який було описано вище.
3. GrovePi: плата розширення, яка дозволяє легко підключати кілька аналогових/цифрових портів Grove до Raspberry Pi. Він має кілька портів Grove: сім цифрових портів, три аналогових порти, три порти I2C, один послідовний порт для GrovePi і послідовний роз'єм для Raspberry Pi.
4. Датчики Grove: датчики, які використовують стандартизований роз'єм Grove, що забезпечує легке підключення до різних плат, як GrovePi. Існує кілька датчиків навколишнього середовища COTS, які реагують на CO<sub>2</sub>, CO або алкоголь.

На рисунку 2 наведено схему керування БПЛА за замкнутим контуром.



Рисунок 2 – Схема керування БПЛА

### Алгоритм виконання поставленої задачі

Для моніторингу забруднення повітря потрібно запрограмувати автоматичні процедури для отримання даних про забрудненні території і надсилати їх на станцію контролю якості повітря в режимі реального часу. У даній статті описано роботу БПЛА, що обробляє дані перед передачею. Дані для тестування беруться

безпосередньо з сенсорів. Цілі проекту включають речовини CO, CO<sub>2</sub>, CH<sub>4</sub>, аміак та концентрації твердих часток які за діаметром є менше 2,5 та 10 мм (відповідно до стандартів ЄС).

Щоб вирішити дану задачу безпілотний літальний апарат має використовувати алгоритм машинного навчання. Метою є досягнення усієї заданої території, щоб забезпечити якість проб за найбільш оптимальним маршрутом, аби виконати задачу за мінімальний час. Побудований оптимальний маршрут буде становити собою граф через вершини якого проходить БПЛА та брати проби якості повітря. Можна припустити, що така задача є однією із видів задачі комівояжера, де умовою є знаходження найбільш вигідного маршруту через усі точки – вершини графа. Для цього розглянемо два класичних алгоритми оптимізації: генетичний алгоритм та алгоритм оптимізації рою частинок.

Основна ідея алгоритму рою частинок полягає в колективній поведінці деяких «агентів», які можуть взаємодіяти один з одним та з навколишнім середовищем для вирішення певного завдання. Він моделює багатоагентну систему, де агенти-частинки рухаються до оптимальних рішень, обмінюючись інформацією із сусідами. Поточний стан частки характеризується координатами у просторі рішень (тобто, власне, пов'язаним із ними рішенням), а також вектором швидкості переміщення. Обидва параметри вибираються випадковим чином на етапі ініціалізації. Крім того, кожна частка зберігає координати кращого із знайдених їй рішень, а також найкраще з пройдених усіма частинками рішень – цим імітується миттєвий обмін інформацією між птахами[8].

Щоб оптимізувати дані використовується функція допасованості. Це особливий тип цільової функції, який застосовують як порівняльний показник якості для підбивання підсумку того, наскільки близьким є задане конструктивне рішення до досягнення поставлених цілей. Блок схема даного алгоритму наведена нижче на рисунку 3:

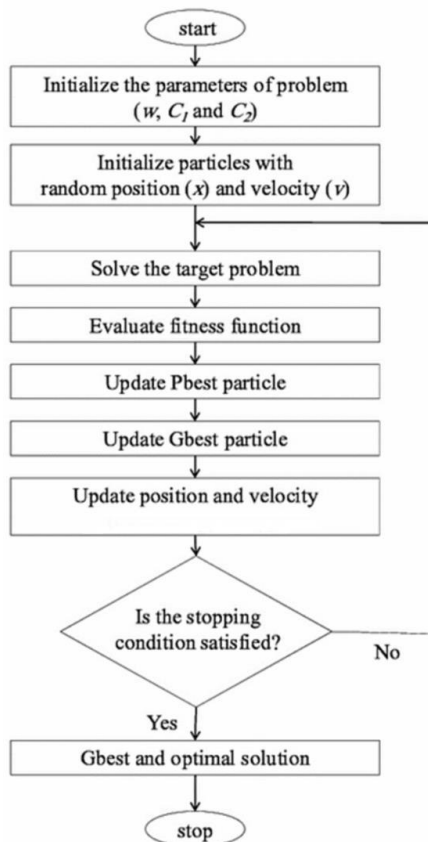


Рисунок 3 – Блок схема алгоритму рою частинок

Головною перевагою такого методу є простота програмної реалізації та висока ефективність при невеликій кількості можливих маршрутів. При збільшенні кількості варіантів можливих маршрутів точність такого

алгоритму падає, так як збільшується можливість ранньої збіжності у якійсь локальній точці, без можливості розгляду альтернатив.

Для вирішення поставленої задачі було обрано генетичний алгоритм. Генетичний алгоритм — це алгоритм, який зазвичай використовується для вирішення задачі комівояжера у дослідженнях планування траєкторії БпЛА[9]. Це випадковий глобальний алгоритм пошуку, запропонований відповідно до теорії «виживання найприспособанішого» в теорії еволюції, який використовує дані у вигляді рядків хромосомних даних. Відповідно до процесу біологічної еволюції в природі, вибираються оптимальні рішення для генетичної варіації; тобто ітераційна оптимізація. На відміну від алгоритму рою частинок наведений алгоритм потребує більше часу на виконання, але має більшу точність.

Раніше генетичний алгоритм не застосовувався на БпЛА через апаратне обмеження, але з появою таких комп'ютерів як RPi4 чи Jetson Nano, оглянутих вище може виконуватись для вирішення поставленої задачі.

Дані, над якими проводяться операції у генетичному алгоритмі:

1. Хромосома – послідовність вершин, які охоплюють маршрут.
2. Популяція, тобто множини хромосом (кількості маршрутів).
3. Особа – набір хромосом, які задовольняють рішення.

Функції допасованості застосовують також в генетичному програмуванні та генетичних алгоритмах, аби скеровувати симуляції до оптимальних конструктивних рішень.

Далі, можна виділити наступні етапи генетичного алгоритму:

1. Створення початкової популяції.
  2. Обчислення функції допасованості для осіб популяції (оцінювання).
  3. Повторювання до виконання критерію зупинки алгоритму. Таким критерієм може бути: знаходження глобального, або надоптимального вирішення; вичерпання числа поколінь, що відпущені на еволюцію; вичерпання часу, відпущеного на еволюцію. У випадку задачі описаної у даній статті таким критерієм буде знаходження надоптимального вирішення.
  4. Вибір індивідів із поточної популяції (селекція)
  5. Схрещення або/та мутація.
    - Етапи схрещення: генерація точки розриву, формування першого нащадка – де беруться гени першого батька до точки розриву і гени другого батька після точки розриву. Якщо залишаються незаповнені гени, додаються неуспадковані гени після точки розриву від першого батька. Аналогічно з другим нащадком, тільки цього разу спочатку використовуються гени другого батька до точки розриву.
    - Мутація – генерується випадкове число від 0 до 100. Якщо число менше заданого відсотку мутацій, відбувається мутація. Обираються 2 випадкових гени, та міняються місцями.
  6. Обчислення функції допасованості для всіх осіб. Відкидаються найбільш неоптимізовані рішення які виникли на даному кроці. Розмір популяції залишається однаковим.
  7. Формування нового покоління.
- Блок-схема генетичного алгоритму наведена на рисунку 4.

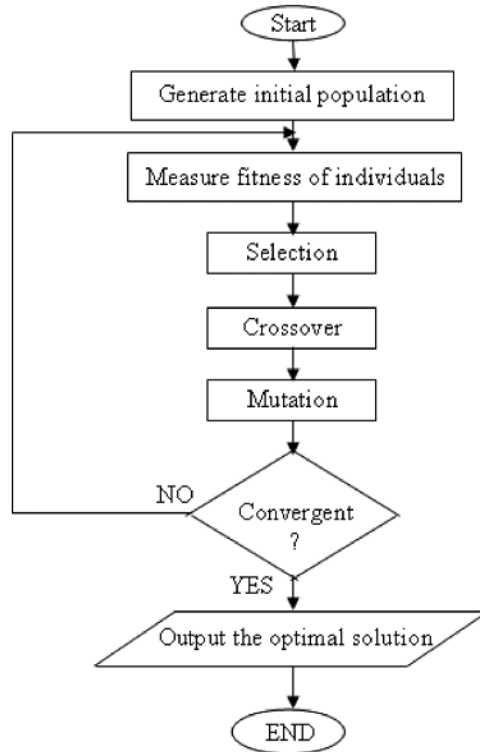


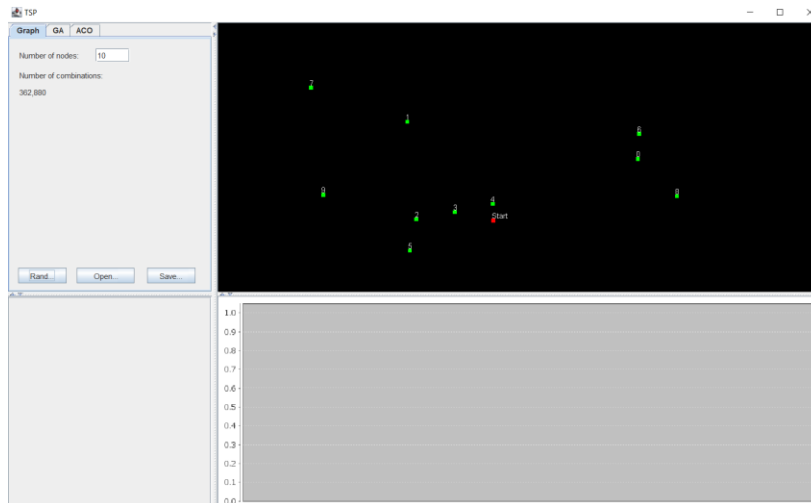
Рисунок 4 – Блок-схема генетичного алгоритму

Для того, щоб досягти вибірки повного охоплення регіону, під час планування шляху спочатку генерується впорядкована послідовність усіх субрегіонів які потрібно дослідити, що є декомпозицією досліджуваної території на менші її частини. Ця послідовність представляє порядок доступу до цільових субрегіонів. На основі цієї послідовності генерується траєкторія покриття, і БПЛА літає відповідно до траєкторії та послідовно відвідує кожен субрегіон, що представляє інтерес, для виконання операцій аналізу якості повітря[10]

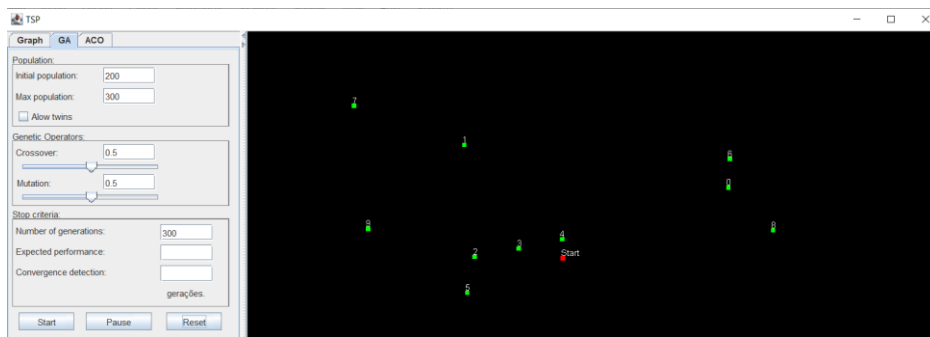
#### **Експериментальний запуск алгоритму**

Нижче наведено експеримент за допомогою програми генетичного алгоритму, що вирішує задачу комівояжера. Програму реалізовано на мові програмування Java. Дану мову було обрано через її швидкість, безпечність та кросплатформеність – через це програму може бути запущено й на інших пристроях без додаткових проблем.

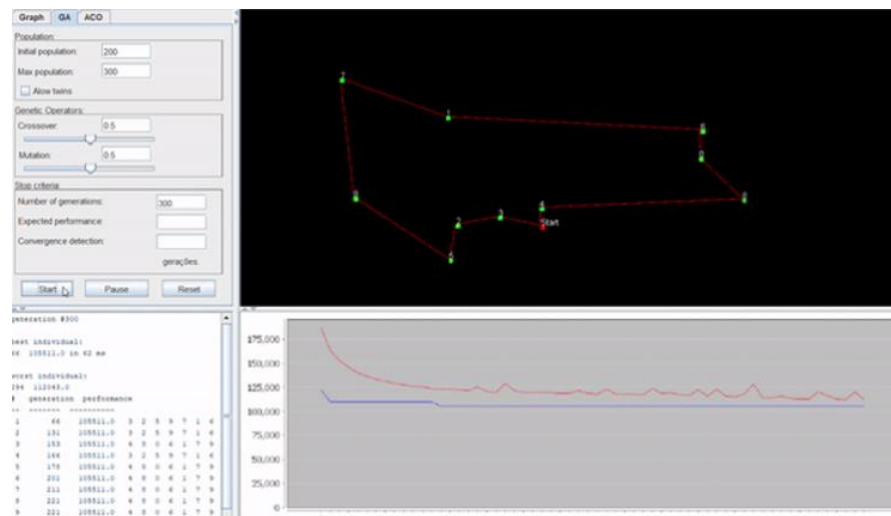
1. Задано кількість вершин:



2. Вкладка GA містить опції задання критеріїв початкової популяції, максимального числа популяції, опцію дозволу «близнюків», тобто існування однакових осіб, регулювання схрещування та мутації. Також за потреби можна вказати критерії зупинки, такі як максимально допустима кількість поколінь, очікувана продуктивність чи певну кількість збіжностей.



3. Виконання програми пошуку оптимального шляху.



4. Результати:



```

generation #300

best individual:
66 105511.0 in 62 ms

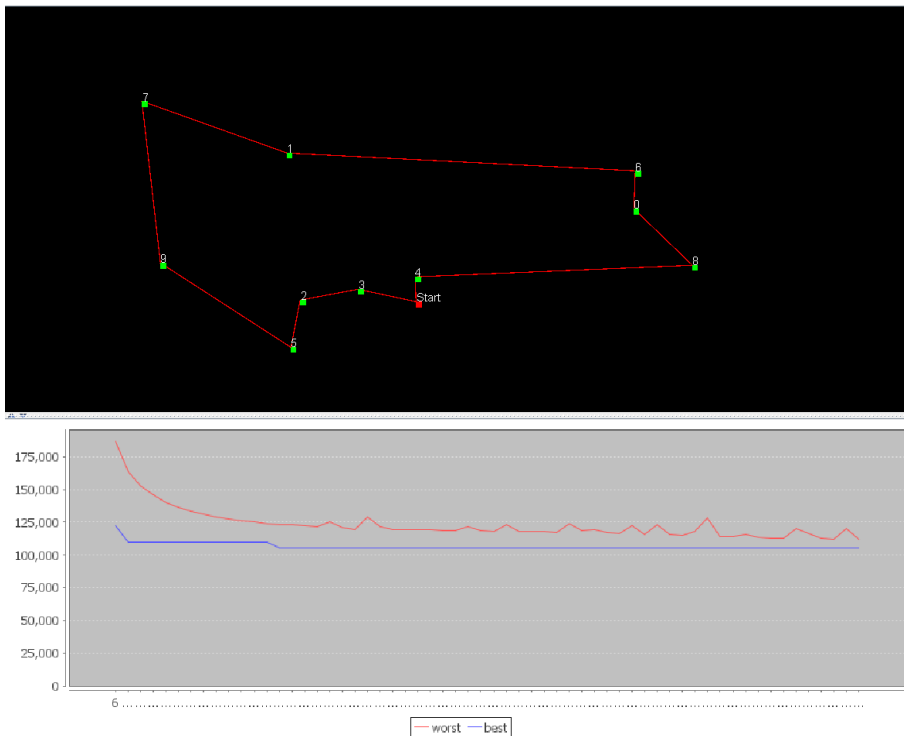
worst individual:
294 112043.0

#  generation  performance
--  -
1      66      105511.0   3 2 5 9 7 1 6 0
2      131     105511.0   3 2 5 9 7 1 6 0
3      153     105511.0   4 8 0 6 1 7 9 5
4      166     105511.0   3 2 5 9 7 1 6 0
5      178     105511.0   4 8 0 6 1 7 9 5
6      201     105511.0   4 8 0 6 1 7 9 5
7      211     105511.0   4 8 0 6 1 7 9 5
8      221     105511.0   4 8 0 6 1 7 9 5
9      221     105511.0   4 8 0 6 1 7 9 5
10     222     105511.0   4 8 0 6 1 7 9 5

elapsed: 284 ms

```

5. Вигляд графу: наведений нижче червоний графік зображує найгірше покоління, синій – найвдаліше.



### Висновок

У даній статті було розглянуто застосування безпілотного літального апарату за допомогою генетичного алгоритму у аспекті оцінки забруднення атмосферного повітря. Досліджено актуальність наведеної проблеми. Запропоновано апаратну конструкцію БПЛА використовуючи сучасні технології, враховуючи продуктивність та вартість компонентів, їх актуальність для вирішення поставленого завдання на основі одноплатного комп'ютера Raspberry Pi4.

Проведено аналіз класичних алгоритмів оптимізації. Обґрунтовано вибір найбільш вигідного за ресурсом часу та якості алгоритм для оптимального вирішення задачі. Було здійснено програмну реалізацію обраного алгоритму за допомогою мови програмування Java.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Як Україна вимірює забруднення повітря? [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://ua-energy.org/uk/posts/yak-ukraina-vymiriuiie-zabrudnennia-povitria> – Назва з екрана.
2. Alvear, Oscar & Calafate, Carlos & Zema, Nicola & Natalizio, Enrico & Hernandez-Orallo, Enrique & Cano, Juan-Carlos & Manzoni, Pietro. (2018). PdUC-D: A Discretized UAV Guidance System for Air Pollution Monitoring Tasks. 10.1007/978-3-319-76111-4\_38.
3. Мокін, В. Б., Собко, Б. Ю., Драгований, М. В., Крижановський, Є. М., & Горячев, Г. В. (2017). СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ ЗАБРУДНЕННЯ АТМОСФЕРНОГО ПОВІТРЯ МІСТА НА ОСНОВІ ТЕХНОЛОГІЇ «ІНТЕРНЕТ РЕЧЕЙ». Вісник Вінницького політехнічного інституту, (3), 49–58. вилучено із <https://visnyk.vntu.edu.ua/index.php/visnyk/article/view/2055>
4. Jetson Nano vs Raspberry Pi 4: The Differences [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://all3dp.com/2/raspberry-pi-vs-jetson-nano-differences/> – Назва з екрана.
5. Raspberry Pi 4 Tech Specs [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/> – Назва з екрана.
6. Jetson Nano – NVIDIA Developer [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://developer.nvidia.com/embedded/jetson-nano> – Назва з екрана.
7. Using UAV-Based Systems to Monitor Air Pollution in Areas with Poor Accessibility [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://www.researchgate.net/publication/318971747> – Назва з екрана.
8. Swarm Intelligence Algorithm [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://www.sciencedirect.com/topics/computer-science/swarm-intelligence-algorithm> – Назва з екрана.
9. Evolution of a salesman: A complete genetic algorithm tutorial [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://towardsdatascience.com/evolution-of-a-salesman-a-complete-genetic-algorithm-tutorial-for-python-6fe5d2b3ca35> – Назва з екрана.
10. TSP problem using GA algorithm [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://sourceforge.net/projects/tsp-problem-ga-aco-comparisson/> – Назва з екрана.

**Науковий керівник – Кулик Ярослав Анатолійович** – к.т.н., доцент кафедри Автоматизації та інтелектуальних інформаційних технологій, Факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: [kulyk.y.a@vntu.edu.ua](mailto:kulyk.y.a@vntu.edu.ua)

**Барановська Анастасія Юрївна** – студентка групи ІІСТ-196, Факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: [xktsumst@gmail.com](mailto:xktsumst@gmail.com)

***Лешок Максим Андрійович*** – студент групи ІІСТ-19б, Факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: [max6leshok@gmail.com](mailto:max6leshok@gmail.com)

***Supervisor – Kulyk Yaroslav Anatolyovich*** – Associate Professor of department of Automatization and Intellectual Informational Technologies, Faculty of Intellectual Informational Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: [kulyk.y.a@vntu.edu.ua](mailto:kulyk.y.a@vntu.edu.ua)

***Baranovska Anastasiia Y.*** – student of ІІСТ-19b, Faculty of Intellectual Informational Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: [xktsumst@gmail.com](mailto:xktsumst@gmail.com)

***Leshok Maksym Andriyovych*** – student of ІІСТ-19b, Faculty of Intellectual Informational Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: [max6leshok@gmail.com](mailto:max6leshok@gmail.com)