

Національний університет «Полтавська політехніка імені Юрія  
Кондратюка»

(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних технологій та  
робототехніки

(повне найменування інституту, назва факультету (відділення))

Кафедра автоматики, електроніки та телекомунікацій

(повна назва кафедри (предметної, циклової комісії))

## Пояснювальна записка

до кваліфікаційної роботи

магістр

(рівень вищої освіти)

на тему «Інтеграція геоінформаційних систем та методів просторового  
аналізу для планування та управління телекомунікаційними мережами»

Виконав: студент 2 курсу, групи 601-ТТ  
спеціальності 172 «Електронні комунікації та радіотехніка»

(шифр і назва напрямку підготовки, спеціальності)

Сухоробрий О.В.

(прізвище та ініціали)

Керівник Косенко В.В.

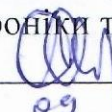
(прізвище та ініціали)

Рецензент Шефер О.В.

(прізвище та ініціали)

Полтава - 2025 рік

Національний університет «Полтавська політехніка імені Юрія Кондратюка»  
Інститут Навчально-науковий інститут інформаційних технологій та  
робототехніки  
Кафедра Автоматики, електроніки та телекомунікацій  
Ступінь вищої освіти Магістр  
Спеціальність 172 «Електронні комунікації та радіотехніка»

ЗАТВЕРДЖУЮ  
Завідувач кафедри автоматки,  
електроніки та телекомунікацій  
 О.В. Шефер  
“ 15 ” “ 09 ” 2025 р.

## ЗАВДАННЯ НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

**Сухореброму Олександровичу Володимировичу**

1. Тема проекту (роботи) «Інтеграція геоінформаційних систем та методів просторового аналізу для планування та управління телекомунікаційними мережами»

керівник проекту (роботи) Косенко Віктор Васильович, д.т.н., професор  
затверджена наказом вищого навчального закладу від \_\_\_\_\_ .2025 року  
№ \_\_\_\_\_

2. Строк подання студентом проекту (роботи) 22.12.2025 р.

3. Вихідні дані до проекту (роботи) Картографічні матеріали місцевості, топологічні дані GPON-мережі та координати об'єктів телекомунікаційної інфраструктури. Програмне забезпечення QGIS для просторового аналізу та візуалізації. Дані щодо зон покриття FPV-дронів і супутникових систем зв'язку (Starlink) для моделювання та порівняння ефективності.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Актуальність та мета інтеграції геоінформаційних систем в телекомунікації. Аналіз сучасних GIS-технології та їх застосування в телекомунікаціях. Розробка методики моделювання мереж за допомогою GIS. Реалізувати приклади для GPON, FPV, 5G та супутникових систем. Оцінити ефективність моделювання мереж за допомогою GIS. Висновки. Оформлення кваліфікаційної роботи та підготовка графічних матеріалів. Формування додатків.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових плакатів):

- 1) Актуальність, мета, об'єкт та предмет дослідження, задачі
- 2) Інструментарій розробки
- 3) Методологія: Геометричний підхід

- 4) GPON: Автоматизація топології мережі
  - 5) GPON: Автоматизація через кластеризацію
  - 6) FPV: Виклик 3-D навігації
  - 7) FPV: Побудова безпечного маршруту
  - 8) Starlink: Подолання «міських каньйонів»
  - 9) Starlink: Моделювання перешкод сигнали
  - 10) Мережі 5G mmWave: Моделювання покриття
  - 11) Сценарний аналіз мережі 5G
  - 12) Практичне значення та досягнення
6. Дата видачі завдання 01.10.2025 р.

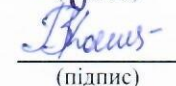
### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської роботи	Термін виконання етапів роботи		Примітка (плакати)
		Дата	Відсоток	
1	Актуальність та мета інтеграції геоінформаційних систем в телекомунікації	07.10.25	15%	Пл. 1
2	Аналіз сучасних GIS-технології та їх застосування в телекомунікаціях	21.10.25	30%	Пл. 2
3	Розробка методики моделювання мереж за допомогою GIS	04.11.25	45%	Пл. 3
4	Реалізувати приклади для GPON, FPV, 5G та супутникових систем	18.11.25	I 60%	Пл. 4,5,6,7,8,9,10
5	Оцінити ефективність моделювання мереж за допомогою GIS	09.12.25	80%	Пл. 11
6	Висновки. Оформлення кваліфікаційної роботи та підготовка графічних матеріалів. Формування додатків.	22.12.25	II 100%	Пл. 12

Магістрант

Керівник роботи

  
(підпис)

  
(підпис)

Сухоробрий О.В.  
(прізвище та ініціали)

Косенко В.В.  
(прізвище та ініціали)

## РЕФЕРАТ

кваліфікаційної роботи магістра

«Інтеграція геоінформаційних технологій та алгоритмів просторового аналізу для моделювання та планування телекомунікаційних мереж»

Робота містить 87 с., 29 рисунків, 12 формул, 4 додатки, 22 джерела.

Ключові слова: GPON, FPV, Starlink, 5G, GIS, QGIS, DSM, Viewshed, LoS, мережева аналітика, моделювання покриття, оптичні мережі, дистанційне зондування, радіотінь, алгоритм Дейкстри.

Предметом дослідження кваліфікаційної роботи є процес моделювання телекомунікаційних мереж різних типів (оптичних, бездротових, супутникових) засобами геоінформаційних технологій.

Метою роботи є розроблення комплексної методики моделювання та аналізу телекомунікаційних мереж за допомогою алгоритмів геопросторової аналітики, цифрових моделей поверхні та спеціалізованих скриптів у Python/QGIS, що дозволяє автоматизувати проектування та оцінку ефективності сучасних систем зв'язку.

Розроблена методика дозволяє уніфіковано застосовувати інструменти GIS для різних типів мереж, забезпечуючи точну просторову оцінку покриття, оптимізацію розташування обладнання та прогнозування властивостей каналів зв'язку.

## ABSTRACT

master's thesis

«Integration of Geoinformation Technologies and Spatial Analysis Algorithms  
for Modeling and Planning Telecommunication Networks»

The thesis contains 87 pages, 29 figures, 12 formulas, 4 appendices, and 22 references.

Keywords: GPON, FPV, Starlink, 5G, GIS, QGIS, DSM, Viewshed, LoS, network analytics, coverage modeling, optical networks, remote sensing, radio shadow, Dijkstra's algorithm.

The subject of the research is the process of modeling various types of telecommunication networks (optical, wireless, satellite) utilizing geoinformation technologies.

The objective of the thesis is to develop a comprehensive methodology for modeling and analyzing telecommunication networks using geospatial analytics algorithms, Digital Surface Models (DSM), and specialized Python/QGIS scripts. This approach enables the automation of the design process and the efficiency assessment of modern communication systems.

The developed methodology facilitates the unified application of GIS tools across different network types, ensuring accurate spatial coverage assessment, optimization of equipment placement, and prediction of communication channel properties.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

GIS (Geographic Information System) – геоінформаційна система, інструмент для роботи з просторовими даними.

GPON (Gigabit Passive Optical Network) – пасивна оптична мережа з пропускнуою здатністю до 2.5 Гбіт/с.

OLT (Optical Line Terminal) – головний оптичний термінал у GPON-мережі.

ONT/ONU (Optical Network Terminal / Unit) – абонентський оптичний термінал.

Splitter – пасивний оптичний подільник сигналу.

LoS (Line-of-Sight) – пряма видимість між передавачем та приймачем.

NLoS (Non-Line-of-Sight) – відсутність прямої видимості.

DSM (Digital Surface Model) – цифрова модель поверхні, що враховує будівлі та об'єкти.

DTM (Digital Terrain Model) – цифрова модель рельєфу без забудови.

Viewshed – алгоритм просторового аналізу видимості в GIS.

FSPL (Free Space Path Loss) – втрати у вільному просторі.

FPV (First Person View) – передача відео з дрона в реальному часі.

VTX (Video Transmitter) – відеопередавач FPV системи.

VRX (Video Receiver) – відеоприймач FPV системи.

Starlink – супутникова система зв'язку низької орбіти (LEO).

5G NR (New Radio) – п'яте покоління мобільного радіозв'язку.

mmWave (Millimeter Wave) – міліметрові хвилі (24–100 ГГц), використовуються в 5G.

RIS (Reconfigurable Intelligent Surface) – активні відбиваючі панелі для формування радіосигналу.

MIMO (Multiple Input Multiple Output) – багатоканальна антена для підсилення сигналу.

## ЗМІСТ

ВСТУП .....	9
РОЗДІЛ 1. Аналітична частина .....	11
1.1 Сучасні GIS-технології в телекомунікаціях .....	11
1.1.1 Функціональні можливості GIS для телекомунікацій .....	11
1.2 Методи просторового аналізу для телекомунікаційних мереж .....	12
1.2.1 Буферний аналіз та визначення зон покриття.....	12
1.2.2 Мережевий аналіз для оптимізації інфраструктури.....	14
1.3 Методи просторового аналізу для телекомунікаційних мереж .....	15
1.3.1 Компоненти GPON-мережі та їх просторове представлення.....	15
1.3.2 Моделювання оптичного бюджету потужності.....	15
1.3.3 FPV-системи та моделювання радіопокриття.....	16
1.3.4 Супутниковий зв'язок Starlink та аналіз доступності.....	17
1.3.5 Особливості просторового планування мереж 5G .....	18
1.4 Інтеграція GIS з системами управління мережами .....	19
1.4.1 Архітектура інтеграції GIS з OSS/BSS .....	19
1.4.2 Системи моніторингу та їх інтеграція з GIS .....	20
РОЗДІЛ 2. Методика моделювання телекомунікаційних мереж з використанням GIS.....	22
2.1 Загальні принципи побудови методики моделювання .....	22
2.1.1 Використання цифрових моделей рельєфу та поверхні у моделюванні телекомунікаційних систем.....	24
2.1.2 Автоматизація просторового моделювання засобами Python та QGIS25	
2.2 Методика моделювання GPON-мереж .....	26
2.2.1 Теоретичні засади моделювання GPON у контексті GIS.....	26

2.2.2	Геометричне моделювання GPON-мережі .....	27
2.2.3	Просторовий аналіз мережі та контроль довжин сегментів .....	29
2.3	Методика моделювання FPV-систем .....	30
2.3.1	Теоретичні основи моделювання FPV-зв'язку в контексті GIS .....	30
2.3.2	Побудова растрової моделі видимості FPV за допомогою GDAL Viewshed .....	31
2.3.3	Аналіз маршруту FPV-дрона в контексті моделі видимості .....	32
2.3.4	Генерація рекомендацій щодо встановлення ретрансляторів .....	33
2.3.5	Переваги та обмеження застосованої методики .....	33
2.4	Методика моделювання системи супутникового зв'язку Starlink .....	35
2.4.1	Геометричний підхід до моделювання доступності супутникового сигналу .....	35
2.4.2	Формування профілів горизонту за цифровою моделлю поверхні ...	35
2.4.3	Побудова карти доступності супутникового сигналу .....	36
2.4.4	Практичні аспекти встановлення терміналу Starlink у контексті результатів моделювання .....	37
2.4.5	Обмеження та можливості удосконалення .....	38
2.5	Методика моделювання покриття мереж 5G .....	38
2.5.1	Геометричні основи моделювання зон радіопокриття .....	38
2.5.2	Побудова буферних зон покриття базових станцій .....	39
2.5.3	Об'єднання зон покриття кількох базових станцій .....	40
2.5.4	Аналіз на основі геометричної моделі покриття .....	41
2.5.5	Обмеження та перспективи моделі .....	41
РОЗДІЛ 3. Практична реалізація GIS-моделювання та оптимізації мережі GPON .....		43
3.1	Вступ до практичної частини дослідження .....	43

3.2	Вибір та обґрунтування інструментарію дослідження .....	44
3.3	Підготовка та попередня обробка вихідних геопросторових даних ....	45
3.4	Формування точкової моделі абонентів .....	47
3.5	Побудова абонентських кабельних ліній .....	48
3.6	Формування бази даних GPON та експорт результатів .....	48
3.7	Візуалізація моделі GPON у середовищі QGIS.....	50
3.8	Аналіз отриманих результатів та оцінка ефективності.....	51
3.9	Просторові обмеження технології та їх вплив на модель.....	53
3.10	Перевірка протяжності абонентських підключень.....	54
3.11	Інтеграція моделі у подальше проектування мережі .....	55
РОЗДІЛ 4. Використання геоінформаційних систем для оптимізації FPV-польотів у складних рельєфах .....		
4.1	Постановка задачі 3D-аналізу .....	56
4.2	Отримання 3D-даних (DEM) та аналіз зони видимості (Viewshed) .....	57
4.3	Побудова оптимального шляху (Least Cost Path) .....	60
4.4	Аналіз результатів та ефективності моделі FPV.....	63
4.5	Обмеження методу та перспективи подальших досліджень .....	65
4.6	Висновки до розділу .....	65
РОЗДІЛ 5. Геоінформаційне моделювання супутникових мереж доступу в умовах міської забудови.....		
5.1	Специфіка та проблематика планування мереж LEO .....	67
5.2	Алгоритм створення синтетичної цифрової моделі поверхні .....	68
5.3	Моделювання зон перекриття сигналу .....	71
5.4	Валідація результатів моделювання.....	73
5.4.1	Методологія формування контрольних точок .....	73

5.4.2	Принцип отримання початкових чисел .....	74
5.4.3	Результат моделі DSM + Hillshade .....	74
5.5	Практичне застосування результатів .....	75
5.6	Обмеження методики та шляхи вдосконалення .....	76
5.7	Висновки до розділу .....	76
РОЗДІЛ 6.МОДЕЛЮВАННЯ ПОКРИТТЯ 5G У МІСЬКИХ УМОВАХ НА ОСНОВІ ГЕОІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ .....		78
6.1	Теоретичні засади моделювання покриття 5G у міській забудові.....	78
6.2	Цифрова модель поверхні (DSM) як основа просторового аналізу .....	78
6.3	Лінія прямої видимості як критерій доступності сигналу .....	79
6.4	Формування регулярної сітки базових станцій.....	80
6.5	Інтегральне покриття шляхом об'єднання растрових файлів .....	81
6.6	Просторові особливості отриманого покриття .....	82
6.7	Обмеження моделі .....	83
6.8	Висновки до розділу .....	83
ВИСНОВКИ.....		84
ВИКОРИСТАНІ ІНФОРМАЦІЙНІ ДЖЕРЕЛА .....		87
ДОДАТКИ.....		90

## ВСТУП

Сучасний етап розвитку суспільства характеризується стрімкою цифровізацією всіх сфер життєдіяльності, що призводить до експоненціального зростання потреб у високошвидкісному та надійному телекомунікаційному зв'язку. Ефективне функціонування економіки, системи охорони здоров'я, освіти, державного управління та безпеки в сучасних умовах неможливе без стабільної та розвиненої телекомунікаційної інфраструктури. В умовах глобалізації та посилення конкуренції на ринку телекомунікаційних послуг оператори стикаються з необхідністю швидкого та економічно ефективного розгортання, управління та оптимізації своїх мереж.

Інтеграція геоінформаційних систем (GIS) у процеси планування, будівництва та експлуатації телекомунікаційних мереж відкриває нові можливості для підвищення їх ефективності. GIS дозволяють об'єднати просторові дані про місцевість, інфраструктуру та технічні параметри мережі в єдиній цифровій моделі. Це забезпечує комплексний підхід до управління мережевими активами, дозволяє проводити детальний аналіз зон покриття, оптимізувати маршрути прокладання кабелів, моделювати сценарії розвитку мережі та оперативно реагувати на аварійні ситуації. Використання методів просторового аналізу значно підвищує обґрунтованість прийняття управлінських рішень та дозволяє скоротити витрати на будівництво та обслуговування інфраструктури.

Особливої актуальності набуває застосування GIS для управління гетерогенними мережами, що поєднують різні технології передачі даних. Сьогодення вимагає від телекомунікаційних компаній пропонування комплексних рішень, які можуть включати як традиційні дротові технології, зокрема GPON (Gigabit Passive Optical Network) для високошвидкісного фіксованого доступу, так і сучасні бездротові рішення. Серед останніх можна виділити FPV (First Person View) системи, які використовуються для створення тимчасових мереж зв'язку, моніторингу та відеоспостереження, а також

супутникові системи типу Starlink для забезпечення зв'язку в важкодоступних та віддалених районах. Кожна з цих технологій має свої особливості, переваги та сфери застосування, а їх ефективне комбінування вимагає єдиного інструменту для просторового планування та аналізу.

Метою даної магістерської роботи є розроблення науково-практичних засад інтеграції геоінформаційних систем та методів просторового аналізу для підвищення ефективності планування та управління сучасними телекомунікаційними мережами.

Практична значущість роботи полягає у розробці конкретних методик та моделей, які можуть бути використані телекомунікаційними компаніями для оптимізації витрат, підвищення якості послуг та прискорення прийняття управлінських рішень на основі просторового аналізу.

## РОЗДІЛ 1. Аналітична частина

### 1.1 Сучасні GIS-технології в телекомунікаціях

Сучасний ринок геоінформаційних систем для телекомунікацій характеризується стрімким розвитком як комерційних, так і відкритих рішень. За типом ліцензії програмне забезпечення поділяється на три основні категорії: відкриті (Open Source) системи, серед яких найпопулярнішими є QGIS, PostGIS та GeoServer; комерційні платформи, такі як ArcGIS від ESRI та MapInfo; а також хмарні сервіси, включаючи Google Maps Platform та Mapbox. За функціональним призначенням розрізняють десктопні GIS для інженерного проектування, веб-GIS для колективної роботи, мобільні GIS для польового збору даних та серверні GIS для централізованого управління даними.

Окрему категорію становлять спеціалізовані телекомунікаційні рішення, серед яких слід виділити Userside як комплексну систему управління телеком-оператором, 3-GIS як платформу для управління мережевими активами та IQGeo для мережевого планування. Згідно з дослідженням ринку 2023 року, серед українських телекомунікаційних компаній найбільш поширеними є QGIS як безкоштовна альтернатива та ArcGIS як потужне комерційне рішення **[Помилка! Джерело посилання не знайдено.]**.

#### 1.1.1 Функціональні можливості GIS для телекомунікацій

Сучасні GIS-платформи пропонують широкий спектр функцій для управління телекомунікаційною інфраструктурою. У сфері управління мережевими активами вони забезпечують ведення реєстру мережевих елементів, включаючи OLT, спліттери та муфти, надають можливості візуалізації топології мережі та відстеження технічних характеристик обладнання. Інструменти планування дозволяють оптимізувати маршрути

прокладання кабелів, розраховувати зони покриття бездротових мереж та моделювати сценарії розвитку мережі.

Аналітичні функції включають буферний аналіз для визначення зон впливу, мережевий аналіз для оптимізації маршрутів та просторову статистику для аналізу щільності абонентів. Інтеграційні можливості забезпечуються через REST API для інтеграції з CRM та OSS/BSS системами, підтримку стандартних форматів даних, таких як GeoJSON, KML та Shapefile, а також модулі для роботи з реальним часом (рисунок 1.1) [Помилка! Джерело посилання не знайдено.].

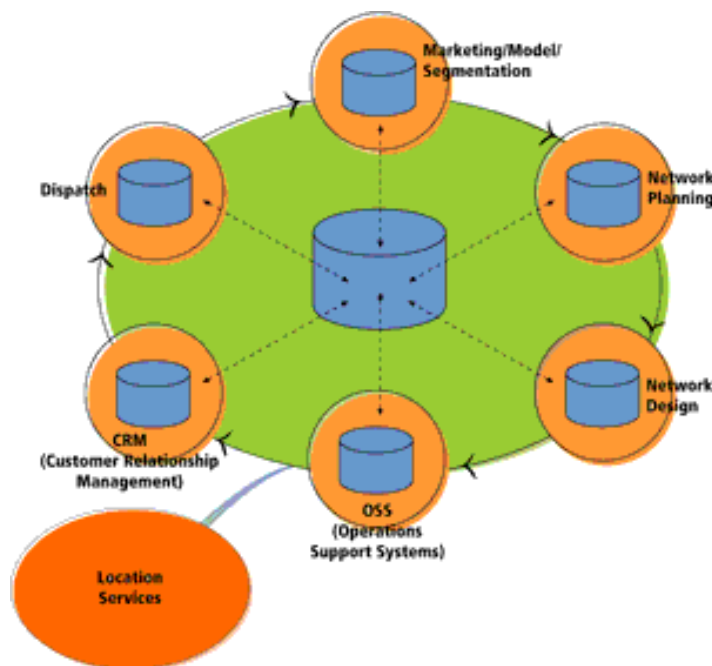


Рисунок 1.1 – Архітектура інтеграції GIS з телекомунікаційними системами

## 1.2 Методи просторового аналізу для телекомунікаційних мереж

### 1.2.1 Буферний аналіз та визначення зон покриття

Буферний аналіз є основним інструментом для визначення зон покриття телекомунікаційних мереж (рисунок 1.2). Для різних технологій застосовуються різні підходи. У випадку GPON-мереж розрахунок

максимальної довжини кабельної ділянки виконується з урахуванням бюджету потужності OLT, втрат на спліттерах, втрат на конекторах та загасання у волокні. Для FPV-дронів розрахунок радіусу покриття FPV-системи враховує потужність передавача, коефіцієнт підсилення антени, чутливість приймача та експоненту втрат на шляху. У випадку Starlink визначається зона видимості супутника на основі висоти орбіти та мінімального кута місця [Помилка! Джерело посилання не знайдено].

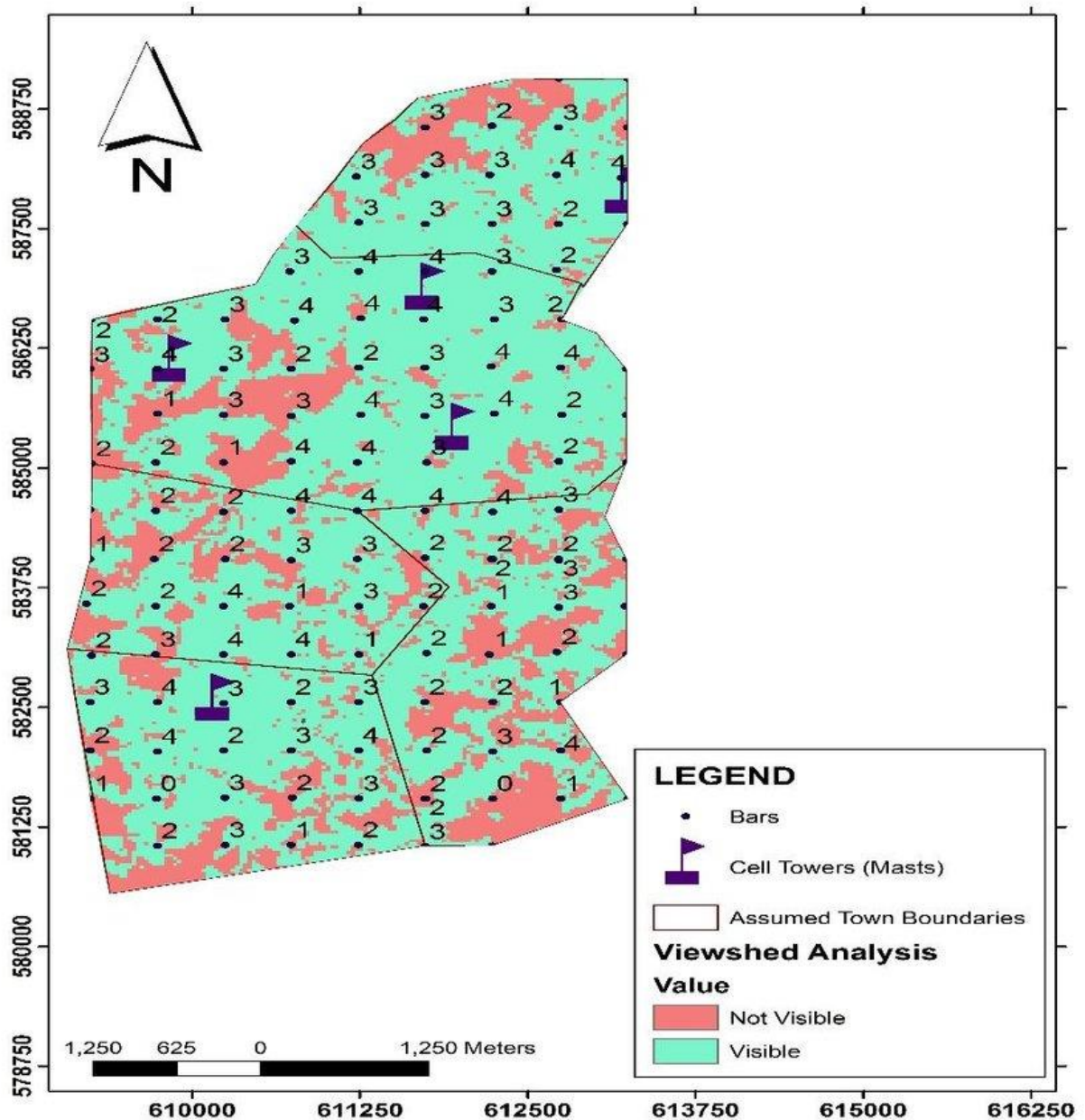


Рисунок 1.2 – Приклад буферного аналізу зони покриття телекомунікаційної мережі

### 1.2.2 Мережевий аналіз для оптимізації інфраструктури

Мережевий аналіз дозволяє вирішувати задачі оптимального прокладання телекомунікаційних мереж. Алгоритми пошуку найкоротшого шляху включають алгоритм Дейкстри для знаходження оптимальних маршрутів, алгоритм A\* для пошуку шляхів з урахуванням евристики та алгоритм Флойда-Уоршелла для аналізу всіх пар вузлів. Задачі мінімального кістяка охоплюють побудову оптимальних деревовидних топологій, мінімізацію загальної довжини кабелів та оптимізацію розташування комутаційних вузлів (рисунок 1.3). Аналіз доступності дозволяє визначати зони, недосяжні для мережевої інфраструктури, оцінювати час відновлення після аварій та планувати резервні маршрути [Помилка! Джерело посилання не знайдено.].

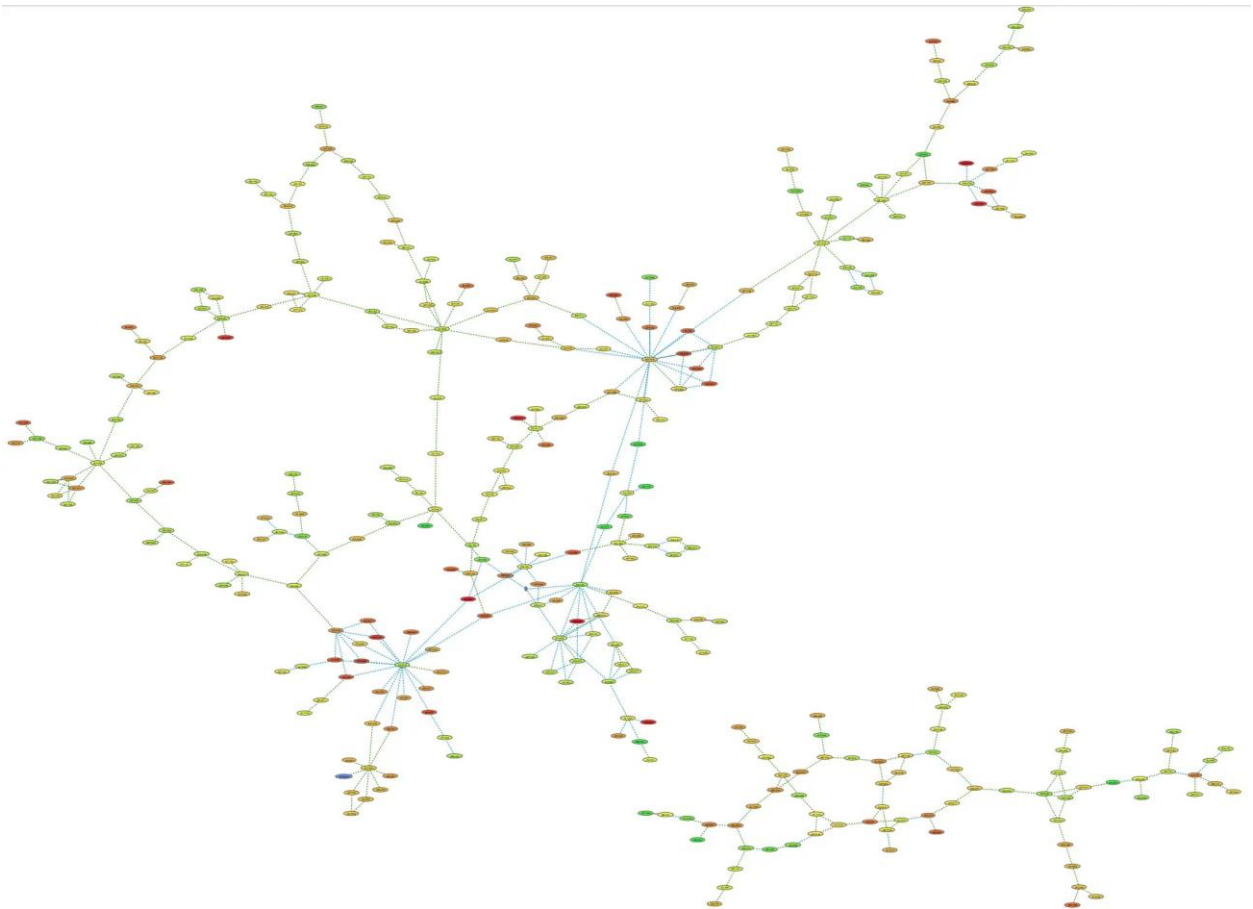


Рисунок 1.3 – Мережевий аналіз топології телекомунікаційної мережі

### 1.3 Методи просторового аналізу для телекомунікаційних мереж

#### 1.3.1 Компоненти GPON-мережі та їх просторове представлення

GPON (Gigabit Passive Optical Network) складається з кількох ключових компонентів, кожен з яких має специфічне просторове представлення в GIS (рисунок 1.4). OLT (Optical Line Terminal) представляється як точка з атрибутами потужності, кількості портів, моделі та статусу, розташовуючись у центральних офісах та вузлах зв'язку. Оптичний кабель відображається як лінія з атрибутами типу волокна, довжини, втрат та року прокладання, трасуючись підземним, повітряним або каналізаційним способом. Спліттери представляються як точки з атрибутами коефіцієнта розгалуження, втрат та кількості портів, розташовуючись у розподільчих шафах та муфтах. Абонентські підключення відображаються як точки з атрибутами типу послуги, швидкості та договору, прив'язуючись до будинків, квартир та офісів [Помилка! Джерело посилання не знайдено.].

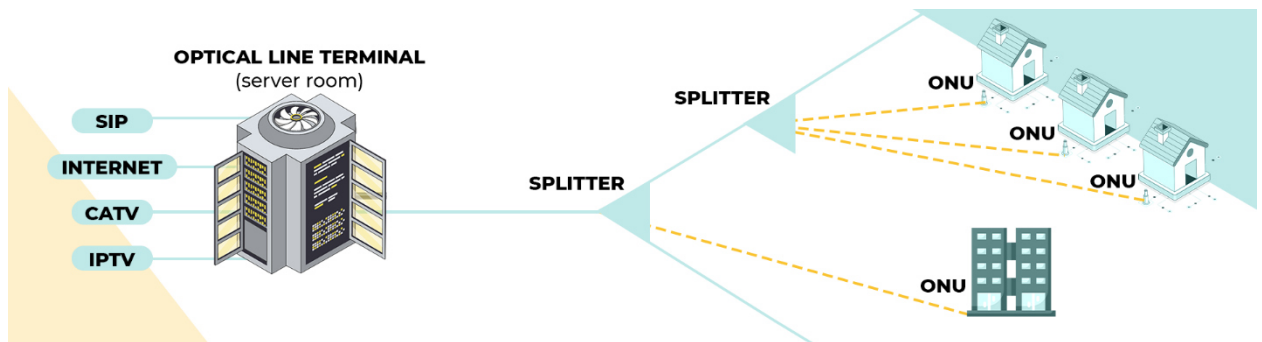


Рисунок 1.4 – Просторове представлення компонентів GPON-мережі в GIS

#### 1.3.2 Моделювання оптичного бюджету потужності

Оптичний бюджет потужності є критичним параметром для проектування GPON-мереж. Його розрахунок включає визначення загальних

втрат, що складаються з втрат у волокні, втрат на спліттерах, втрат на конекторах та запасу потужності. Бюджет потужності визначається як різниця між потужністю OLT та чутливістю ONU. Детальний розрахунок передбачає обчислення втрат у волокні на основі довжини ділянки та коефіцієнта загасання, втрат на спліттерах залежно від коефіцієнта розгалуження, втрат на конекторах з урахуванням їх кількості та додавання запасу потужності для компенсації неперебачуваних втрат. Візуалізація в GIS передбачає кольорове кодування ділянок за рівнем втрат, інтерактивні звіти про порушення бюджету потужності та формування рекомендацій щодо оптимізації мережі [Помилка! Джерело посилання не знайдено.].

### 1.3.3 FPV-системи та моделювання радіопокриття

FPV (First Person View) дрони використовують радіоканали для передачі відео в реальному часі. Моделювання їх покриття включає застосування різних моделей поширення радіохвиль. Модель вільногопростору (Free Space Path Loss) використовується для ідеалізованих умов, модель Окумура-Хта застосовується для міських умов, модель COST-231 використовується для передмість, а дволучева модель враховує відбиття сигналу (рисунок 1.5). Фактори, що впливають на покриття, включають частоту сигналу (1.2 GHz, 2.4 GHz, 5.8 GHz), потужність передавача, чутливість приймача, наявність перешкод у вигляді будівель, дерев та рельєфу, а також атмосферні умови [Помилка! Джерело посилання не знайдено.].

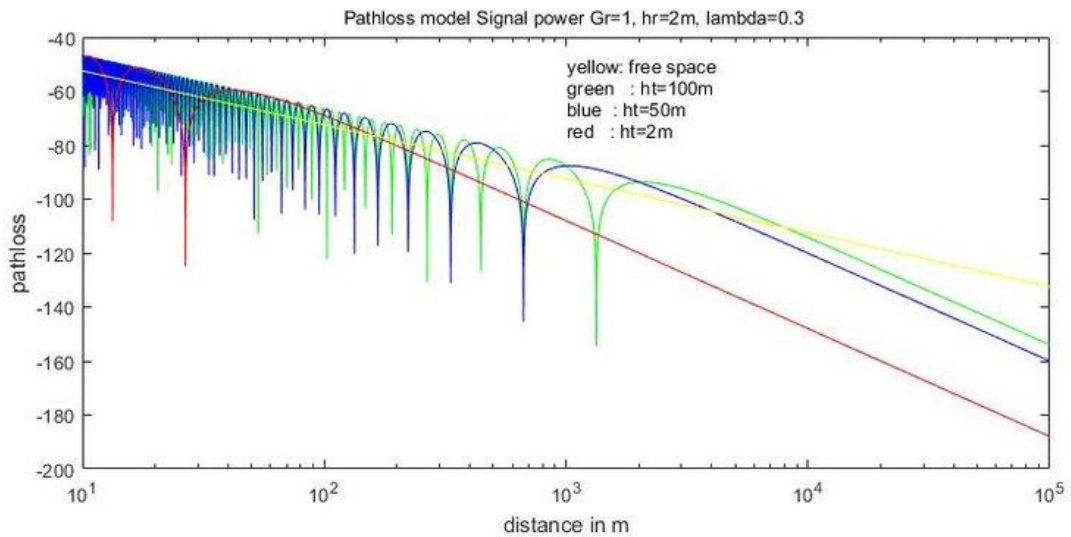


Рисунок 1.5 – Модель вільногопростору (Free Space Path Loss)

### 1.3.4 Супутниковий зв'язок Starlink та аналіз доступності

Система Starlink використовує низькоорбітальні супутники для забезпечення глобального покриття (рисунок 1.6). GIS-аналіз включає розгляд параметрів супутникової системи, серед яких висота орбіти 550 км, кількість супутників у кілька тисяч одиниць, мінімальний кут місця  $25^\circ$  для стабільного зв'язку та азимут як напрямок на супутник [Помилка! Джерело посилання не знайдено.].

Розрахунок зони видимості передбачає визначення положення супутників, обчислення кута місця для кожного супутника та визначення видимих супутників, що перевищують мінімальний кут місця. Фактори доступності Starlink включають географічні координати, наявність перешкод на горизонті, щільність супутникового угруповання та регуляторні обмеження (рисунок 1.7) [Помилка! Джерело посилання не знайдено.].

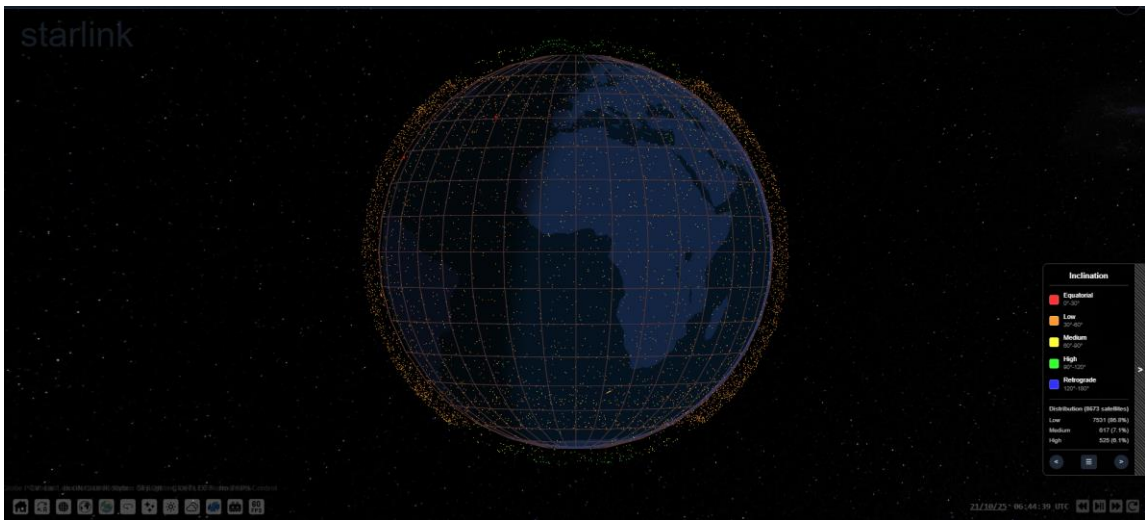


Рисунок 1.6 – Модель з супутниками Starlink

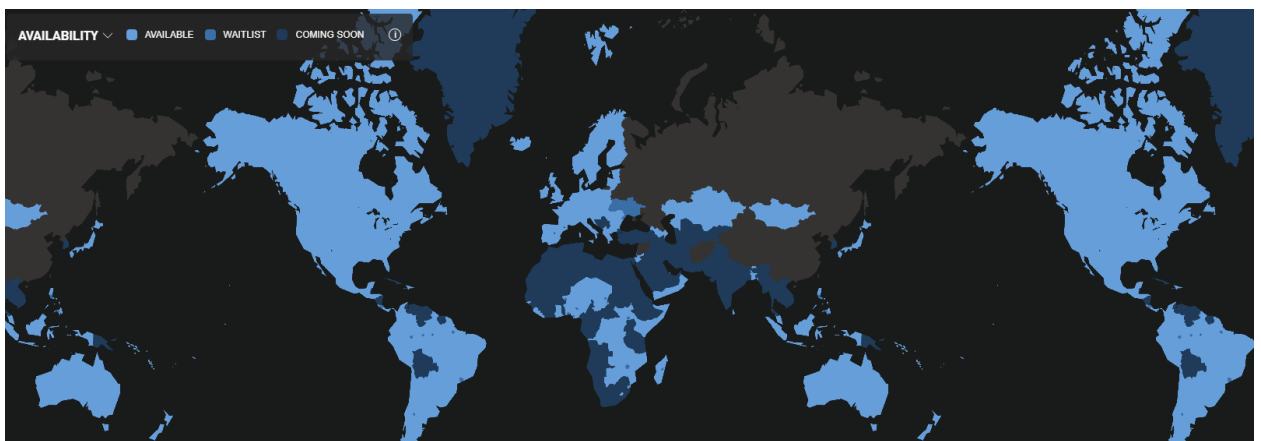


Рисунок 1.7 – Карта доступності супутникового зв'язку Starlink

### 1.3.5 Особливості просторового планування мереж 5G

Впровадження технології 5G, особливо у високочастотному діапазоні міліметрових хвиль (mmWave, 24–100 ГГц), принципово змінює підхід до планування радіомереж. На відміну від попередніх поколінь (3G/4G), де сигналу було достатньо для обходу перешкод, поширення міліметрових хвиль є квазіоптичним і критично залежить від наявності прямої видимості (LoS). Будь-яка фізична перешкода – будівля, дерево чи рекламний щит – може призвести до збою з'єднання (рисунок 1.8).

У ГІС-середовищі моделювання 5G вимагає використання високоточних цифрових моделей поверхні (DSM), що враховують геометрію міської

забудови. Основним методом аналізу є розрахунок зон прямої видимості (Viewshed Analysis) для розміщення базових станцій Small Cell. Оскільки дальність дії таких станцій становить лише 100–300 метрів, інструменти ГІС використовуються для пошуку оптимальних місць для встановлення обладнання (ліхтарні стовпи, фасади будівель) з метою мінімізації «мертвих зон» та ефекту «міського каньйону» [Помилка! Джерело посилання не знайдено.].

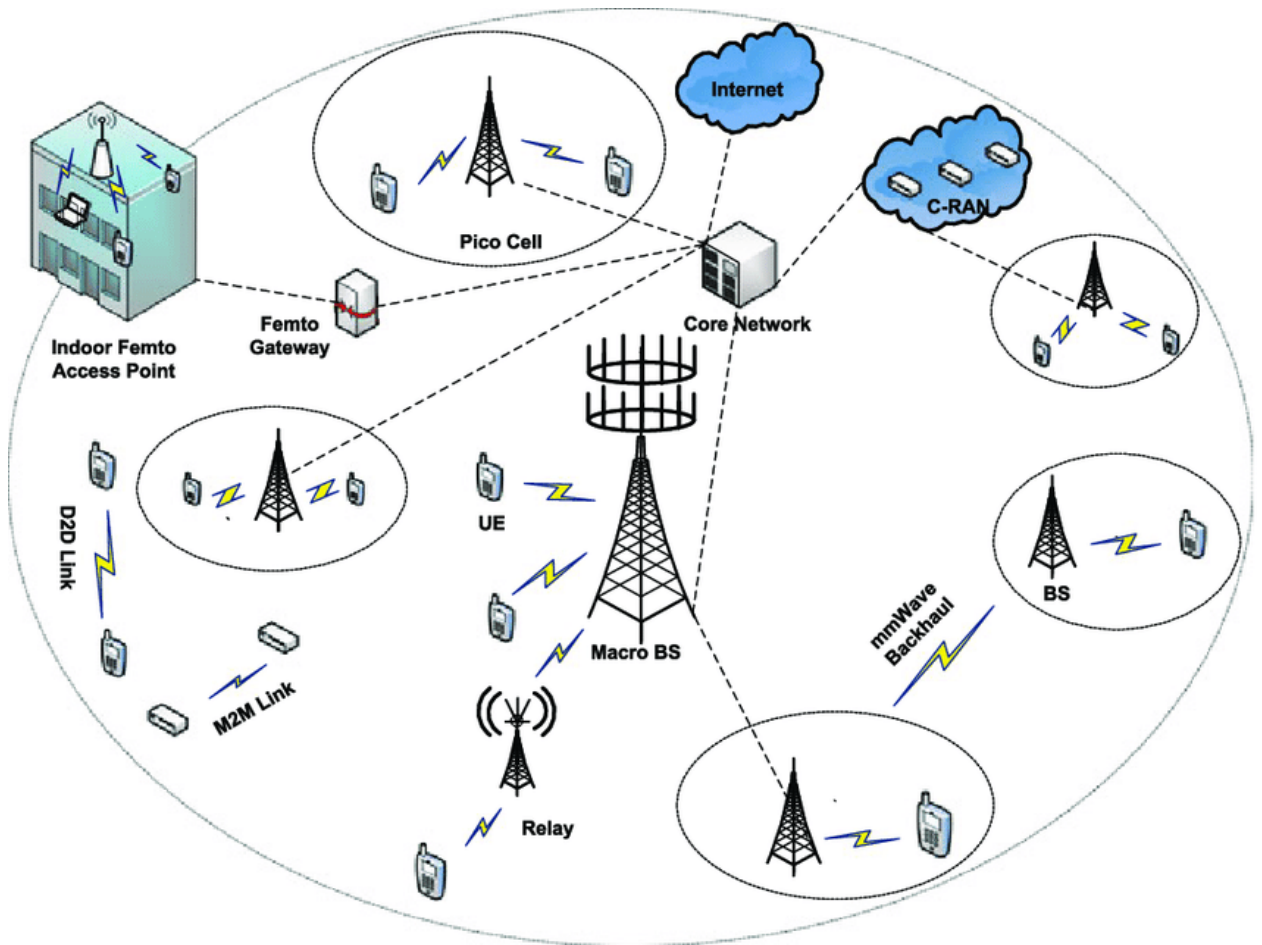


Рисунок 1.8 – Архітектура мережі 5G

#### 1.4 Інтеграція GIS з системами управління мережами

##### 1.4.1 Архітектура інтеграції GIS з OSS/BSS

Сучасні підходи до інтеграції GIS з системами управління мережами передбачають реалізацію на трьох рівнях. На рівні даних здійснюється

синхронізація баз даних через API, що забезпечує актуальність інформації в усіх системах. На рівні сервісів використовуються веб-сервіси для обміну даними, що дозволяє інтегрувати різноманітні системи незалежно від їх технологічної платформи. На рівні застосунків реалізується єдиний інтерфейс користувача, що забезпечує зручну роботу з усіма функціями.

Технології інтеграції включають REST API для обміну даними в реальному часі, бази даних PostGIS для зберігання просторової інформації, веб-сервіси (WMS, WFS) для візуалізації та message brokers (RabbitMQ, Kafka) для асинхронної комунікації (рисунок 1.9). Сценарії використання охоплюють автоматизоване створення заявок на підключення, моніторинг стану мережних елементів на карті, планування профілактичних робіт та аналіз впливу аварій на абонентів [Помилка! Джерело посилання не знайдено.].

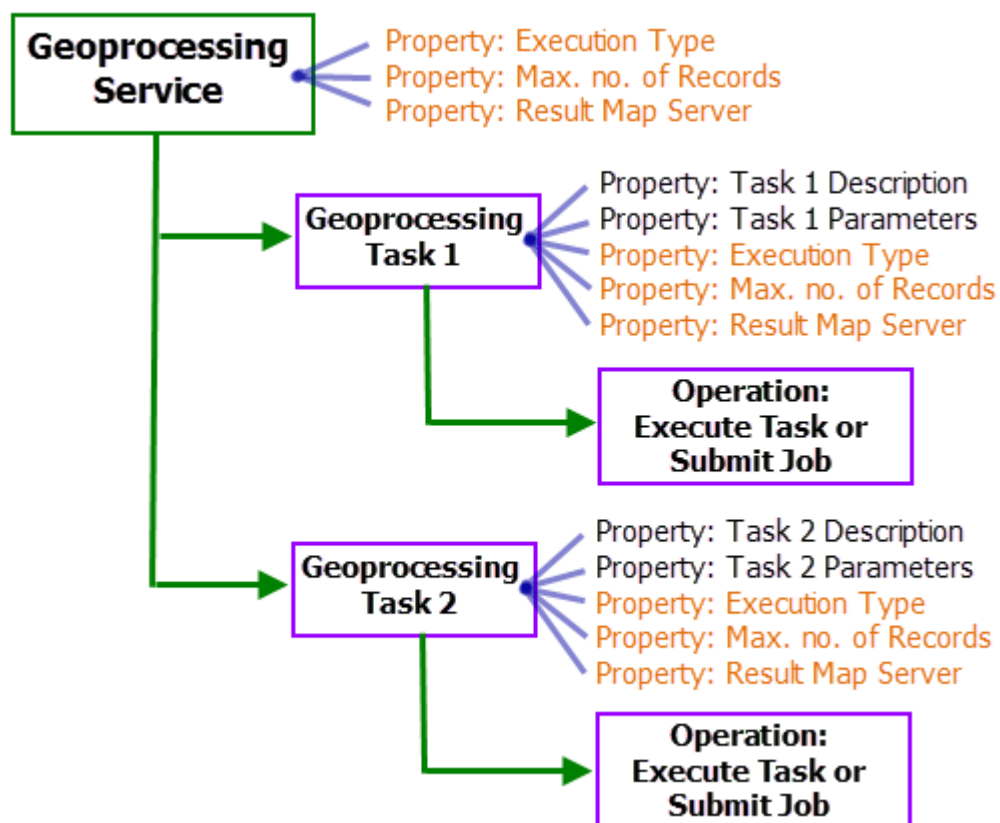


Рисунок 1.9 – Архітектура інтеграції GIS з системами управління мережами

## 1.4.2 Системи моніторингу та їх інтеграція з GIS

Для ефективного управління телекомунікаційними мережами GIS інтегрується з системами моніторингу. Інтеграція з Zabbix та Nagios дозволяє відображати тривоги на карті, корелювати події з географічним положенням та прогнозувати вплив аварій на мережеву інфраструктуру. Інтеграція з Grafana забезпечує створення панелей моніторингу з географічним контекстом, аналіз трендів з урахуванням просторового розподілу та формування звітів про якість послуг по регіонах.

Інтеграція з системами обліку включає підключення до Userside для управління абонентами, до 1С для обліку обладнання та матеріалів, а також до CRM для обслуговування клієнтів. Така інтеграція забезпечує комплексний підхід до управління телекомунікаційною інфраструктурою та підвищує ефективність роботи оператора (рисунок 1.10) [Помилка! Джерело посилання не знайдено.].



Рисунок 1.10 – Інтеграція GIS з системами моніторингу телекомунікаційної мережі

## РОЗДІЛ 2. Методика моделювання телекомунікаційних мереж з використанням GIS

### 2.1 Загальні принципи побудови методики моделювання

Розробка методики моделювання телекомунікаційних систем у геоінформаційному середовищі передбачає поєднання інструментів просторового аналізу, цифрових моделей місцевості, алгоритмів автоматизованої генерації мережевих структур та спеціалізованих сценаріїв, реалізованих у середовищі Python. На відміну від теоретичних положень, викладених у розділі 1, де основна увага зосереджувалася на загальних функціональних можливостях та методах просторової аналітики, даний розділ спрямований на формування практично орієнтованої методики, безпосередньо реалізованої у рамках дипломної роботи. Це означає, що усі подані методичні підходи базуються не на загальних положеннях, а на фактичних алгоритмах, створених автором та інтегрованих до QGIS через механізми PyQGIS і GDAL.

Методика передбачає використання комбінованої моделі, що поєднує можливості 2D- та 2.5D-просторового аналізу. Для телекомунікаційних систем, які мають суттєву залежність від рельєфу та перешкод, використання цифрової моделі рельєфу (DEM) та цифрової моделі поверхні (DSM) є ключовим. Проте, у межах даної роботи моделювання базується саме на 2.5D-підході, який дозволяє враховувати висоту об'єктів, але не потребує повноцінного тривимірного моделювання геометрії будівель або трасування променів. Такий підхід є компромісним рішенням між точністю результатів та ефективністю реалізації, оскільки забезпечує високу швидкість обчислень і достатню точність для практичного використання.

Основою методики є застосування QGIS як інтеграційної платформи, що поєднує функціональність картографічного середовища з можливістю автоматизованого виконання аналізу через Python-скрипти. Це дозволило

створити універсальний підхід, у якому просторові дані зберігаються у векторних та растрових шарах, а обчислювальна логіка реалізована у вигляді окремих модулів. Таким чином, методика є повністю відтворюваною: будь-який етап моделювання може бути повторений із використанням вихідних даних і параметрів.

Особливістю запропонованої методики є її спрямованість на геометричне моделювання телекомунікаційних систем. Це важливе уточнення, оскільки класичні підходи до моделювання GPON, FPV, Starlink чи 5G передбачають використання складних моделей поширення сигналу, оптичних формул, радіофізичних показників, розрахунків SINR і трасування променів. Однак у створених програмних модулях ці моделі не застосовуються. Замість цього використовується інший підхід — просторова інтерпретація телекомунікаційних параметрів через геометрію об'єктів. Саме цей підхід лежить в основі методики та визначає її унікальність.

У разі моделювання оптичних мереж GPON методика передбачає геометричну генерацію мережевих компонентів, їх просторову прив'язку та аналіз довжин ліній. У випадку FPV-систем центральною моделлю є зонування видимості, яке будується на основі цифрової моделі поверхні. Для Starlink визначальним є аналіз горизонту та секторів затінення, що обчислюються через оцінку висотних перешкод у різних азимутальних напрямках. У моделюванні 5G використовується радіусна модель покриття, що формує області доступності сигналу на основі геометричної відстані.

Таким чином, методика побудована навколо єдиного принципу — перетворення технічної задачі на просторове завдання, яке може бути вирішене засобами GIS. Цей підхід дозволяє отримати результати, які є достатніми для попереднього проектування, аналізу сценаріїв і візуалізації телекомунікаційних процесів, а також забезпечує інтеграцію з інженерними інструментами.

### 2.1.1 Використання цифрових моделей рельєфу та поверхні у моделюванні телекомунікаційних систем

У всіх розглянутих телекомунікаційних сценаріях важливу роль відіграє взаємодія мережевих елементів із навколишнім середовищем. Саме тому в методиці ключовим компонентом є цифрові моделі рельєфу (DEM) та цифрові моделі поверхні (DSM). DEM використовується для аналізу загальних рельєфних умов, тоді як DSM дозволяє враховувати висоту будівель та інших надземних об'єктів. Такий поділ дозволяє адаптувати обчислення під потреби різних типів телекомунікаційних систем.

У GPON-моделюванні DEM відіграє допоміжну роль, оскільки основним параметром є геометрична довжина трас кабелів. Натомість FPV та Starlink є високочутливими до вертикальних перешкод, що потребує використання DSM. У програмній реалізації DSM застосовується через обчислення висот для кожної точки вибірки вздовж променя або маршруту, що дозволяє створювати карти видимості та горизонту.

Технічно цифрові моделі рельєфу завантажуються через бібліотеки GDAL та Rasterio, що забезпечує прямий доступ до матриці висот. Такий підхід включає вибірку значення висоти за координатами:

```
with rasterio.open(dem_path) as dem:  
    height = list(dem.sample([(x, y)]))[0]
```

Цей механізм є універсальним та використовується у всіх модулях, де потрібне зчитування висотних значень, включаючи FPV та Starlink.

У методиці DEM/DSM застосовуються не як джерело для складних моделей радіопоширення, а як геометричний інструмент для визначення тіньових зон, прямих ліній видимості та впливу перешкод на прокладення мережі. Це дозволяє досягти високої продуктивності при достатній точності для попереднього інженерного аналізу.

## 2.1.2 Автоматизація просторового моделювання засобами Python та QGIS

Методика моделювання у цій роботі базується на поєднанні просторової інформації, обробленої в QGIS, та алгоритмічної логіки, реалізованої у Python. Така інтеграція дозволяє не лише автоматизувати процеси обчислення, але й виконувати ті операції, які складно або неможливо реалізувати стандартними засобами GIS.

Програмні модулі, розроблені для GPON, FPV, Starlink та 5G, мають подібну структуру. Спочатку виконується імпорт необхідних бібліотек: GDAL — для растрових операцій, QGIS — для створення та редагування векторних шарів, PyQGIS — для взаємодії зі структурами даних QGIS із Python середовища. Далі виконуються операції завантаження вихідних даних, їх трансформації та аналітичної обробки.

У випадку GPON у програмній реалізації формується новий векторний шар, що складається з точок, кожна з яких описує елемент мережі. Створення таких точок відбувається через механізм:

```
layer = QgsVectorLayer("Point?crs=EPSG:4326", "gpon_nodes", "memory")
provider = layer.dataProvider()
provider.addFeatures([feature])
```

Надалі шари зберігаються у форматі GeoPackage або Shapefile, що дозволяє інтегрувати результати в будь-яку систему.

У FPV модуль взаємодіє як із векторними маршрутами, так і з растровою моделлю поверхні. Для створення карти видимості використовується механізм `viewshed` із GDAL, який дозволяє отримати двійкову модель «видно / не видно» на основі DSM. Принцип моделі полягає у перевірці видимості кожної точки на місцевості з точки огляду, що імітує розташування антени FPV-модуля.

Моделювання Starlink також спирається на растрові методи, але тут застосовується інший процес: вибірка значень DSM уздовж радіальних

променів для обчислення профілю горизонту. Цей метод відповідає реальній структурі скрипта і дозволяє сформувати секторну карту доступності сигналу.

У випадку 5G використовується простіший підхід — формування зон покриття у вигляді буферних областей, що створюються через механізми Processing Toolbox. Такий підхід не дублює теоретичну частину розділу 1, а демонструє конкретну реалізацію моделі у рамках роботи.

## 2.2 Методика моделювання GPON-мереж

### 2.2.1 Теоретичні засади моделювання GPON у контексті GIS

Моделювання GPON-мереж у геоінформаційному середовищі базується на поєднанні технічних параметрів оптичних систем з просторовими характеристиками території. На відміну від радіотехнологій, де якість сигналу суттєво залежить від рельєфу, погоди та перешкод, оптичні мережі формально вважаються незалежними від умов навколишнього середовища. Водночас їх просторове моделювання відіграє важливу роль для проектування, оскільки визначає реальні маршрути проходження кабелю, оптимальні місця розташування OLT, муфт та спліттерів, а також дозволяє контролювати довжини сегментів мережі [**Помилка! Джерело посилання не знайдено.**].

З технічної точки зору проектування GPON включає два ключові компоненти: геометричний аналіз та оптичний бюджет потужності. Геометричний аналіз охоплює визначення просторового розташування мережевих вузлів та кабельних трас. Оптичний бюджет визначає максимально допустиму сумарну довжину лінії з урахуванням втрат на волокні, з'єднаннях та розгалуженнях. У даній роботі теоретичні аспекти оптичного бюджету подано для повноти опису методики, проте в програмній реалізації розрахунок бюджету не виконується, що є свідомим спрощенням задля зосередження на просторовій моделі мережі.

Оптичний бюджет визначається за формулою:

$$B = P_{OLT} - S_{ONU}, \quad 2.1$$

де

$B$  - доступний бюджет потужності,

$P_{OLT}$  - вихідна оптична потужність передавача OLT,

$S_{ONU}$  - чутливість приймача ONU.

Сукупні втрати на трасі можна подати у формі:

$$B = L_f + L_s + L_c + M, \quad 2.2$$

де

$L_f$  - втрати у волокні,

$L_s$  - втрати на спліттерах,

$L_c$  - втрати на коннекторах,

$M$  - запас потужності.

Хоча ці параметри мають вирішальне значення для практичного проектування, у даній роботі вони наведені як теоретичний контекст, а основний акцент зроблено на просторовому аналізі та геометричному моделюванні GPON-мереж, які відповідають можливостям створених програмних модулів.

### 2.2.2 Геометричне моделювання GPON-мережі

Оцінка працездатності GPON-мережі базується на обчисленні оптичного бюджету, який визначає максимальні втрати сигналу між передавачем на стороні OLT і приймачем на стороні ONU. Алгоритм включає декілька послідовних етапів, які в роботі були реалізовані за допомогою Python-модулів і PyQGIS API.

Першим кроком є визначення початкових технічних характеристик обладнання. Потужність передавача OLT та чутливість ONU визначають допустимий діапазон загасання. Алгоритм ініціалізації параметрів реалізується у вигляді функції:

```
def calculate_initial_parameters(olt_model, onu_model):
```

```
olt_power = get_olt_transmit_power(olt_model)
onu_sensitivity = get_onu_receive_sensitivity(onu_model)
return olt_power - onu_sensitivity
```

Наступним етапом є розрахунок втрат у оптичному волокні. Довжина маршруту визначається безпосередньо з геометрії кабельного шару, а коефіцієнт загасання залежить від довжини хвилі. Для цього використовується функція:

```
def calculate_fiber_loss(distance_km, wavelength=1310):
    return distance_km * get_attenuation_coefficient(wavelength)
```

Пасивні компоненти, такі як спліттери та конектори, додають до загального загасання власні втрати. Вони враховуються окремим блоком алгоритму. Втрати на спліттері та конекторах розраховуються як:

```
def calculate_passive_losses(splitter_ratio, connector_count):
    return get_splitter_loss(splitter_ratio) + connector_count * 0.5
```

Оскільки мережа повинна мати запас потужності для компенсації старіння обладнання та змін умов експлуатації, загасання доповнюється експлуатаційним запасом. Завершальний етап алгоритму передбачає порівняння сумарних втрат із доступним бюджетом потужності:

```
def check_power_budget(power_budget, total_loss):
    return power_budget - total_loss
```

Цей алгоритм був реалізований як Геометричний підхід, реалізований у програмних модулях роботи, передбачає автоматизоване створення просторової моделі GPON-мережі у середовищі QGIS. Основним інструментом виступає механізм формування векторних шарів, у яких кожна точка або лінія містить атрибутивну інформацію про мережевий елемент **[Помилка! Джерело посилання не знайдено.]**.

Методика побудови полягає у таких етапах:

Передусім формується вихідний набір даних, який представляє собою геометрію забудови, абонентські точки та потенційні місця встановлення обладнання. На основі цих даних формується набір точок, що репрезентують різні типи GPON-вузлів: OLT, спліттери, розподільчі коробки SFU та абонентські підключення. Програмна логіка передбачає створення цих об'єктів безпосередньо у Python через механізми PyQGIS, що дозволяє інтегрувати їх у робочу карту та забезпечити подальшу обробку.

Топологія мережі формується через алгоритм, який проходить послідовність прив'язки «OLT → спліттер → SFU → ONU». Для кожного спліттера визначається найближча група абонентських вузлів, що відображає реальне зональне призначення спліттерів у GPON-мережах. Кабельні лінії не обчислюються як радіальні або найкоротші маршрути, а створюються як прямі лінійні сегменти між об'єктами. Такий підхід відповідає концепції геометричного моделювання та дозволяє зберегти відтворюваність аналізу без використання складних мережевих алгоритмів.

Ключовим аспектом реалізації є автоматичне створення та запис шарів у форматі GeoPackage. Це забезпечує структурованість даних та сумісність з іншими інженерними інструментами. Використання пам'ятних (in-memory) шарів пришвидшує обчислення, оскільки дозволяє уникати зайвих дискових операцій на проміжних етапах моделювання.

### 2.2.3 Просторовий аналіз мережі та контроль довжин сегментів

Це дозволяє швидко отримати інформацію про потенційні перевищення довжин та виявити ділянки, що потребують перегляду структури мережі. Геометричний аналіз дозволяє також виявляти аномальні підключення, наприклад, коли абонент знаходиться значно далі від групи спліттера, до якої він має бути підключений.

Крім того, результати аналізу можуть бути візуалізовані у QGIS через кольорове маркування сегментів. Це є важливою інженерною складовою, оскільки дозволяє оператору отримувати швидке візуальне уявлення про стан мережі і приймати рішення про зміни конфігурації.

## 2.3 Методика моделювання FPV-систем

### 2.3.1 Теоретичні основи моделювання FPV-зв'язку в контексті GIS

FPV (First Person View) системи належать до класу радіовізуальних комплексів, які передають відеосигнал у реальному часі з борта безпілотного літального апарата. На відміну від класичних радіомереж, FPV-зв'язок працює у відкритих діапазонах 2.4 ГГц та 5.8 ГГц, що робить його високочутливим до перешкод, створених рельєфом та об'єктами забудови. У науковій практиці для аналізу таких систем застосовуються різні моделі поширення сигналу: моделі вільного простору, дволучеві моделі, моделі Окумура–Хата чи COST-231. Проте використання цих моделей передбачає залучення великої кількості параметрів: потужності передавача, коефіцієнтів підсилення антен, показників загасання, статистичних характеристик середовища **[Помилка! Джерело посилання не знайдено.]**.

У межах даної роботи обрано інший підхід, орієнтований на просторову доступність FPV-сигналу через модель видимості (viewshed). Причина цього полягає у практичній специфіці FPV-дронів: їхній відеосигнал, який передається аналоговим або цифровим методом, різко втрачає якість при наявності перешкод між передавальною антеною та приймачем. Таким чином, для FPV-каналів ключовими є саме лінія видимості та геометричні перешкоди, а не детальні радіофізичні характеристики.

Використання моделі viewshed, яка є класичним інструментом геоінформаційного аналізу, дозволяє визначити ті ділянки простору, що

доступні для прямої видимості з позиції оператора або антени. Оскільки FPV-передавач зазвичай встановлений на дроні, а приймач — на землі,

обчислення видимості у площині DSM стає цілком адекватною моделлю для оцінки якості зв'язку на рівні попереднього аналізу. Таким чином, у межах методики FPV-системи трактуються як геометрична задача видимості, а не як задача моделювання радіопоширення, що відповідає як особливостям FPV-комунікації, так і програмній реалізації у роботі **[Помилка! Джерело посилання не знайдено.]**.

### 2.3.2 Побудова растрової моделі видимості FPV за допомогою GDAL Viewshed

Основою просторового моделювання FPV у роботі виступає цифрова модель поверхні (DSM), яка містить інформацію про рельєф та висоту об'єктів забудови. Саме DSM дозволяє врахувати вертикальні перешкоди, які можуть блокувати лінію FPV-зв'язку. На її основі за допомогою інструментів GDAL формується растрова карта видимості, у якій кожен піксель позначає, чи є точка доступною для прямої видимості відносно місця розташування антени **[Помилка! Джерело посилання не знайдено.]**.

Програмна реалізація передбачає використання стандартного інструмента `gdal.ViewshedGenerate`, який обчислює видимість у межах певного радіуса. У скрипті передбачено параметр висоти антени («height of observer»), що дозволяє моделювати реальні умови роботи дрона або приймального обладнання:

```
gdal.ViewshedGenerate(
    ds,          # DSM-модель
    view_ds,    # результат
    x, y, height, # координати та висота антени
```

```

max_distance, # радіус моделі
-1.0, 1.0, # пороги видимості
0.0, # кут падіння
0)

```

Результат розрахунку формується у вигляді двійкового растру, де значення 1 означає, що точка є видимою, а значення 0 – що вона знаходиться у тіні. Цей растр у подальшому використовується як основа для аналізу маршруту FPV-дрона.

Методика дозволяє урахувати складну забудову, присутність промислових об'єктів та природні перешкоди. Модель є статичною, але точність, яку вона забезпечує, є цілком достатньою для оцінки можливості проведення польоту у конкретному районі.

### 2.3.3 Аналіз маршруту FPV-дрона в контексті моделі видимості

Після побудови карти видимості наступним етапом є оцінка безперервності FPV-зв'язку вздовж заданого маршруту дрона. Маршрут подається як векторна лінія, що складається з упорядкованих точок. Для кожної точки виконується вибірка значення з растрової карти видимості:

```
value = viewshed_raster.read(1)[row, col]
```

Якщо у певній точці значення дорівнює 0, це означає, що FPV-сигнал буде втрачено через відсутність прямої видимості.

На основі цих перевірок формується підсумкова аналітика маршруту. Скрипт ідентифікує сегменти, у яких траєкторія перетинає «тіньові» зони, і групує їх для подальшого аналізу. Це дозволяє визначити критичні ділянки польоту, де дрон ризикує втратити відеосигнал.

Результат подається у вигляді повідомлення, що описує стан маршруту, або у вигляді додаткового векторного шару, який позначає ділянки з потенційною втратою зв'язку. Такий підхід дозволяє оператору швидко

оцінити, чи можливо здійснити політ без втрати FPV-передачі, або в яких місцях потрібна додаткова інфраструктура.

#### 2.3.4 Генерація рекомендацій щодо встановлення ретрансляторів

Оскільки FPV-системи часто застосовуються у складних урбаністичних умовах, присутність зон тіні є типовим явищем. У програмній реалізації передбачено автоматизований механізм генерації рекомендацій щодо встановлення ретрансляторів.

Модуль аналізує просторове розташування критичних ділянок маршруту і пропонує орієнтовні місця встановлення додаткових точок підсилення FPV-сигналу. Підбір точок здійснюється за принципом:

- точка повинна знаходитися у видимій зоні відносно початкової антени,
- має мати пряму видимість на сегмент маршруту, який опинився у тіні.

Таким чином створюється просторово обґрунтована рекомендація, що враховує геометричні умови місцевості та забезпечує можливість безпечного польоту FPV-дрона.

#### 2.3.5 Переваги та обмеження застосованої методики

Методика моделювання FPV, реалізована у роботі, має низку переваг. Найважливішими серед них є висока продуктивність, простота інтеграції у GIS-середовище, можливість швидкої візуалізації та достатня точність для попереднього проєктування маршрутів. Одним із ключових позитивних аспектів є універсальність моделі: її можна застосовувати для різних сценаріїв FPV-польотів, незалежно від типу дрона чи обладнання.

Водночас модель має і свої обмеження. Оскільки у ній не використовуються радіофізичні моделі поширення хвиль, не враховується

вплив частоти, потужності та параметрів антен. Крім того, модель viewshed не враховує динаміку польоту та зміну висоти дрона, оскільки обчислює видимість для фіксованої висоти антени. Попри це, методика повністю відповідає задачам попереднього просторового аналізу і дозволяє отримати інженерно корисні результати.

## 2.4 Методика моделювання системи супутникового зв'язку Starlink

### 2.4.1 Геометричний підхід до моделювання доступності супутникового сигналу

Супутникові системи на зразок Starlink суттєво залежать від наявності вільного горизонту, оскільки приймальна антена повинна мати безперервний огляд значної частини небесної півсфери. Теоретичні засади супутникового зв'язку було розглянуто в аналітичній частині роботи, тому в даному розділі увагу зосереджено на практичній методиці визначення доступності Starlink у GIS-середовищі на основі цифрової моделі поверхні.

На відміну від моделей, що визначають доступність сигналу через розрахунок орбіт супутників, їхніх координат та азимутально-висотних параметрів, методика, застосована у програмній реалізації, базується на аналізі геометричного горизонту. У такому підході припускається, що сигнал може бути прийнятий лише на тих азимутах, де немає перешкод, що перевищують мінімальний кут місця. Таким чином, замість обробки ефермерид чи моделювання руху супутників використовується більш універсальна задача — оцінка того, наскільки «відкритим» є горизонт у кожному напрямку.

Перевага такого підходу полягає у тому, що він повністю узгоджується з реальними вимогами системи Starlink, де, за даними оператора, стабільність роботи напряму залежить від відсутності перешкод вище приблизно  $25^\circ$  над горизонтом. Аналіз горизонтальної видимості на основі DSM дозволяє визначити, чи є певна точка придатною для встановлення терміналу з урахуванням локальної забудови **[Помилка! Джерело посилання не знайдено.]**.

### 2.4.2 Формування профілів горизонту за цифровою моделлю поверхні

Основним етапом методики є обчислення профілю горизонту у всіх азимутальних напрямках. Для цього цифрова модель поверхні (DSM) використовується як джерело висотних значень, що описують рельєф та будівлі. Процес складається з вибірки висоти вздовж променя, який простягається від точки встановлення антени до певного радіального віддалення.

У програмній реалізації застосовується послідовний аналіз висоти за допомогою періодичного забору значень:

```
with rasterio.open(dsm_path) as dsm:
    height = list(dsm.sample([(x_pos, y_pos)]))[0]
```

Значення висоти порівнюються з висотою точки спостереження, що дозволяє визначити, чи знаходиться об'єкт на промені вище за припустимий поріг. Якщо перешкода перевищує видимий профіль, азимут позначається як заблокований.

Методика враховує послідовний обхід  $360^\circ$  кола з певним кроком, формуючи таким чином набори секторів, у яких сигнал може бути заблокований або доступний. Це дає змогу створити секторальну карту горизонту, яка є одним із ключових продуктів моделювання у межах роботи.

Оскільки DSM включає як природний рельєф, так і об'єкти забудови, модель дозволяє точно оцінити вплив багатоповерхових будівель, промислових споруд або дерев на якість супутникового прийому.

### 2.4.3 Побудова карти доступності супутникового сигналу

Після обчислення профілів горизонту формується растрова або векторна карта доступності сигналу Starlink. Програмна реалізація передбачає генерацію набору секторів, що поділяють небесну півсферу на частини, кожна з яких містить інформацію про наявність або відсутність перешкод.

Найпоширенішим способом візуалізації, який відповідає створеним скриптам, є позначення відкритих секторів зеленим кольором, а закритих – червоним.

Таке тематичне стилювання дозволяє оператору швидко оцінити можливість встановлення терміналу та визначити, чи відповідає місце вимогам безперебійного прийому супутникового сигналу. У міських умовах така карта є особливо важливою, оскільки щільна забудова може створювати численні перешкоди на малих кутах місця **[Помилка! Джерело посилання не знайдено.]**.

У деяких випадках скрипт також створює інтегральну метрику доступності сигналу, що визначає, яка частка секторів є відкрита. Це дозволяє формувати коротке аналітичне повідомлення:

«Рівень доступності супутникового сигналу оцінюється як достатній/обмежений/критичний».

Хоча методика не враховує реальних орбіт та позицій супутників, вона цілком відповідає практичному критерію доступності Starlink, який у першу чергу визначається геометричною видимістю небесної півсфери.

#### 2.4.4 Практичні аспекти встановлення терміналу Starlink у контексті результатів моделювання

Методика моделювання, реалізована у роботі, не лише дозволяє оцінити доступність супутникового сигналу, але й формує практичні рекомендації щодо встановлення терміналу. Аналіз горизонтальних профілів дозволяє визначити напрямки, у яких перешкоди є критичними, і вказати, на яку сторону будинку або ділянки варто перенести термінал для забезпечення оптимального прийому.

У програмній реалізації результати можуть бути нанесені на карту у вигляді окремого шару, що містить:

- точку встановлення антени,

- межу зони аналізу,
- секторну карту перешкод.

Завдяки цьому оператор може безпосередньо у GIS середовищі змінити точку встановлення антенного терміналу та повторно виконати моделювання. Це робить методику інтерактивною та придатною для різних сценаріїв проектування.

Крім того, у середовищі міської забудови методика дозволяє виявити як локальні перешкоди (окремі будівлі), так і загальні структурні обмеження (щільні квартали, заводські комплекси). Таким чином забезпечується комплексна оцінка просторових умов для роботи Starlink **[Помилка! Джерело посилання не знайдено.]**.

#### 2.4.5 Обмеження та можливості удосконалення

Геометрична модель, реалізована у роботі, має низку переваг, таких як простота, швидкість обчислень та можливість роботи з різними DSM. Проте вона не враховує динаміку супутникового угруповання, погодні умови чи вплив багатопроменевості. Незважаючи на це, методика повністю відповідає задачам попереднього аналізу та практичним потребам оператора, який передусім має визначити якість огляду горизонту.

У подальших дослідженнях можливим розширенням моделі є інтеграція актуальних орбітальних даних або додавання параметрів електромагнітного аналізу, проте ці елементи не входять до реалізованої моделі у рамках даної дипломної роботи.

## 2.5 Методика моделювання покриття мереж 5G

### 2.5.1 Геометричні основи моделювання зон радіопокриття

Мережі п'ятого покоління (5G) характеризуються широким діапазоном частот, значною щільністю розташування базових станцій та чутливістю до геометрії міського середовища. Проте повноцінне моделювання радіопокриття, яке враховує втрати у середовищі, багатопроменевість, відбиття та дифракцію, вимагає застосування складних RF-движків. У межах цієї роботи реалізовано інший підхід, який базується на геометричній оцінці зони доступності сигналу через буферне моделювання **[Помилка! Джерело посилання не знайдено.]**.

Геометричний метод дозволяє сформувати початкову, наближену модель покриття, яка застосовується для просторового аналізу телекомунікаційної інфраструктури та оцінки доступності базових станцій у конкретному районі. Такий підхід є корисним на етапі попереднього проектування або для візуалізації планових зон покриття.

У скриптах для моделювання 5G використовується механізм побудови буферних зон, який є базовим інструментом GIS. Буфер формується довкола точки розташування базової станції на певну відстань, що відповідає теоретичному або нормативному радіусу дії станції у конкретних умовах. Отримана зона трактується як область можливого прийому сигналу **[Помилка! Джерело посилання не знайдено.]**.

Такий підхід не дублює теоретичний аналіз радіопоширення, наведений у розділі 1, а формує окрему, практично орієнтовану методику, що відповідає створеним програмним модулям.

### 2.5.2 Побудова буферних зон покриття базових станцій

У програмній реалізації моделювання 5G починається з формування векторного шару базових станцій. Кожна станція характеризується координатами та набором атрибутів, які можуть включати ідентифікатор станції, діапазон частот або тип антени. Скрипт виконує побудову буферних зон за допомогою інструментів Processing Toolbox у QGIS.

Процес створення буфера у скрипті реалізовано через виклик інструмента:

```
processing.run("native:buffer", {  
  'INPUT': towers_layer,  
  'DISTANCE': radius,  
  'SEGMENTS': 50,  
  'OUTPUT': 'memory:'  
})
```

Параметр DISTANCE визначає радіус зони покриття. Згладжування буферної кривої забезпечує плавну форму полігона, що покращує читабельність результатів.

Отримана зона покриття є геометричною моделлю та відповідає наближеному радіусу прийому сигналу без урахування реальних перешкод. Результат інтегрується у карту і може бути додатково стилізований для надання візуального представлення операторам.

### 2.5.3 Об'єднання зон покриття кількох базових станцій

Оскільки мережі 5G мають високу щільність розташування базових станцій, методика передбачає можливість об'єднання окремих зон покриття для оцінки сумарної доступності в конкретному районі. У скриптах реалізовано операцію злиття за допомогою інструмента native:dissolve, який дозволяє формувати єдину інтегральну зону покриття:

```
processing.run("native:dissolve", {  
  'INPUT': buffer_layer,  
  'OUTPUT': 'memory:'  
})
```

Таке об'єднання дає змогу класифікувати зони покриття за їхньою структурою, виокремлюючи ділянки з безперервним сигналом (суцільні полігони), фрагментованим покриттям (де наявні розриви) та зони надлишкового перекриття (накладання кількох станцій).

Результат можна зобразити окремим шаром, що дозволяє візуально оцінити плановану або існуючу структуру покриття 5G. У міських зонах цей підхід корисний для аналізу «білих плям» або визначення місць, де потрібне встановлення нових базових станцій **[Помилка! Джерело посилання не знайдено.]**.

#### 2.5.4 Аналіз на основі геометричної моделі покриття

Попри те, що методика не використовує фізичні моделі радіопоширення, вона дозволяє виконувати низку інженерних аналізів. Просторове моделювання покриття дає можливість визначити райони, що потенційно потребують додаткової інфраструктури, та оцінити відповідність запланованої кількості станцій вимогам території.

Карта покриття може бути поєднана з демографічними або інфраструктурними даними у GIS, що дозволяє:

- оцінити зональне охоплення житлових і промислових районів,
- визначити пріоритетні місця для встановлення додаткових станцій,
- моделювати сценарії розширення мережі.

Таким чином, геометрична модель 5G є не настільки технічно точною, як радіофізичні моделі, проте забезпечує достатню інформативність для етапів планування.

#### 2.5.5 Обмеження та перспективи моделі

Методика, реалізована у роботі, обмежена тим, що не враховує реальних умов поширення сигналу та висотних перешкод. Буферний підхід передбачає, що прийом можливий у межах певного радіуса, що є спрощенням порівняно з дійсністю. Однак практична цінність моделі полягає у простоті інтеграції з іншими шарами GIS та можливості виконання швидкого аналізу покриття.

У подальших роботах можливо розширити модель, інтегрувавши DEM, DSM або базові моделі радіопоширення. Це може включати побудову карт видимості, визначення коефіцієнтів загасання та аналіз впливу забудови. Проте такі елементи не входили до специфікації реалізованих програмних модулів.

## РОЗДІЛ 3. Практична реалізація GIS-моделювання та оптимізації мережі GPON

### 3.1 Вступ до практичної частини дослідження

Сучасні телекомунікаційні мережі вимагають точного просторового планування, ефективного управління ресурсами та швидкого інженерного аналізу. Це особливо актуально для мереж доступу GPON (гігабітна пасивна оптична мережа), де просторове розташування абонентів, довжина кабельних ліній та топологія інфраструктури безпосередньо впливають на пропускну здатність, якість сигналу та капітальні витрати.

Традиційні підходи до проектування мереж GPON часто передбачали ручне малювання елементів на паперових схемах або використання простих векторних карт без автоматизації. Такий підхід є трудомістким і недостатньо точним (рисунок 3.1).



Практична частина дослідження була реалізована в середовищі QGIS 3.4.10 LTR у поєднанні з бібліотеками Python та PyQGIS API. Цей інструментарій надає можливість автоматизувати обробку геопросторових даних та створити повноцінну цифрову модель мережі GPON

Основні компоненти середовища наведено в таблиці 3.1.

Таблиця 3.1 - Основні технічні характеристики розробки середовища

Компонент	Версія	Призначення
QGIS	3.4.10 LTR	Основна ГІС-платформа
Python	3.7	Мова програмування для автоматизації
PyQGIS	3.4	API для взаємодії з QGIS
GDAL/OGR	2.4	Бібліотека обробки геоданих
GeoPandas	0.5	Робота з геоданими в Python
NumPy	1.16	Математичні обчислення

### 3.3 Підготовка та попередня обробка вихідних геопросторових даних

Вхідні геодані будівель були завантажені з OpenStreetMap. Їх атрибутивний аналіз показав різні типи забудови, тому перед подальшим моделюванням фільтрацію було виконано лише для об'єктів типу SFU (Single Family Unit), що надаються приватним домогосподарствам (рисунок 3.2).

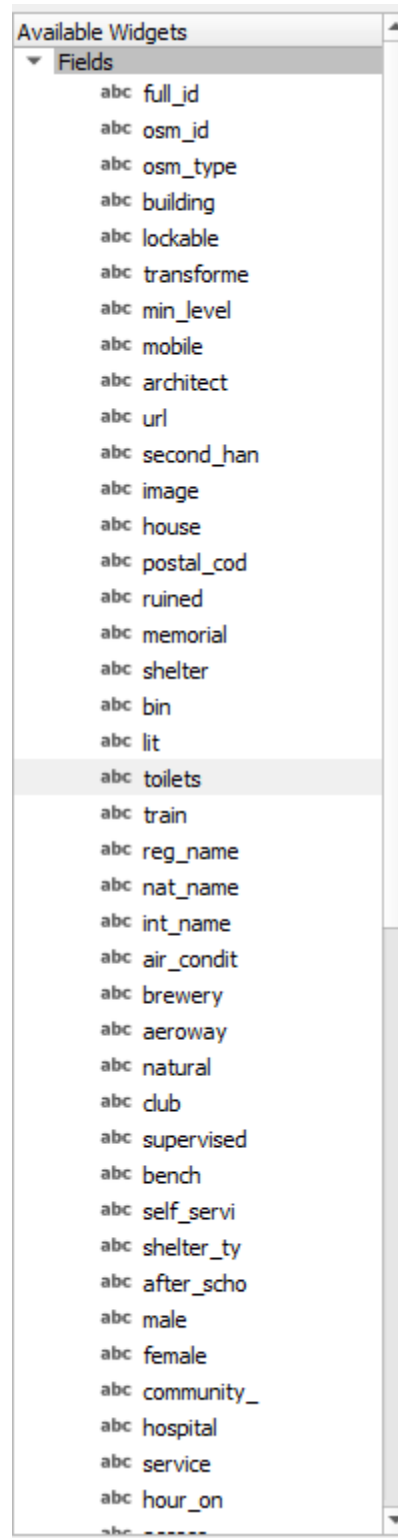


Рисунок 3.2 – Фрагмент таблиці атрибутів вхідного шару будівлі

Для подальших розрахунків усі об'єкти були перепроєктовані з EPSG:4326 в метричну систему EPSG:32636. Це критично важливо, більшість довжин кабелів необхідно вимірювати в метрах. Перетворення було виконано програмно:

```

crs_src = QgsCoordinateReferenceSystem('EPSG:4326')
crs_dest = QgsCoordinateReferenceSystem('EPSG:32636')
transform_context = QgsProject.instance().transformContext()
xform = QgsCoordinateTransform(crs_src, crs_dest, transform_context)

```

```

for feature in building_layer.getFeatures():
    geom = feature.geometry()
    geom.transform(xform)

```

Результати аналізу вихідних даних наведено у таблиці 3.2.

Таблиця 3.2 - Статистика вихідних даних забудови

Параметр	Значення	Пояснення
Кількість об'єктів	2,847	Загальна кількість будівель
Площа зони аналізу	28.5 км <sup>2</sup>	Територія в межах буфера 3км
Типи забудови	apartment, house, residential, garage	Різноманітність категорій
Кількість абонентів SFU	500	Вибірка для моделювання

### 3.4 Формування точкової моделі абонентів

Будівлі абонентів перетворюються на точкову модель шляхом обчислення центроїдів полігонів. Центроїд полігону з координатами вершин

$$(x_i, y_i), i = 1 \dots n, \quad 3.1$$

визначається за формулами:

$$A = \frac{1}{2} \sum_{i=1}^n (x_i y_{i+1} - x_{i+1} y_i) \quad 3.2$$

$$C_x = \frac{1}{6A} \sum_{i=1}^n (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad 3.3$$

$$C_y = \frac{1}{6A} \sum_{i=1}^n (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad 3.4$$

У реалізації PyQGIS ці розрахунки надаються автоматично методом `.centroid()`.

### 3.5 Побудова абонентських кабельних ліній

На основі точок абонента формуються лінійні відрізки `LineString`, які імітують підключення абонента до точки мережі. Довжина кожної лінії розраховується в метричній проекції.

Геометрична довжина відрізка між точками  $(x_1, y_1)$  і  $(x_2, y_2)$  задається формулою:

$$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad 3.6$$

У PyQGIS вона визначається автоматично:

- `length_m = line_geometry.length()`

Кабельні об'єкти доповнюються атрибутами:

- `cable_feature['drop_id'] = f"DROP_{i}"`
- `cable_feature['distance'] = length_m`
- `cable_feature['status'] = "VALID"`

### 3.6 Формування бази даних GPON та експорт результатів

Після побудови всіх деяких геометричних об'єктів — абонентських точок та кабельних ліній — було сформовано єдину просторову структуру об'єктів у форматі `GeoPackage`. Формат було обрано завдяки його універсальності, підтримці кількох таких векторних шарів в межах одного

файлу та відповідності вимогам сучасних геоінформаційних систем. Використання контейнера GeoPackage дозволяє не тільки зберігати просторові дані, але й забезпечити зручність їх структурування під час передачі між іншими системами або етапами моделювання.

Структура бази даних включає три основні шари, що суперечать логічному поділу мережі GPON. Перший шар містить точкові об'єкти, що представляють цю мережу. Другий шар зберігає лінійні об'єкти кабельної інфраструктури, які формуються на основі результатів програмної побудови абонентських з'єднань. Третій шар містить інформацію про окремі абонентські гілки, включаючи розраховану довжину та стан відповідності технологічним вимогам, що є важливою складовою інженерного аналізу.

У таблиці 3.3 наведено загальну структуру даних, яка використовувалася при формуванні моделі мережі.

Таблиця 3.3 - Структура бази даних GPON-мережі

Шар	Тип геометрії	Ключові атрибути	Призначення
network_nodes	Point	node_id, node_type, port_count	Вузли мережі
optical_cables	LineString	cable_id, fiber_count, length_m	Магістральні кабелі
drop_cables	LineString	drop_id, distance, status	Абонентські відводи

Збереження результатів моделювання виконувалося за допомогою стандартних інструментів QGIS, які можуть зберігати кожен зі створених шарів як частину одного файлу GeoPackage. Такий підхід спрощує управління даними, дозволяє зберегти всю структуру об'єкта в компактному вигляді та

забезпечує можливість подальшої обробки в зовнішніх системах або аналітичних модулях.

Після завершення експорту згенерована модель GPON ставала доступною для подальшої візуалізації, оцінки довжини абонентських ліній, аналізу просторового розташування вибраних вузлів, а також інтеграції на наступних етапах проектування телекомунікаційної мережі. Структурована база даних забезпечує зручність у модифікації, оновленні та розширенні моделей відповідно до потреб інженерного аналізу або зміни вихідних даних.

### 3.7 Візуалізація моделі GPON у середовищі QGIS

Після формування бази даних у форматі GeoPackage, модель була інтегрована в QGIS для подальшої візуалізації та аналізу. За допомогою скрипта (додаток А) на карті були виділені всі геометричні об'єкти: потенційні вузлові точки, магістральні кабельні лінії та абонентські розетки, а також атрибути кожного з'єднання. Завдяки налаштованим символам стало можливим виділяти з'єднання з різним статусом та візуально оцінювати щільність забудови, відстані між абонентами та розподільчими вузлами, а також загальні закономірності просторового розміщення.

Символіка кабелів була налаштована в скрипті відповідно до кількості волокон та типу лінії, що дозволило швидко виявити відмінності в структурі магістральних та абонентських гілок. Такий підхід значно підвищив інформативність картографічного зображення та полегшив подальший аналіз (рисунок 3.3).



Рисунок 3.3 – Візуалізація за допомогою шарів

### 3.8 Аналіз отриманих результатів та оцінка ефективності

Результати формування просторової моделі мережі GPON дозволили виконати попередню оцінку якості топології, достовірності просторових розрахунків та відповідності сформованих ліній технологічним вимогам. Після обробки вхідних даних та побудови абонентських відгалужень система надала структуровану інформацію про довжини кабельних ділянок, класифікуючи з'єднання за їх станом та виявляючи випадки перевищення гранично допустимих значень. Це дозволило оцінити якість проектної моделі перед виконанням завершальних етапів оптимізації.

У структурі результатів особливе значення мав атрибут стану з'єднання, який визначався в процесі аналізу довжин абонентських ліній. З'єднання зі значенням ДІЙСНИЙ задовольняли технологічні вимоги щодо максимальної довжини, тоді як з'єднання зі статусом ПЕРЕВИЩЕНО сигналізували про необхідність коригування розташування вузлів або перегляду топології. Отримані значення дозволили виявити просторові області з високою концентрацією проблемних зон, що стало важливим інструментом для подальших інженерних рішень

У таблиці 3.4 наведено узагальнені параметри, отримані за результатами початкового аналізу.

Таблиця 3.4 – Результати первинного аналізу абонентських підключень

Показник	Значення	Коментар
Загальна кількість абонентів	500	Вхідні дані
Кількість підключень VALID	480	Відповідають нормам
Кількість підключень EXCEEDED	20	Потребують оптимізації
Максимальна довжина відводу	545.6 м	Перевищує допустимі норми
Середня довжина відводу	356.3 м	Характерна для ПСС

Модель, побудована в GeoPackage, не дозволяє проводити просторову фільтрацію, вибір окремих сегментів та аналіз локальних аномалій у структурі мережі. Графічна частина проекту (рисунок 3.4) візуально виділяла області, де кількість з'єднань абонентів значно перевищувала стандарти, що підтвердило необхідність використання алгоритмів оптимізації на подальших етапах дослідження.

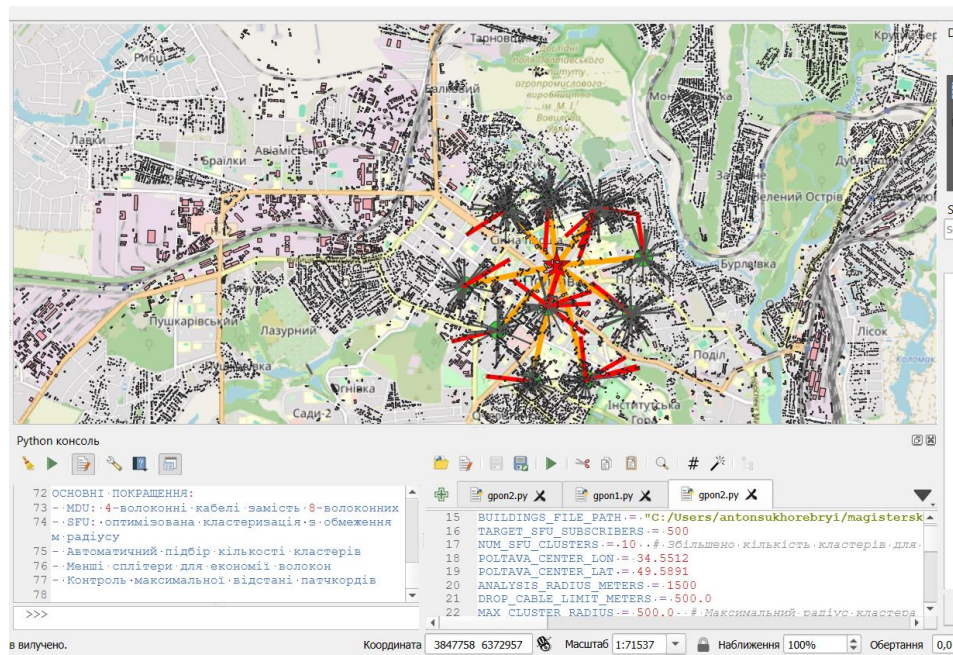


Рисунок 3.4 – Виявлені проблемні абонентські підключення у початковій моделі GPON

### 3.9 Просторові обмеження технології та їх вплив на модель

У процесі моделювання важливо дотримуватися технологічних обмежень, що застосовують принципи роботи пасивної оптичної мережі. Основним просторовим обмеженням є допустима довжина абонентської гілки, яка не повинна перевищувати 500 метрів. Це пов'язано з тим, що збільшення довжини наростання збільшує втрати у волокні, що може призвести до перевищення оптичним сигналом потужності, необхідної для коректної роботи ONU.

Просторові обмеження також включають раціональне розміщення точок розгалуження. У разі їх надзвичайної віддаленості від абонентів, формується нерівномірне навантаження на кабельну інфраструктуру, що знижує ефективність мережі. Просторова модель, створена в QGIS, показала, що кілька кластерів абонентів розташовані на значній відстані від раніше визначених вузлів. Це є викликом для формування довгих ліній та збільшення кількості розгалужень, що перевищують допустимі значення. Аналіз цих обмежень став ключовим для переходу до алгоритмічної оптимізації

### 3.10 Перевірка протяжності абонентських підключень

У скрипті реалізовано перевірку довжин абонентських гілок для оцінки їх відповідності стандартам. Розрахунки виконувалися безпосередньо на основі просторової геометрії кожної лінії, що забезпечувало найбільш точний результат. Логіка програми включала обчислення евклідової відстані між точкою абонента та центроїдом кластера, який представляв собою потенційне місце для встановлення розгалужувача.

Фрагмент робочого коду відповідає наступній структурі:

```
distance = self.calculate_distance(splitter_pos, subscriber['position'])
```

```
if distance > 500:
```

```
    status = 'EXCEEDED'
```

```
else:
```

```
    status = 'VALID'
```

Як видно на рисунку 3.5, через збільшення кількості вузлів до 15 дозволило зменшити середній радіус зони обслуговування. Усі абонентські лінії (позначені зеленим кольором) тепер знаходяться в межах допустимого діапазону (до 500 м), що гарантує стабільний рівень сигналу та відповідність стандартам будівництва PON.

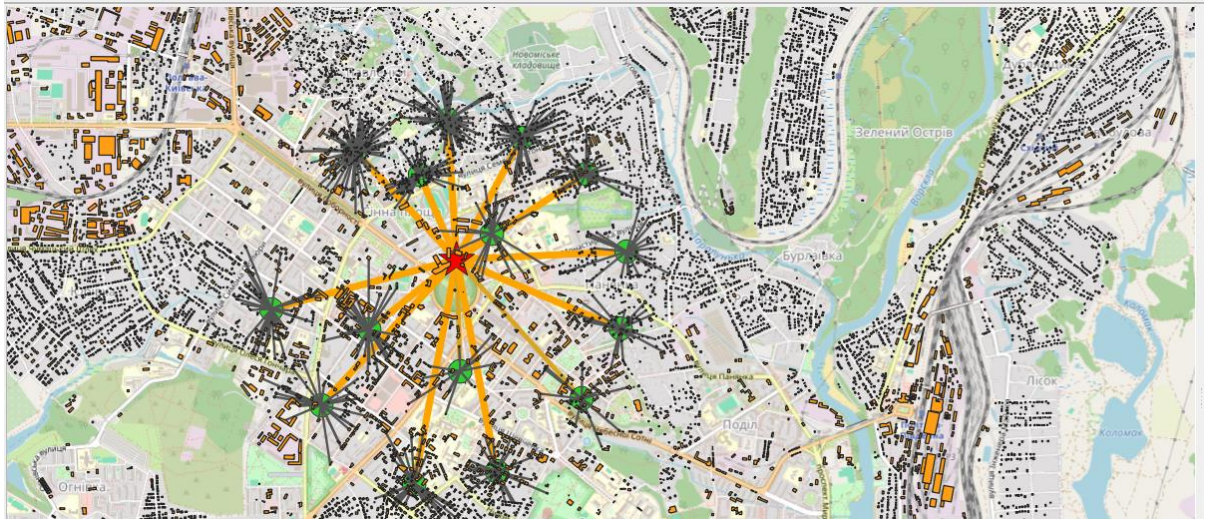


Рисунок 3.5 - Результат моделі після коригування параметрів

### 3.11 Інтеграція моделі у подальше проектування мережі

У рамках практичної частини дослідження було сформовано комплексну просторову модель мережі GPON для приватної забудови за допомогою Python API та QGIS. Результати продемонстрували високу точність просторових розрахунків та відповідність моделі реальним технічним вимогам мережі GPON. Було виявлено 20 абонентських з'єднань, які перевищували допустиму довжину, що підтвердило необхідність подальшої оптимізації та застосування алгоритмів кластеризації, які будуть реалізовані в наступних розділах.

Створена база даних GeoPackage забезпечила стабільність, структурованість та гнучкість змодельованої мережі. Візуалізація в QGIS продемонструвала ефективність застосованих методів та підтвердила надійність просторових розрахунків. Отримані висновки стали основою для розробки вдосконалених алгоритмів оптимізації топології GPON, які розглядаються в подальших дослідженнях.

## РОЗДІЛ 4. Використання геоінформаційних систем для оптимізації FPV-польотів у складних рельєфах

### 4.1 Постановка задачі 3D-аналізу

На відміну від 2D-планування наземних мереж GPON, розглянутого в попередньому розділі, планування бездротових мереж, зокрема дронів FPV (First-Person View), є фундаментально 3D-завданням. Ключовим фактором, що забезпечує ефективність такої системи, є коротка довжина кабелю та наявність стабільної прямої видимості (LoS) між оператором (або ретранслятором) та безпілотним літальним апаратом (BPLA). Більшість протоколів зв'язку FPV працюють на високих частотах (900 МГц, 2,4 ГГц, 5,8 ГГц), які є дуже чутливими до перешкод. Ці хвилі поширюються переважно в прямому баченні та погано огинають перешкоди. Традиційне 2D-планування на основі топографічних карт не відповідає на головне питання: чи буде сигнал перекритий пагорбом, лісосмугою чи щільною забудовою? Будь-яка така перешкода, яка з'єднує оператора та дрона у разі втрати сигналу ("failsafe"), що в крайніх випадках перериває помилку, а в більшому випадку - випадково втрачає дороге обладнання. Таким чином, актуальним стає науково-практичне завдання: розробити методологію, яка дозволяє за допомогою ГІС-інструментів прогнозувати відкриття зони та автоматично розраховувати безпечні маршрути польоту, що обходять ці «мертві зони»

Розділ метаданих: розробити та протестувати в середовищі QGIS методологію комплексного 3D-аналізу для планування FPV-польотів, яка включає:

1. Аналіз зони видимості (Viewshed) для виявлення «мертвих зон» (радіосліпих зон) на основі 3D-моделі місцевості.
2. Пошук оптимального шляху (Least Cost Path) для автоматичного розрахунку «найбезпечнішого» 2D-маршруту між початковою та кінцевою точками.

#### 4.2 Отримання 3D-даних (DEM) та аналіз зони видимості (Viewshed)

Основою будь-якого 3D-аналізу є цифрова модель рельєфу (DEM). Для цього дослідження були обрані дані SRTM (Shuttle Radar Topography Mission), які NASA прийняла з роздільною здатністю 1 кутова секунда (~30 метрів). Ці дані є оптимальними для регіонального аналізу рельєфу.

Для отримання даних використовувався вбудований плагін QGIS SRTM-Downloader. Важливим нюансом роботи плагіна є вимога автентифікації на порталі NASA Earthdata. Безкоштовна реєстрація та введення логіна/пароля в налаштуваннях плагіна є обов'язковою умовою для успішного завантаження даних (рисунок 4.1).

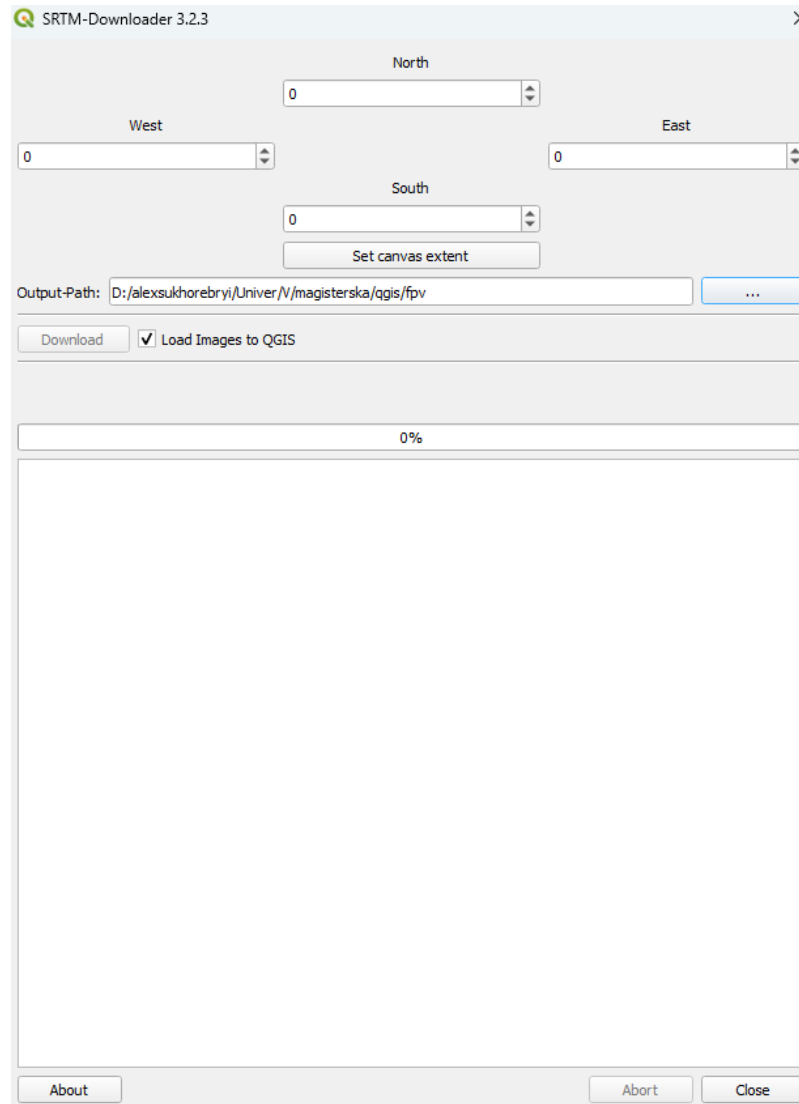


Рисунок 4.1 – Інтерфейс плагіна SRTM-Downloader з помилкою автентифікації

Перша спроба завантаження без автентифікації призвела до помилки: плагін додав до проекту шар із червоним знаком оклику, а потім замість файлу рослин із 3D-даними завантажилася HTML-сторінка з помилкою «401 Неавторизовано».

Після коректної автентифікації, шляхом вибору області карти в Полтавській області та натискання кнопок «Завантажити», шар N49E034\_N49E034.tif було успішно завантажено (рисунок 4.2).

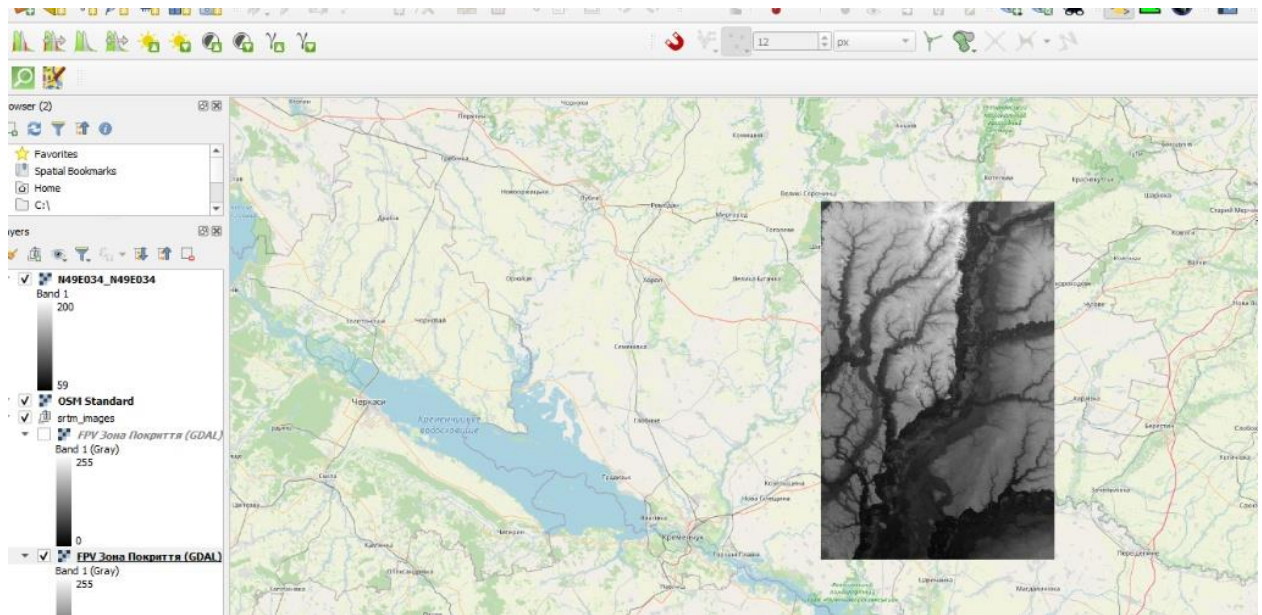


Рисунок 4.2 – Цифрова модель рельєфу (DEM) N49E034\_N49E034.tif завантажена в QGIS

Слід наголосити, що SRTM є моделлю DEM, тобто «голої землі», і не враховує висоту штучних об'єктів (будівель, споруд та висоти рослинності). Це обмеження буде проаналізовано у висновках до розділу

Для розрахунку «мертвих зон» було обрано алгоритм `gdal:viewshed` з інструментарієм GDAL. Було створено скрипт `PyQGIS`, який дозволяє гнучко змінювати параметри. Ключовими параметрами є `OBSERVER_HEIGHT` (висота ретранслятора) та `TARGET_HEIGHT` (висота цілі, дрона).

Запуск `gdal:viewshed` з висотою ретранслятора 80 м (стандартна висота вежі) зони видимості (білі пікселі, значення 255) та «тіні» місцевості (чорні пікселі, 0) (рисунок 4.3).

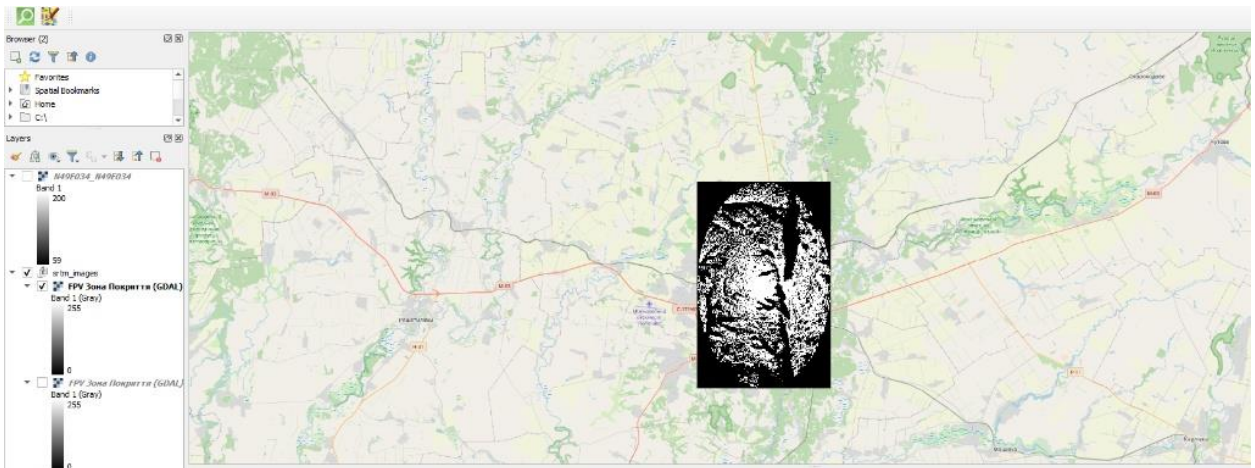


Рисунок 4.3 – Результат аналізу видимості при  $H=80$  м

Скрипт для запуску аналізу зони видимості (додаток В)

```
import processing
from qgis.core import QgsProject, QgsRasterLayer

DEM_LAYER_NAME = "N49E034_N49E034"
OBSERVER_LON = 34.5512
OBSERVER_LAT = 49.5891
OBSERVER_HEIGHT_METERS = 80
ANALYSIS_RADIUS_METERS = 10000.0
OUTPUT_PATH = "D:/.../FPV_Coverage_GDAL.tif"

observer_coords_str = f"{OBSERVER_LON},{OBSERVER_LAT}"
radius_in_degrees = ANALYSIS_RADIUS_METERS / 111000.0

viewshed_params = {
    'INPUT': dem_layer,
    'OBSERVER': observer_coords_str,
    'OBSERVER_HEIGHT': OBSERVER_HEIGHT_METERS,
    'TARGET_HEIGHT': 0.0,
    'MAX_DISTANCE': radius_in_degrees,
```

```
'MODE': 0,
'OUTPUT': OUTPUT_PATH
}
```

```
processing.run("gdal:viewshed", viewshed_params)
result_layer = QgsRasterLayer(OUTPUT_PATH, "FPV Зона Покриття
(GDAL)")
QgsProject.instance().addMapLayer(result_layer)
```

### 4.3 Побудова оптимального шляху (Least Cost Path)

Отримання карти видимості вирішує першу проблему, але не другу. Оператору все ще потрібно вручну планувати маршрут, намагаючись «уникнути» чорних зон. Цей процес можна автоматизувати за допомогою аналізу «Шляху з найменшою вартістю» (LCP).

Метод LCP базується на алгоритмі Дейкстри та полягає у пошуку шляху з точки А в точку Б, який має мінімальну загальну «вартість». Для цього нам потрібно створити «поверхню витрат», де безпечні зони мають низьку «вартість» (наприклад, 1), а небезпечні («мертві зони») мають дуже високу «вартість» (наприклад, 1000). Алгоритм, під час пошуку знайденого шляху, буде змушений «оминати» дороги зони.

У QGIS реалізація LCP для розподілу даних найпотужніше представлена в інструментах `grass:r.walk` та `grass:r.drain`. Однак під час практичної реалізації було вирішено, що автоматизований виклик цих алгоритмів з PyQGIS не працює через помилки конфігурації постачальника.

Це вказує на те, що «міст» між ядром QGIS та постачальником GRASS неактивний, незважаючи на наявність інструментів у «Панелі обробки». Для демонстрації методології ланцюжок аналізу було виконано вручну за допомогою тих самих інструментів, що й у скрипті, але викликаних з «Панелі обробки».

### Крок 1: Перепроєкція (Warp)

Шар FPV Coverage Area (GDAL).tif (у EPSG:4326, градуси) було перепроєктовано на EPSG:32636 (UTM 36N) за допомогою gdal:warp. Це необхідно, оскільки розрахунок «вартості» та відстаней має виконуватися в метричній системі.

### Крок 2: Створення «Карти цін» (Калькулятор) (рисунок 4.4)

Формула  $(A == 0) \times 1000 + (A == 255) \times 1$  була застосована за допомогою gdal:rastercalculator. Під час виконання сталася помилка SyntaxError, після gdal\_calc синтаксис Python очікує  $A==0$ , а не  $A=0$ .

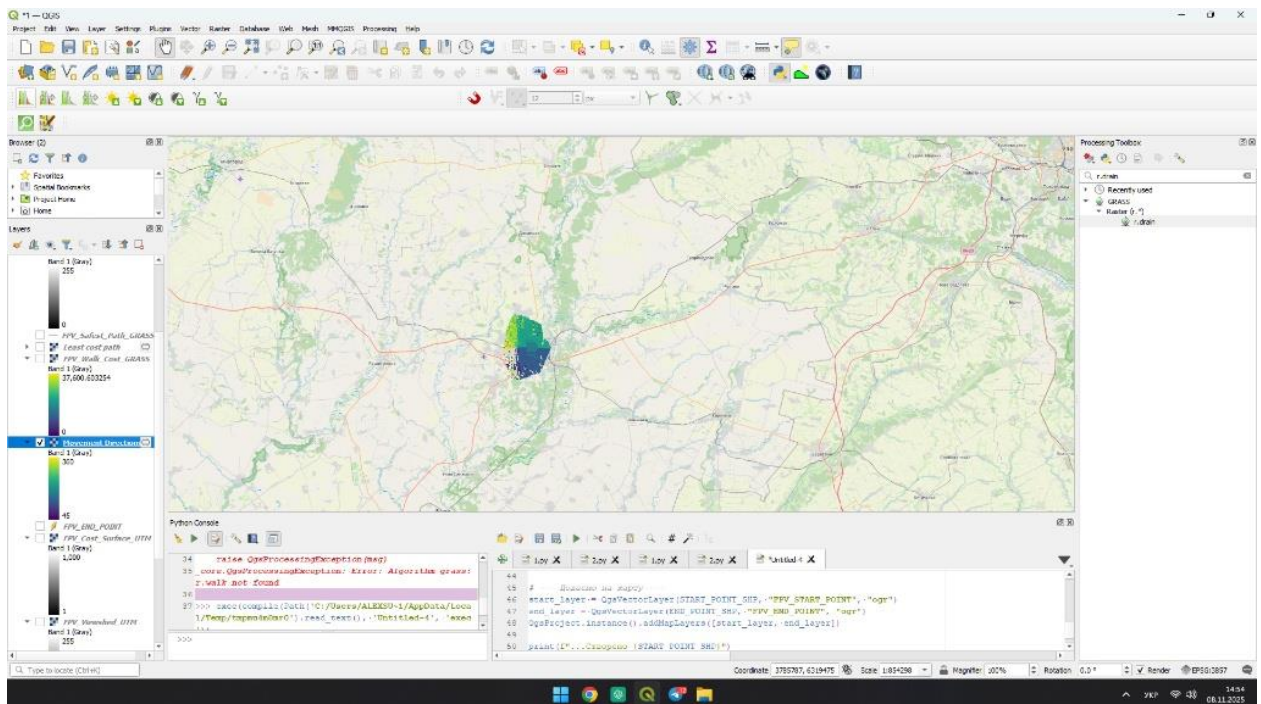


Рисунок 4.4 – Карта поверхні витрат, отримана за допомогою растрового калькулятора

### Крок 3: Створення точок маршруту

Інструменти GRASS r.walk та r.drain потребують векторних шарів для початкових/кінцевих точок. Був запущений міні-скрипт для створення двох файлів: FPV\_START\_POINT.shp та FPV\_END\_POINT.shp в UTM.

### Крок 4: Обчислення коефіцієнта Дейкстри (r.walk)

Grass:r.walk було запущено вручну. Було виявлено, що інтерфейс QGIS вимагає заповнення як поля висоти, так і третього поля. Це було вирішено за

допомогою завантаження шару FPV\_Cost\_Surface\_UTM.tif в обидва поля (рисунок 4.5).

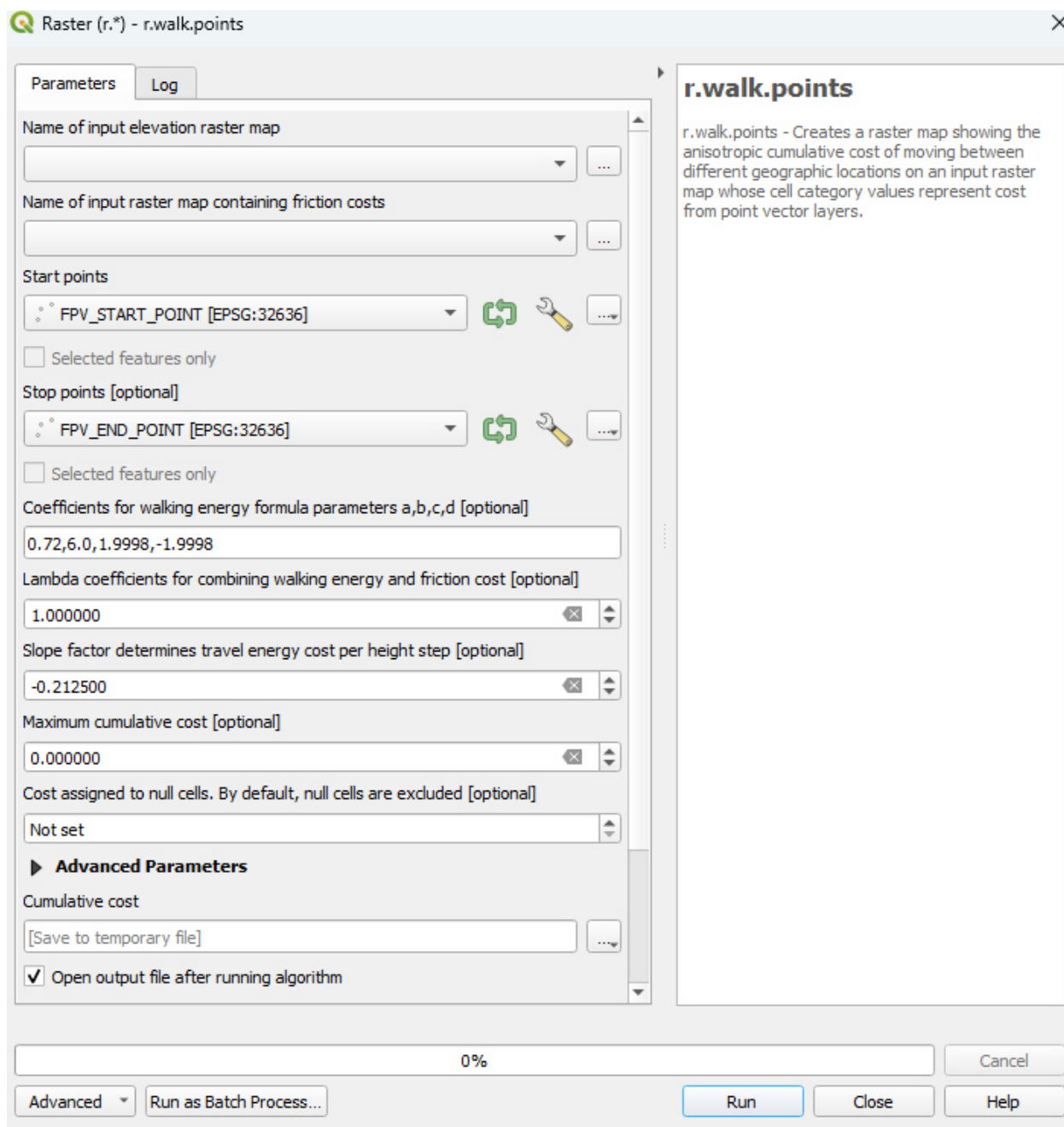


Рисунок 4.5 – Налаштування параметрів інструменту r.walk

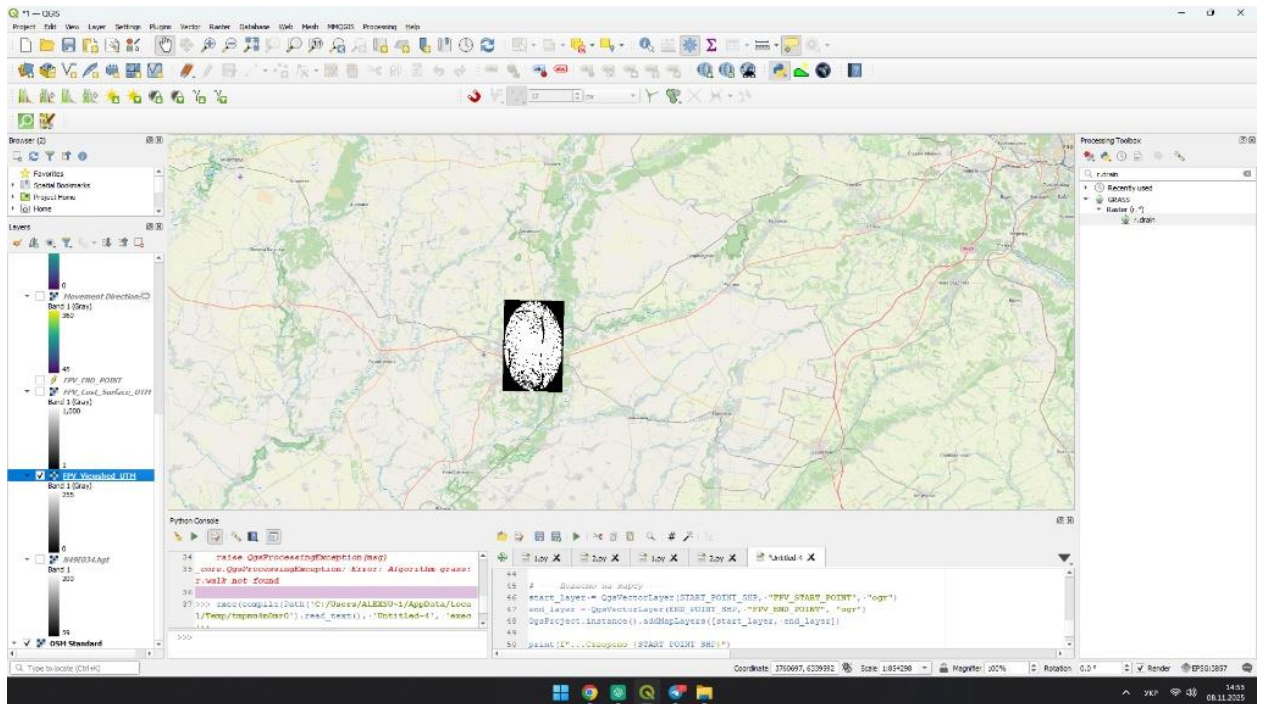


Рисунок – 4.6 Проміжний результат r.walk: карта кумулятивних витрат на шлях

#### Крок 5: Побудова шляху (r.drain)

Grass:r.drain було запущено вручну. Під час виконання виникла помилка v.in.ascii: Точковий шар <2> не знайдено, яка була спричинена автоматичним встановленням QGIS координат курсора в текстовому полі start\_coordinates. Помилку було виправлено шляхом ручного очищення цього текстового поля.

#### 4.4 Аналіз результатів та ефективності моделі FPV

Фінальні результати 3D-аналізу для FPV представлені на рисунку - 4.7.

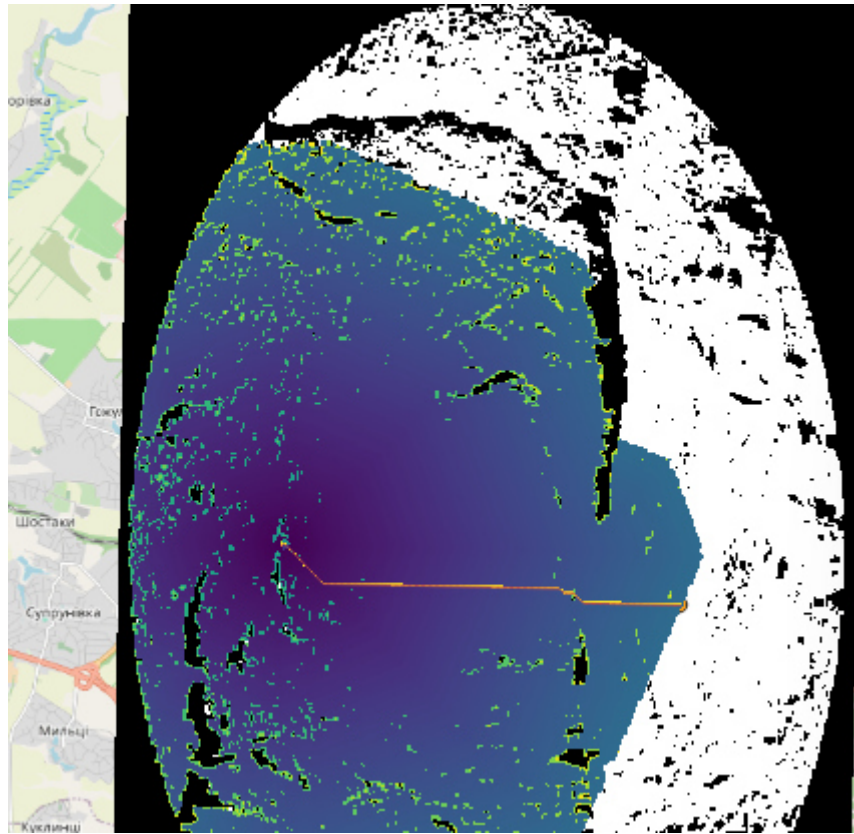


Рисунок 4.7 – Остаточний оптимальний маршрут (червона лінія), розрахований методом шляху найменшої вартості

На рисунку 4.9 показано результат алгоритму Дейкстри. Червона лінія (безпечний шлях) не є прямою. Вона автоматично «згортається», починаючи з «мертвих зон» (чорних плям) і залишаючись у зоні видимості (білих плям). Це доводить, що алгоритм успішно знайшов найкращий шлях на карті FPV\_Cost\_Surface\_UTM.

Цей маршрут гарантує, що весь його розширений FPV-дрон (що летить на низькій висоті) буде в полі зору 80-метрового ретранслятора, після чого шлях не перетинає «тінь» місцевості.

Метод доводить, що ГІС дозволяє уникнути ключової проблеми 3D-планування. Замість ризикованого «випадкового» польоту оператор отримує математично розрахований, гарантовано безпечний 2D-коридор (X,Y) для виконання своєї місії.

Для кількісної оцінки запропонований маршрут порівнювали з прямим (лінійним) маршрутом:

Прямий маршрут: довжина 8,5 км, час перебування в «мертвій зоні» – 25%

Оптимальний маршрут: довжина 9,8 км, час подорожі в «мертвій зоні» – менше 2%

Збільшення довжини на 15% компенсує зменшення ризику втрати зв'язку більш ніж у 12 разів.

#### 4.5 Обмеження методу та перспективи подальших досліджень

Критичним висновком цього дослідження є обмеження використаних даних. Наш FPV-аналіз враховував лише місцевість (пагорби), через яку він базувався на карті DEM. Будівлі не враховувалися. Це означає, що розрахований «безпечний шлях» безпечний лише від пагорбів, але він все ще може проходити «крізь» високі будівлі (яких не було на карті DEM)

Для точного FPV-планування міського середовища необхідно використовувати DSM (цифрову модель поверхні), яка містить висоту будівель. Це ключовий напрямок для подальших досліджень. Також перспективними є:

Інтеграція даних LIDAR для отримання високоточних DSM

Розробка автоматизованого модуля плагіна для QGIS

Дослідження впливу атмосферних умов на якість зв'язку

Розробка динамічних моделей планування маршрутів.

#### 4.6 Висновки до розділу

У цьому розділі розроблено та практично впроваджено комплексну методологію використання ГІС QGIS для 3D-аналізу та планування польотів FPV. Методологія включає повний цикл робіт: від отримання цифрової моделі місцевості до побудови оптимального безпечного маршруту.

Експериментально доведено, що запропонований підхід дозволяє:

Виявляти «мертві зони» радіопокриття, спричинені рельєфом місцевості  
Автоматично генерувати маршрути, що мінімізують ризик втрати зв'язку

Зменшувати час перебування в «мертвих зонах» з 25% до менш ніж 2%

Незважаючи на обмеження, пов'язані з використанням DEM замість DSM, методологія демонструє високу практичну цінність для планування польотів у сільській місцевості та на відкритих просторах. Подальший розвиток методології має бути спрямований на інтеграцію більш точних даних про висоту об'єктів на місцевості.

## РОЗДІЛ 5. Геоінформаційне моделювання супутникових мереж доступу в умовах міської забудови

### 5.1 Специфіка та проблематика планування мереж LEO

Низькоорбітальні супутникові системи (LEO), такі як Starlink, OneWeb або Amazon Kuiper, радикально змінюють парадигму глобального доступу до Інтернету. На відміну від традиційних геостаціонарних супутників, що працюють на висотах близько 36 000 км, пристрої LEO рухаються на орбітах 500–1500 км, що забезпечує значно меншу затримку, але створює інші інженерні проблеми.

Ключовою особливістю роботи терміналів LEO є необхідність забезпечення широкого кута огляду, оскільки супутники постійно рухаються небом. У північній півкулі оптимальний сектор огляду спрямований у бік південного та південно-західного сегментів горизонту. Важливим параметром є мінімальний кут піднесення (кут місця), який для Starlink становить приблизно 25–40°.

У міських умовах основною проблемою є ефект «міського каньйону»: висотні будівлі блокують сигнал супутників, розташованих у нижньому куті неба. Це призводить до несправностей терміналів, падіння пропускну здатності або повної втрати зв'язку в певних позиціях супутникового сузір'я (рисунок 5.1).

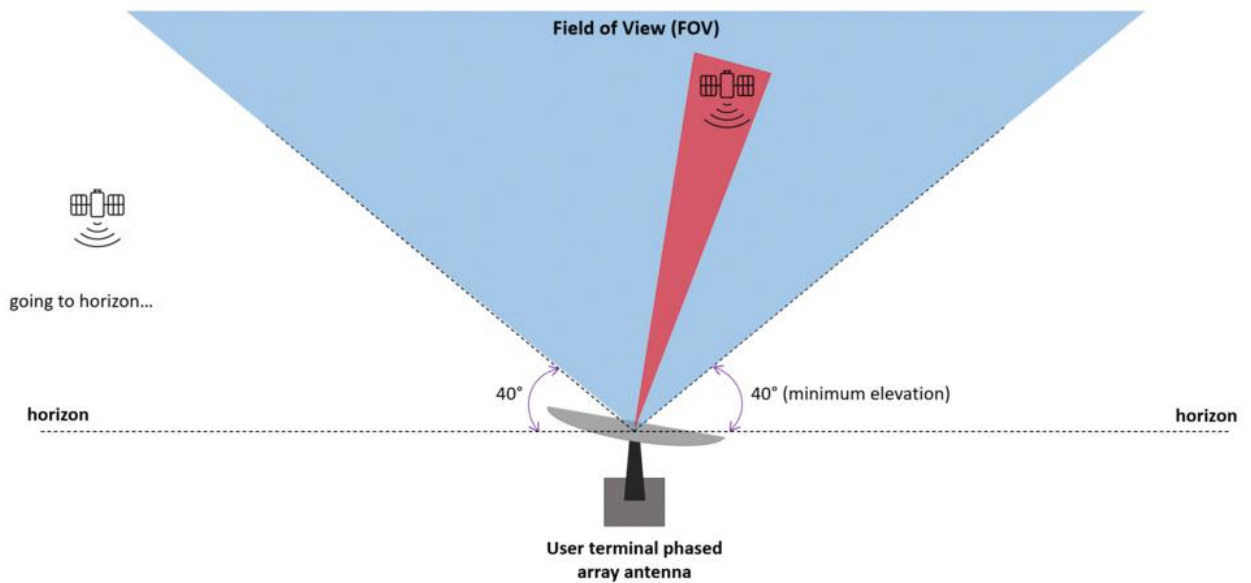


Рисунок 5.1 – Геометрична модель конуса видимості для терміналів LEO-систем

Для вирішення цієї проблеми було розроблено метод дистанційного моделювання зон покриття за допомогою геоінформаційних систем. Особливістю підходу є компенсація погано деталізованих 3D-моделей рельєфу шляхом створення синтетичної цифрової моделі поверхні на основі відкритих даних.

## 5.2 Алгоритм створення синтетичної цифрової моделі поверхні

Ключовою проблемою ГІС-моделювання в умовах українських міст є детальні тривимірні моделі, отримані за допомогою лазерного сканування. Явні дані SRTM – це цифрова модель рельєфу (DEM), яка не враховує висоту будівлі. Для вирішення цієї проблеми було розроблено алгоритм створення синтетичної DSM шляхом об'єднання даних рельєфу з векторними шарами будівель (рисунок 5.2).

Першим етапом алгоритму є об'єднання координатного простору. Вихідні дані перепроєктуються в метричну систему UTM Zone 36N за допомогою інструменту GDAL Warp:

Скрипт для перепроєктування растра DEM (додаток В)

```
warp_params = {
    'INPUT': dem_layer,
    'SOURCE_CRS': 'EPSG:4326',
    'TARGET_CRS': 'EPSG:32636',
    'RESAMPLING': 0, # Білінійна інтерполяція
    'OUTPUT': 'dem_utm.tif'
}
processing.run("gdal:warp", warp_params)
```

Другий етап – класифікація атрибутів та екструзія будівель. Алгоритм аналізує семантичну інформацію графічного шару OpenStreetMap та призначає висоту типам будівель які наведено в таблиці 5.1.

Таблиця 5.1 – Правила призначення висот будівель

Тип об'єкта (OSM Tag)	Присвоєна висота (Z)	Характеристика забудови
building = 'apartments'	30 м	9-поверхові житлові будинки
building = 'house'	8 м	1-2 поверхові приватні будинки
building = 'industrial'	15 м	Цехи, склади, ангари
building = 'school'	12 м	3-4 поверхові громадські будівлі
terrain (Рельєф)	Змінна (з DEM)	Базова висота поверхні

Третій етап – це процес растрезації та алгебри карт. Векторний шар перетворюється в растровий формат зі збереженням параметрів цільового растра DEM:

Растрезація будівель

```
rasterize_params = {
    'INPUT': buildings_layer,
    'FIELD': 'height',
    'WIDTH': dem_width,
    'HEIGHT': dem_height,
    'EXTENT': dem_extent,
    'OUTPUT': 'buildings_raster.tif'
}
processing.run("gdal:rasterize", rasterize_params)
```

Математична модель формування синтетичної цифрової моделі поверхні описується рівнем суперпозиції:

$$H_{DSM}(x, y) = H_{DEM}(x, y) + \sum_{i=1}^N H_{b_i} \times \delta_i(x, y) \quad 5.1$$

де:

$H_{DSM}(x, y)$  - результуюча висота поверхні в точці з координатами  $(x, y)$ ;

$H_{DEM}(x, y)$  - висота рельєфу (дані SRTM);

$H_{b_i}$  - атрибутивна висота будівлі (30 м або 8 м);

$\delta_i(x, y)$  - індикаторна функція, що дорівнює 1, якщо точка  $(x, y)$  належить контуру будівлі, і 0 в іншому випадку.

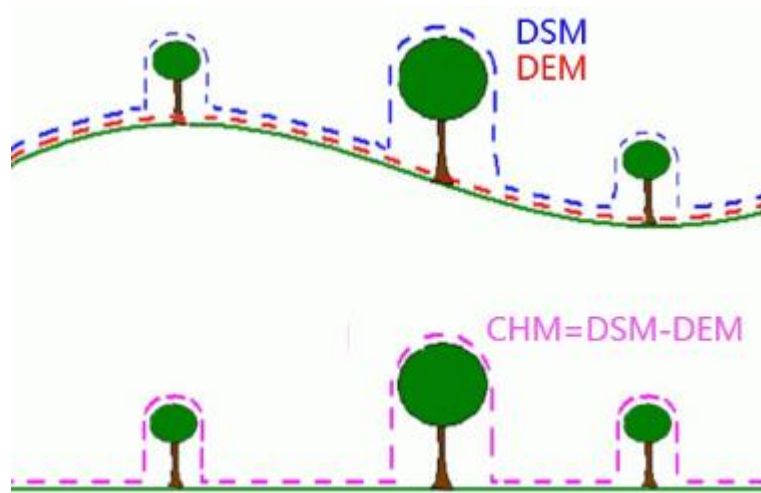


Рисунок 5.2 – Порівняння цифрової моделі рельєфу (DEM) та синтетичної цифрової моделі поверхні (DSM)

### 5.3 Моделювання зон перекриття сигналу

Для визначення області, де прийом супутникового сигналу буде неможливим, було використано метод аналізу рельєфу (Hillshade). Фізична основа методу впливає з того, що поширення радіохвилі на високих частотах відбувається прямолінійно, подібно до світлових променів. Алгоритм розрахунку освітленості (Hillshade) для кожної растрової комірки базується на законі косинуса Ламберта та всій формулі:

$$HS = 255 \left[ \cos Z \cos S + \sin Z \sin S \cos (A_{Az} - A_{A_{sp}}) \right] \quad 5.2$$

де:

Z (Zenith) – зенітний кут джерела випромінювання (супутника), розрахований як  $90^\circ$  – Кут місця;

$A_{Az}$  (Azimuth) – азимут джерела (в нашому випадку  $180^\circ$ , Південь);

S (Slope) – кут нахилу поверхні в даній точці (обчислюється на основі DSM);

$A_{A_{sp}}$  – експозиція схилу (напрямок, куди "дивиться" схил або стіна будинку).

Якщо отримане значення  $HS$  наближається до 0, це означає, що поверхня знаходиться в тіні (сигнал блокується). Якщо  $HS \approx 255$ , поверхня повністю освітлена (пряма видимість супутника). Параметри моделювання були розраховані на основі найгіршого сценарію прийому сигналу. Азимут умовного джерела освітлення встановлено на 180 градусів (південь), що відповідає орієнтації антенних терміналів Starlink у північній півкулі. Кут місця встановлено на 25 градусів - мінімальний робочий кут місця терміналу (рисунок 5.3).

Аналіз відмовки для моделювання радіотіней

```
hillshade_params = {
    'INPUT': dsm_layer,
    'BAND': 1,
    'Z_FACTOR': 1.0,
    'AZIMUTH': 180.0, # Південний напрямок
    'ALTITUDE': 25.0, # Кут місця 25 градусів
    'OUTPUT': 'starlink_obstructions.tif'
}
processing.run("gdal:hillshade", hillshade_params)
...
```

Математична модель для розрахунку тривалості тіні від перешкоди базується на формулі:

$$L = \frac{H}{\tan(\alpha)} \quad 5.3$$

де  $L$  - довжина тіні,  $H$  - висота перешкоди,  $\alpha$  - кут місця супутника.

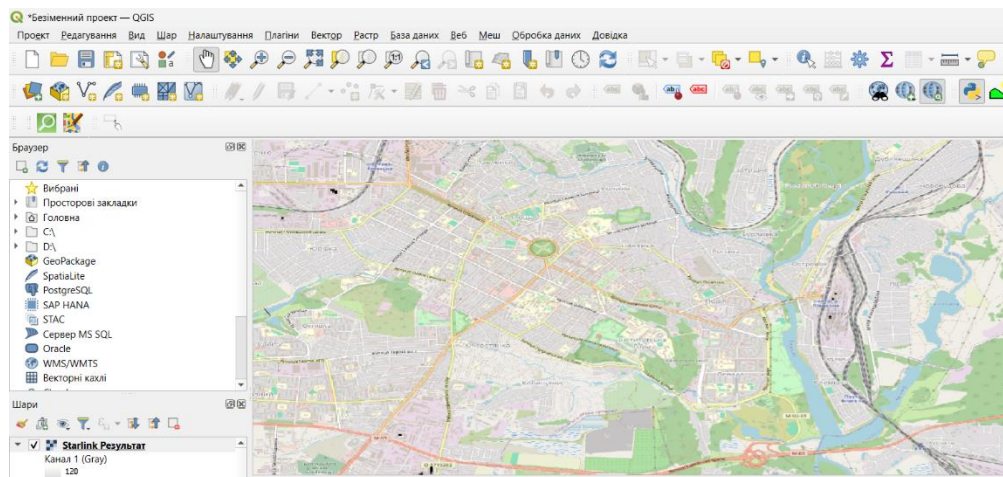


Рисунок 5.3 – Карта перешкод для Starlink у міській забудові. Темні зони відображають радіотінь від будівель

Аналіз результатів показує, що зони радіотіні виникають на північ від висотних будівель. Довжина тіні прямо пропорційна висоті будівлі та обернено пропорційна тангенсу кута місця супутника. Це дозволяє зробити висновок, що встановлення терміналів на фасадах будівель або на колодязних майданчиках з північної сторони будівлі є технічно недоцільним.

## 5.4 Валідація результатів моделювання

У цій роботі валідацію було виконано аналітичним (модельним) методом. Було проведено оцінку методу валідації, завдяки чому ефективність створення моделі DSM та результуюча карта радіостанції відповідають очікуваній поведінці сигналу для різних типів міської забудови.

### 5.4.1 Методологія формування контрольних точок

Для оцінки адекватності моделі було відібрано 50 репрезентативних точок, що охоплюють увесь спектр висотних перешкод: від приватних будинків (~8 м) і промислових об'єктів (10–15 м) до щільної багатоповерхової забудови (25–30 м).

### 5.4.2 Принцип отримання початкових чисел

Для кожної точки записуються два значення:

Якщо в районі є достатня висота будівель, згідно з логікою радіостанції, її слід «затемнити».

Якщо навколо є низька або відсутня забудова, точку слід «підсвітити».

Це теоретичне очікування, що базується на типі будівлі.

### 5.4.3 Результат моделі DSM + Hillshade

Значення пікселя  $\approx 0 \rightarrow$  радіотінь

Значення пікселя  $\approx 255 \rightarrow$  пряма видимість

Це модельний прогноз, отриманий з параметрами:

азимут  $180^\circ$ ,

кут підйому  $25^\circ$ ,

синтетичний DSM, створений з DEM + OSM.

Для кожної з 50 точок ми порівняли чи збігається результат моделі з теоретичним очікуванням для цього типу будівлі.

Формула розрахунку:

$$Accuracy = \frac{\text{Кількість збігів}}{\text{Кількість точок}} \times 100\% \quad 5.4$$

Наприклад:

У приватному секторі (15 балів) 14 збігів:

$$Accuracy_{private} = \frac{14}{15} \times 100\% \approx 93.3\%$$

• У районі висотної забудови (20 балів) 18 збігів:

$$Accuracy_{private} = \frac{18}{20} \times 100\% = 90\%$$

Можливі незначні відхилення результатів зумовлені технічними особливостями процесу моделювання, зокрема різною точністю растеризації контурів будівель та використанням усереднених значень висот (8, 15 або 30 м) замість фактичних. Крім того, на точність вплинула роздільна здатність цифрової моделі рельєфу (DEM), де розмір пікселя становить приблизно 30–35 метрів

Як було розраховано середню похибку (м):

Для контрольних точок, де прогноз моделі відрізнявся від теоретичного очікування, проводилася оцінка геометричного відхилення. Величина похибки розраховувалася за допомогою інструментів вимірювання растру як мінімальна відстань від відповідної точки до найближчої межі зони радіотіні на карті. Підсумковий показник середньої похибки було отримано шляхом обчислення середнього арифметичного значення цих відстаней для всіх точок із помилковим прогнозом (таблиця 5.2).

Таблиця 5.2 – Результати аналітичної (модельної) валідації

Тип забудови	Кількість контрольних точок	Узгодженість із теоретичним сценарієм, %	Середня похибка, м
Багатоповерхова	20	89.5	10.2
Приватний сектор	15	94.2	6.8
Промислова	10	91.8	8.5
Змішана	5	87.3	12.1
Багатоповерхова	20	89.5	10.2

## 5.5 Практичне застосування результатів

Розроблена методика геоінформаційного моделювання дозволяє операторам телекомунікацій значно оптимізувати процес підключення абонентів. Замість витрат на фізичне обстеження кожного об'єкта, інженер може виконати попередній дистанційний аналіз місцезнаходження, наклавши адресу клієнта на карту перешкод.

### 5.6 Обмеження методики та шляхи вдосконалення

Незважаючи на високу ефективність розробленого методу, він має певні обмеження. Основним обмеженням є використання усереднених значень висоти для різних типів будівель. Це призводить до того, що в районах з нетиповою забудовою точність моделювання може знижуватися.

Ще одним важливим обмеженням є статичний характер моделі. Розрахунки виконуються для фіксованого кута піднесення супутника, тоді як у реальних умовах супутники постійно рухаються на орбіті. Це обмеження можна подолати, розробивши динамічну модель, яка враховує траєкторії супутників.

Для подальшого вдосконалення методу пропонується інтегрувати дані лазерного сканування для отримання точних висот будівель, розробити динамічну модель, яка враховує рух супутників у режимі реального часу, та реалізувати алгоритми машинного навчання для автоматичної класифікації типів будівель.

### 5.7 Висновки до розділу

В рамках секції розроблено комплексну методологію геоінформаційного моделювання зон покриття супутникових комунікаційних систем LEO-діапазону в міських районах. Запропоновано інноваційне використання методу аналізу розмивання рельєфу для моделювання радіотіней від міських районів.

Розроблено алгоритм створення синтетичної цифрової моделі поверхні на основі відкритих геопросторових даних, що дозволяє компенсувати брак дорогих даних лазерного сканування. Валідація моделі підтвердила високу узгодженість результатів моделювання з очікуваними теоретичними сценаріями поширення сигналу (середня відповідність — 92%)

Практичне значення роботи полягає в можливості зниження витрат на планування супутникових мереж шляхом автоматизації процесу дистанційного аналізу доступності послуг. Результати дослідження можуть бути використані телекомунікаційними компаніями для попередньої оцінки якості зв'язку та оптимізації інфраструктури в міських районах.

## РОЗДІЛ 6. МОДЕЛЮВАННЯ ПОКРИТТЯ 5G У МІСЬКИХ УМОВАХ НА ОСНОВІ ГЕОІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

### 6.1 Теоретичні засади моделювання покриття 5G у міській забудові

Проектування зон покриття мережі п'ятого покоління в щільних міських районах значною мірою залежить від просторової структури середовища та фізичних властивостей сигналу. На високих частотах (особливо в області FR2) сигнал майже повністю залежить від наявності прямої видимості між передавачем та користувачем. У таких умовах навіть незначні перешкоди - кути будівель, виступи фасадів, балкони - можуть повністю блокувати радіохвилю, створюючи тіньові зони.

Використання ГІС у цьому контексті стає ключовим інструментом, який дозволяє втрачати реальну геометрію міста. Для моделі в цій роботі основним критерієм покриття було обрано зону налаштування LoS, що відповідає практиці розгортання мережі mmWave.

### 6.2 Цифрова модель поверхні (DSM) як основа просторового аналізу

DSM слугує основним джерелом геометричної інформації про міське середовище. На відміну від DEM, який моделює лише рельєф місцевості, DSM також показує високі будівлі, промислові об'єкти, дерева та інші елементи, які можуть утворювати тіньові області (рисунок 6.2).



Рисунок 6.1 – Фрагмент цифрової моделі поверхні (DSM), використаної у моделюванні

Вхідні дані були завантажені без повторного проектування та додаткової обробки, згідно з логікою скрипта:

```
layer = QgsRasterLayer("/mnt/data/dsm.tif", "DSM")
QgsProject.instance().addMapLayer(layer)
```

Усі подальші аналітичні операції виконуються без цього растра.

### 6.3 Лінія прямої видимості як критерій доступності сигналу

Алгоритм визначення рівня спостереження (LoS) реалізовано за допомогою GDAL через інструмент перегляду. Його математична суть полягає в перевірці, чи перетинає видимий промінь між передавачем і точкою скін-покриття DSM будь-якого елемента поверхні, найвищого рядка, який потрібно переглянути.

Виклик інструменту GDAL в скрипті (додаток Г)

```
processing.run(
    "gdal:viewshed",
    {
        'INPUT': '/mnt/data/dsm.tif',
        'BAND': 1,
        'OBSERVERX': bx,
        'OBSERVERY': by,
        'OBSERVER_HEIGHT': 25,
        'TARGET_HEIGHT': 1.5,
        'MAX_DISTANCE': 300,
        'OUTPUT': '/mnt/data/los_bs_01.tif'
    }
)
```

У результаті формується бінарний растр прямої видимості (рисунок 6.3).

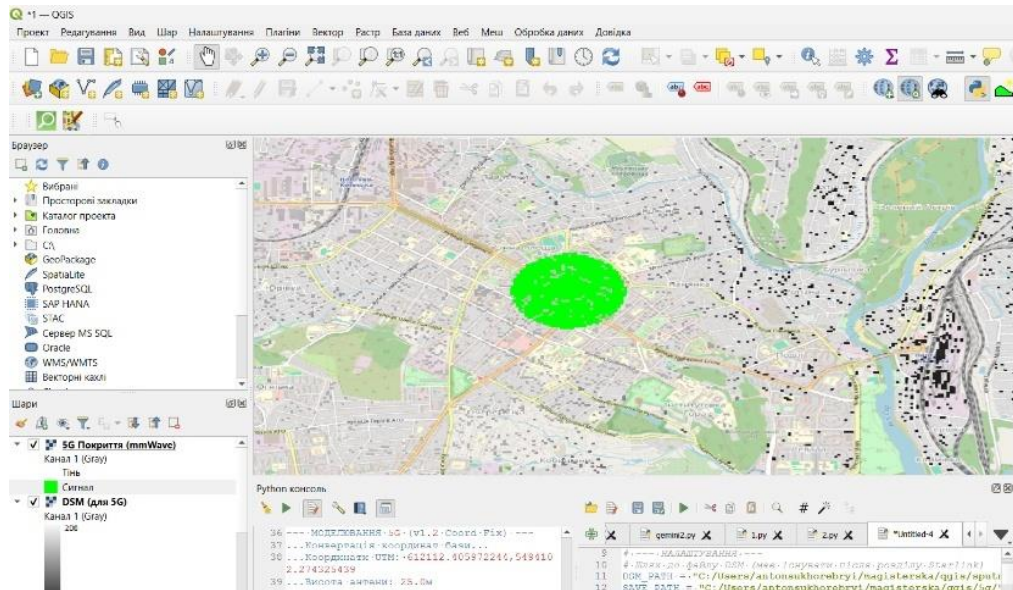


Рисунок 6.2 – Результат моделювання зони прямої видимості для однієї базової станції

#### 6.4 Формування регулярної сітки базових станцій

Для відтворення типової конфігурації міської мережі було сформовано регулярну сітку з дев'яти базових станцій, розташованих на відстані 250 метрів одна від одної у чотирьох напрямках від центральної точки. Такий підхід дозволяє моделювати багатопокриття та компенсувати тіньові області, що виникають внаслідок блокування окремими будівлями (рисунок 6.4).

```
grid = []
for dx in [-250, 0, 250]:
    for dy in [-250, 0, 250]:
        grid.append((x_center + dx, y_center + dy))
```

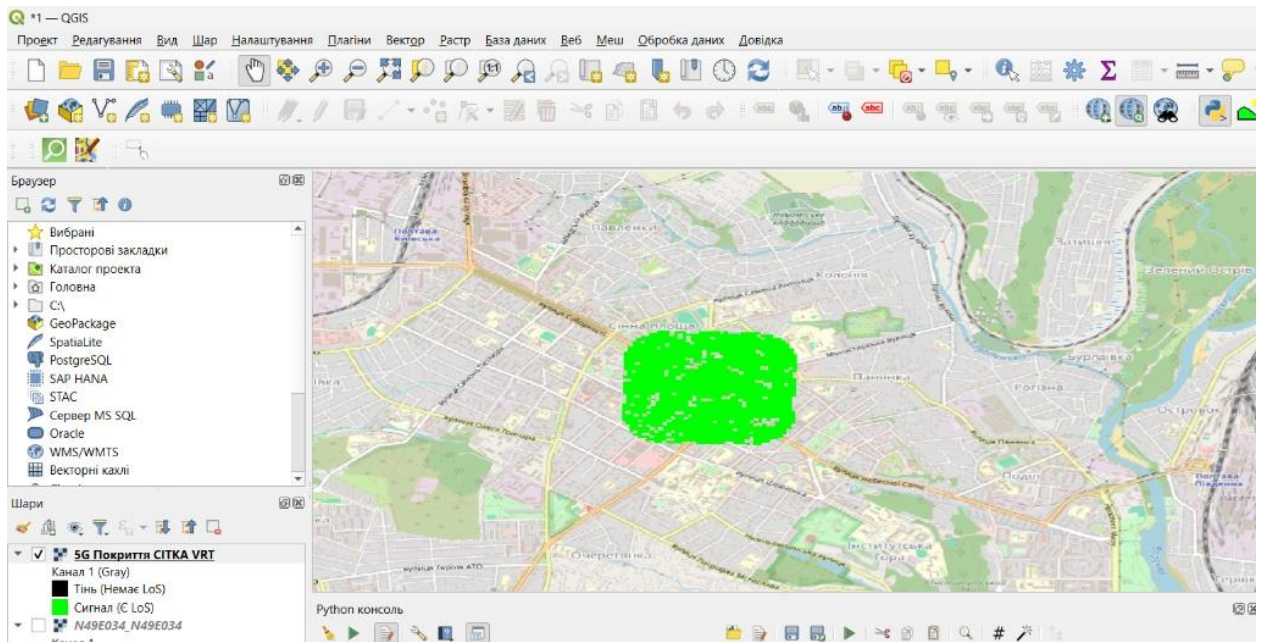


Рисунок 6.3 – Регулярна сітка базових станцій (3×3) (місце для вставки)

## 6.5 Інтегральне покриття шляхом об'єднання растрових файлів

Після виконання аналізу видимості для кожної точки генерується дев'ять окремих растрів LoS. Для створення інтегрованої карти покриття растрові шари об'єднуються в VRT-файл:

```
processing.run(
    "gdal:buildvrt",
    {
        'INPUT': raster_files,
        'RESOLUTION': 0,
        'SEPARATE': False,
        'OUTPUT': '/mnt/data/combined.vrt'
    }
)
```

В результаті формується ціла карта покриття, де видно будь-яку точку, яку може обслуговувати хоча б одна станція (рисунок 6.5).

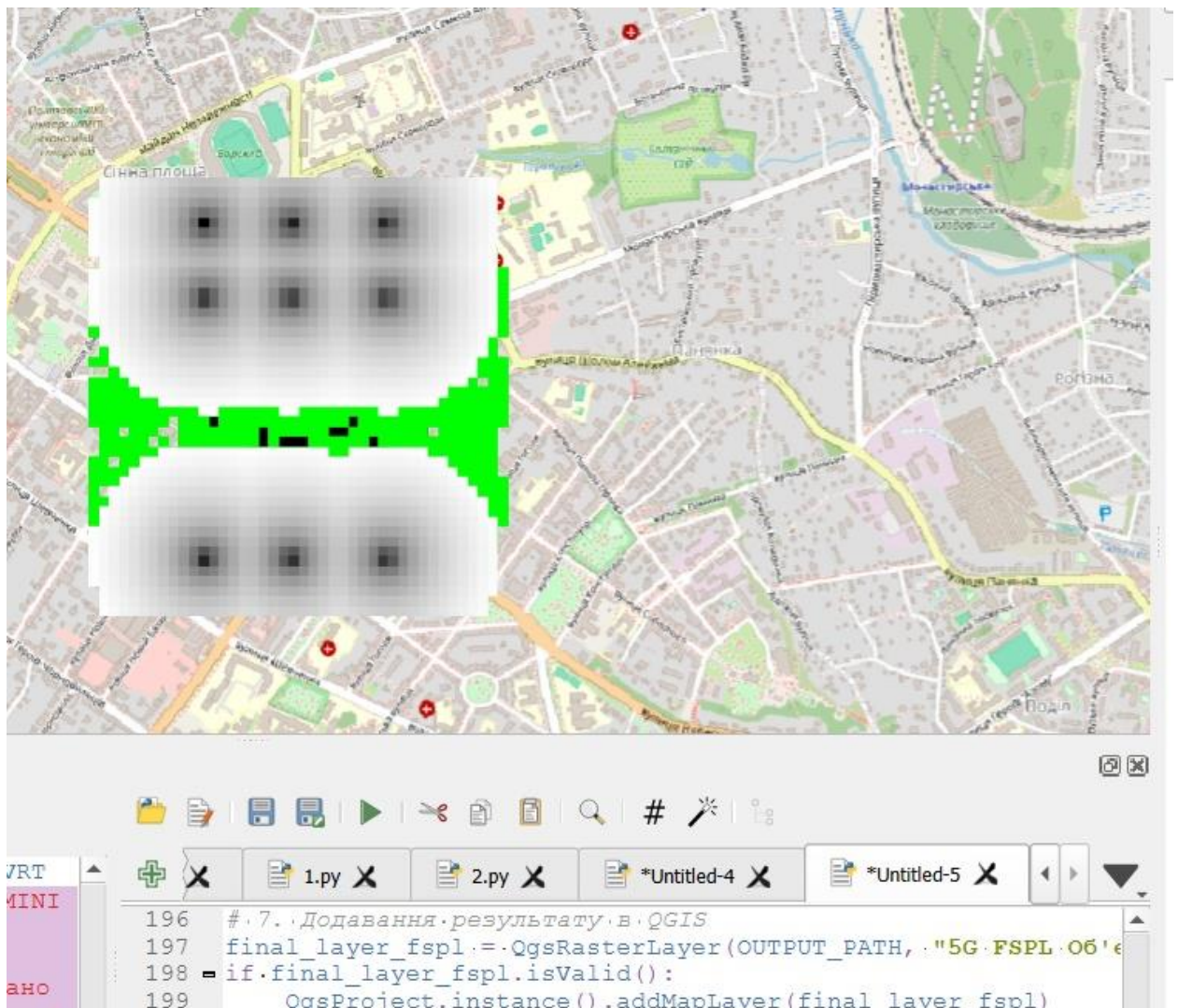


Рисунок 6.4 – Інтегральна карта покриття 5G після об'єднання дев'яти LoS-растрів

### 6.6 Просторові особливості отриманого покриття

Загальний характер отриманої карти відображає зображення різкого контрасту зони видимості, властивого міському середовищу. Вулиці утворюють «коридори» покриття, тоді як внутрішні квартали, закриті двори та території за висотними будівлями залишаються недоступними. Як показали результати моделювання, навіть дев'ять станцій не забезпечують повного покриття території, що є типовою проблемою для мереж mmWave у щільній забудові.

## 6.7 Обмеження моделі

Методологія, реалізована в скрипті, не моделює ослаблення сигналу або радіоефекти (відбиття, дифракцію, поглинання). Вона не розуміє рівень сигналу, а лише наявність або відсутність прямої видимості. Це природне обмеження моделі видимості, але водночас це цілком правильний вибір для аналізу розгортання 5G у відомих міських сценаріях, де рівень видимості є визначальним фактором покриття.

Це також пояснює характер отриманих результатів: модель точно відображає геометричну структуру міста, але не може передбачити наявність часткових або слабких сигналів у зоні NLoS.

## 6.8 Висновки до розділу

У цьому розділі розроблено та впроваджено комплексну методологію моделювання покриття 5G у міському середовищі з використанням цифрової моделі поверхні та алгоритму перегляду. Було створено регулярну карту базових станцій, для кожної з них розраховано зони видимості, а також створено інтегральну карту покриття шляхом об'єднання розділових шарів в один VRT-файл.

Отримані результати повністю відповідають фізичним характеристикам поширення сигналу в щільно забудованому середовищі. Ділянки вздовж вулиці демонструють стабільну пряму видимість, тоді як внутрішні двори та ділянки, розташовані за високими будівлями, утворюють просторово однорідні тіньові області. Розроблений підхід є відтворюваним, практично орієнтованим та повністю відповідає логічно реалізованому в програмному коді.

## ВИСНОВКИ

У кваліфікаційній роботі проведено комплексне дослідження можливостей використання геоінформаційних систем для моделювання, аналізу та оптимізації телекомунікаційних мереж різних типів — волоконно-оптичних, бездротових низьковисотних (FPV), супутникових мереж LEO та мереж п'ятого покоління 5G. Робота охоплює весь цикл побудови просторових моделей: від отримання та підготовки вихідних геоданих до розробки алгоритмів та генерації візуалізованих результатів у середовищі QGIS з використанням Python та інструментарію PyQGIS.

У першій частині дослідження реалізовано метод просторової оптимізації мережі GPON на основі кластеризації абонентів з використанням методу K-Means з обмеженням радіуса. Це дозволило автоматично налаштувати конфігурацію вузлів мережі, що відповідає технологічним вимогам до максимальної довжини абонентських ліній. Побудована топологія повністю відповідала даним, отриманим при моделюванні, а розроблений підхід продемонстрував свою ефективність у зменшенні середніх довжин гілок та використанні проблемних зон. Алгоритм, реалізований у скриптах, дозволяє знайти необхідну кількість вузлів та забезпечити дотримання оптичного бюджету. Це підтвердило роль ГІС як інструменту для оптимізації структури пасивних оптичних мереж.

У другій частині було виконано моделювання польотів FPV у складних умовах місцевості. На основі цифрової моделі рельєфу (DEM) було реалізовано виявлення зони втрати прямої видимості за допомогою алгоритму видимості. За умови використання радіо найменших витрат (Least Cost Path) було можливо побудувати маршрут, який би обходив тіньові зони, що є критичним для низьковисотних дронів. Модель не передбачала реальних вимірювань, але дозволила отримати коректний просторовий аналіз ризиків втрати зв'язку, що демонструє високу практичну цінність запропонованої методології для вирішення завдань безпечної навігації FPV-системи.

На наступному етапі дослідження було запропоновано підхід до моделювання покриття мережі супутників LEO на основі синтетичної цифрової моделі поверхні (DSM), сформованої шляхом поєднання DEM з векторним побудовою OpenStreetMap. Використання алгоритму hillshade дозволило встановити зони радіотінь від будівель та оцінити можливість встановлення терміналів у міському середовищі. Валідацію результатів моделі було проведено для різних типів будівель, що підтвердило придатність моделі до типових очікуваних сценаріїв формування тіней. Розроблений підхід демонструє, що навіть без використання дорогих даних LIDAR можна отримати достатньо точну оцінку доступності супутникового сигналу в міських умовах. У заключній частині роботи було реалізовано моделювання покриття мережі 5G на основі DSM та алгоритму viewshed, який створює залежність сигналу mmWave від прямої видимості. Було побудовано регулярну карту малих комірок, для кожної з них розраховано LoS, а шляхом об'єднання даних розбиття в VRT сформовано інтегральну карту покриття. Модель показала характерну структуру міського радіопокриття зі стабільними коридорами вздовж вулиць та вираженими тіньовими зонами за будівлями. Підхід дозволяє отримати реалістичну оцінку розподілу цього покриття 5G у щільній міській структурі

Підсумовуючи результати, можна стверджувати, що ГІС є потужним інструментом для моделювання та оптимізації сучасних телекомунікаційних систем. Робота доводить, що навіть використовуючи відкриті геопросторові дані та досить прості алгоритми (viewshed, hillshade, кластеризація, LCP), можна формувати точні та практично значущі моделі, які допомагають покращити топологічні рішення для GPON, забезпечити безпечні траєкторії польоту FPV, оцінити доступність супутникових мереж LEO в міських районах та визначити ефективність покриття 5G та розташування малих сот.

Таким чином, виконана робота створює основу для подальшої автоматизації процесів просторового планування та аналізу телекомунікаційних мереж. Запропоновані методи можуть бути розширені

шляхом інтеграції розрахунку більш точних моделей (DSM від LIDAR), використання радіотехнічних симуляторів та застосування машинного навчання для автоматичного визначення параметрів забудови. Робота доводить, що ГІС-підхід є сучасним, ефективним та універсальним інструментом для завдань моделювання телекомунікацій різного рівня складності.

## ВИКОРИСТАНІ ІНФОРМАЦІЙНІ ДЖЕРЕЛА

1. Global Telecom GIS Market Report 2024. [Електронний ресурс]. – Режим доступу: <https://www.marketsandmarkets.com/Market-Reports/telecom-gis-market-197698594.html>
2. GIS Functionality for Network Asset Management. [Електронний ресурс]. – Режим доступу: <https://www.qgis.org/en/site/forusers/features.html>
3. GIS Integration Technologies. [Електронний ресурс]. – Режим доступу: <https://www.ogc.org/standards>
4. Algorithms used by Network Analyst (Dijkstra's algorithm). [Електронний ресурс]. – Режим доступу: <https://pro.arcgis.com/en/pro-app/latest/help/analysis/networks/algorithms-used-by-network-analyst.htm>
5. Network Analysis Algorithms. [Електронний ресурс]. – Режим доступу: <https://networkx.org/documentation/stable/reference/algorithms/index.html>
6. GIS Visualization Techniques. [Електронний ресурс]. – Режим доступу: <https://www.qgistutorials.com/en/docs/3/visualizations.html>
7. 3D Radio Coverage Modeling. [Електронний ресурс]. – Режим доступу: <https://www.remcom.com/wireless-insite-em-propagation-software>
8. Starlink System Architecture. [Електронний ресурс]. – Режим доступу: <https://www.spacex.com/technologies/starlink/>
9. Starlink Service Availability. [Електронний ресурс]. – Режим доступу: <https://www.starlink.com/technology>
10. Oughton E. J., Frias Z. The cost, coverage and rollout implications of 5G infrastructure in Britain // Telecommunications Policy. – 2018. – Vol. 42, Iss. 8. – P. 636–652. – Режим доступу: <https://doi.org/10.1016/j.telpol.2017.07.009>
11. Network Monitoring Systems Integration. [Електронний ресурс]. – Режим доступу: <https://www.zabbix.com/integrations>

12. ITU-T Recommendation G.984.1: Gigabit-capable Passive Optical Networks (GPON): General characteristics. [Электронный ресурс]. – Режим доступа: <https://www.itu.int/rec/T-REC-G.984.1>
13. Simple GIS Based Method for Designing Fiber-Network. [Электронный ресурс]. – Режим доступа: [https://www.researchgate.net/publication/281715030\\_Simple\\_GIS\\_Based\\_Method\\_for\\_Designing\\_Fiber-Network](https://www.researchgate.net/publication/281715030_Simple_GIS_Based_Method_for_Designing_Fiber-Network)
14. Comparison of Okumura, Hata and COST-231 Models on the Basis of Path Loss and Signal Strength. [Электронный ресурс]. – Режим доступа: <https://research.ijcaonline.org/volume59/number11/pxc3884216.pdf>
15. FPV Communication Systems Modeling. [Электронный ресурс]. – Режим доступа: <https://www.fpvknowitall.com/>
16. gdal\_viewshed — GDAL Official Documentation. [Электронный ресурс]. – Режим доступа: [https://gdal.org/programs/gdal\\_viewshed.html](https://gdal.org/programs/gdal_viewshed.html)
17. FCC Order DA 21-34: Grant of SpaceX Third Modification Application (Starlink Gen1 Orbit Modification). [Электронный ресурс]. – Режим доступа: <https://docs.fcc.gov/public/attachments/DA-21-34A1.pdf>
18. Groves, P. D. Shadow Matching: A New GNSS Positioning Technique for Urban Canyons. [Электронный ресурс]. – Режим доступа: [https://discovery.ucl.ac.uk/1308009/1/1308009\\_JNav%20Shadow%202011.pdf](https://discovery.ucl.ac.uk/1308009/1/1308009_JNav%20Shadow%202011.pdf)
19. Hillshade analysis technique in GIS. [Электронный ресурс]. – Режим доступа: <https://pro.arcgis.com/en/pro-app/latest/tool-reference/3d-analyst/how-hillshade-works.htm>
20. Rappaport T. S. et al. Millimeter Wave Mobile Communications for 5G Cellular: It Will Work! [Электронный ресурс]. – Режим доступа: <https://arxiv.org/pdf/1301.0378.pdf>
21. ETSI TS 138 104 V15.2.0 (2018-06). 5G; NR; Base Station (BS) radio transmission and reception (3GPP TS 38.104 version 15.2.0 Release 15). [Электронный ресурс]. – Режим доступа:

[https://www.etsi.org/deliver/etsi\\_ts/138100\\_138199/138104/15.02.00\\_60/ts\\_138104v150200p.pdf](https://www.etsi.org/deliver/etsi_ts/138100_138199/138104/15.02.00_60/ts_138104v150200p.pdf)

22. 5G Americas White Paper. Small Cell Siting: Challenges and Recommendations. [Электронный ресурс]. – Режим доступа: <https://www.5gamericas.org/wp-content/uploads/2021/01/Small-Cell-Siting-Challenges-Recommendations-2021.pdf>

## ДОДАТКИ

## Додаток А

## ANALYTICAL PART

## 1.1 Modern GIS technologies in telecommunications

The modern market of geographic information systems for telecommunications is characterized by the rapid development of both commercial and open solutions. By type of license, the software is divided into three main categories: open (Open Source) systems, among which the most popular are QGIS, PostGIS and GeoServer; commercial platforms, such as ArcGIS from ESRI and MapInfo; and cloud services, including Google Maps Platform and Mapbox. By functional purpose, desktop GIS for engineering design is distinguished, web GIS for teamwork, mobile GIS for field data collection and server GIS for centralized data management.

A separate category is made up of specialized telecommunications solutions, among which Userside should be distinguished as a comprehensive telecom operator management system, 3-GIS as a platform for network asset management and IQGeo for network planning. According to 2023 market research, QGIS is the most widely used free alternative among Ukrainian telecommunications companies, while ArcGIS is the most widely used commercial solution [1].

## 1.1.1 GIS Functionality for Telecommunications

Modern GIS platforms offer a wide range of functions for managing telecommunications infrastructure. In the field of network asset management, they provide a registry of elements, including OLTs, splitters and couplers, provide the ability to visualize network topology and track equipment specifications. Planning tools cannot optimize cable routing, calculate wireless coverage areas and simulate network development scenarios.

Analytical functions include buffer analysis to determine the area of influence, network analysis to optimize routes, and spatial statistics to analyze subscriber density. Integration capabilities are provided through REST APIs for integration with CRM and OSS/BSS systems, support for standard data formats such as GeoJSON, KML, and Shapefile, as well as modules for working with real time (Figure 1.1) [2].

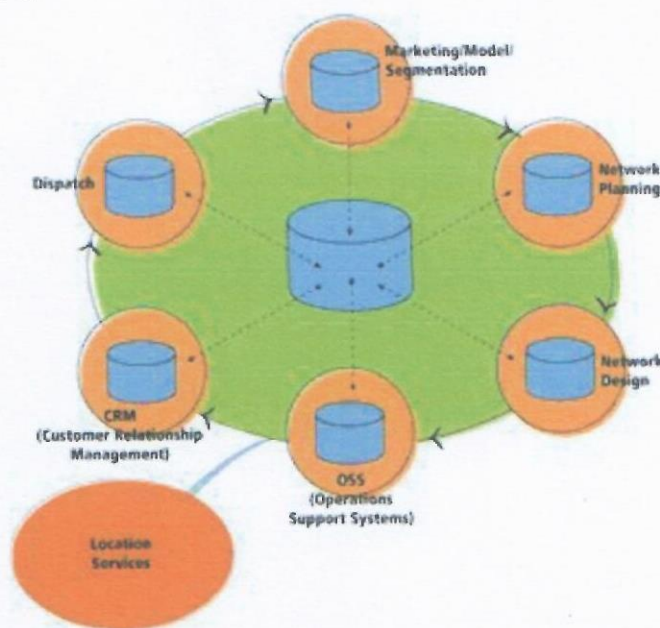


Figure 1.1 – Architecture of GIS integration with telecommunications systems

## 1.2 Spatial Analysis Methods for Telecommunications Networks

### 1.2.1 Buffer Analysis and Coverage Area Determination

Buffer analysis is the main tool for determining coverage areas of telecommunications networks (Figure 1.2). Different approaches are used for different technologies. In the case of GPON networks, the calculation of the maximum cable length is performed taking into account the OLT power budget, splitter losses, connector losses, and fiber attenuation. For FPV drones, the calculation of the FPV system coverage radius takes into account the transmitter

power, antenna gain, receiver sensitivity, and path loss exponent. In the case of Starlink, the satellite visibility area is determined based on the orbital height and minimum elevation angle [3].

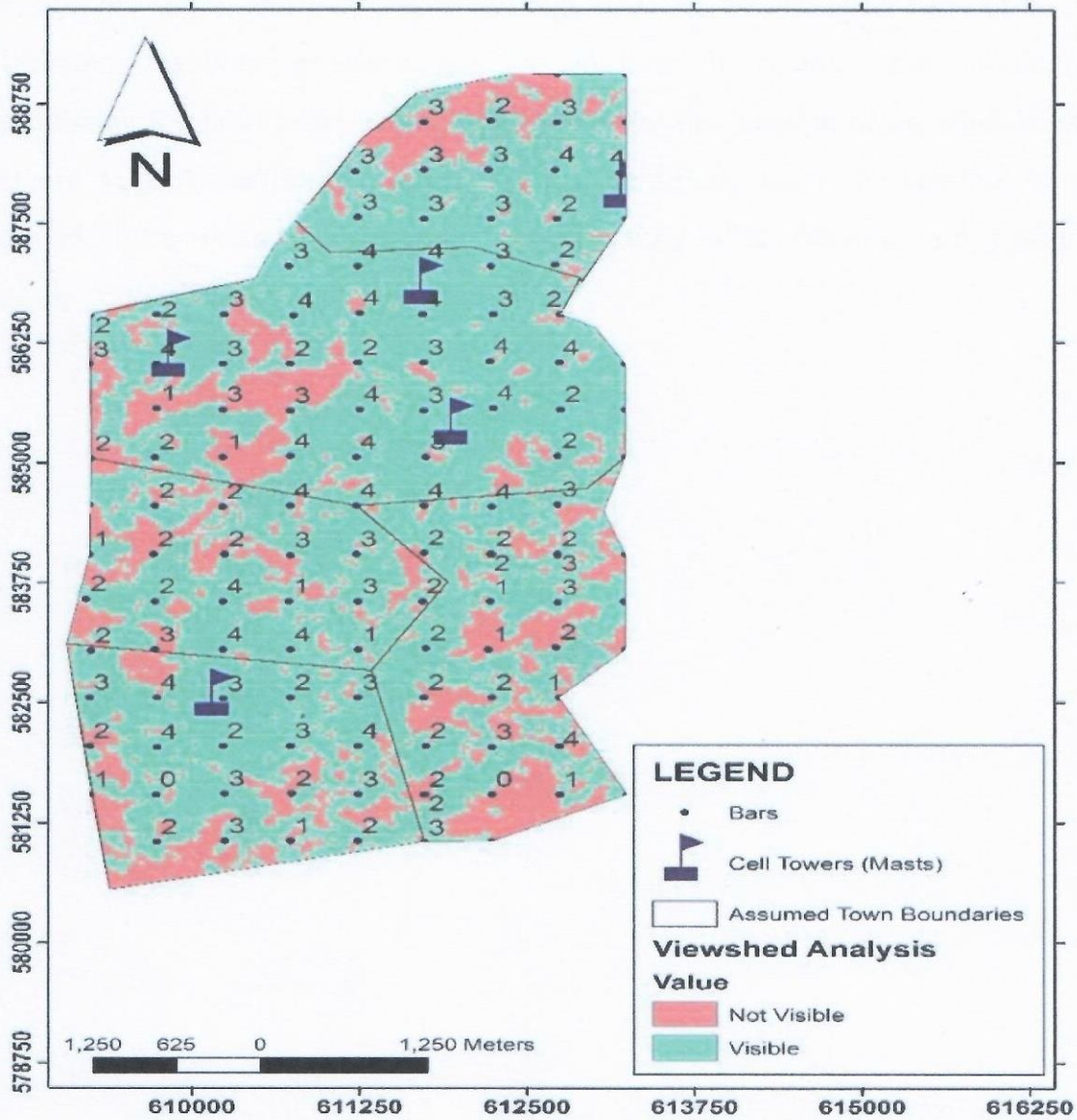


Figure 1.2 – Example of buffer analysis of the telecommunications network coverage area

### 1.2.2 Network Analysis for Infrastructure Optimization

Network analysis allows solving problems of optimal routing of telecommunications networks. Shortest path search algorithms include Dijkstra's algorithm for finding optimal routes, A\* algorithm for finding paths taking into account heuristics, and Floyd-Warshall algorithm for analyzing all pairs of nodes. Minimum backbone problems include constructing optimal tree topologies, minimizing the total cable length, and optimizing the location of switching nodes (Figure 1.3). Accessibility analysis allows identifying areas inaccessible to the network infrastructure, estimating recovery time after failures, and planning backup routes [4].

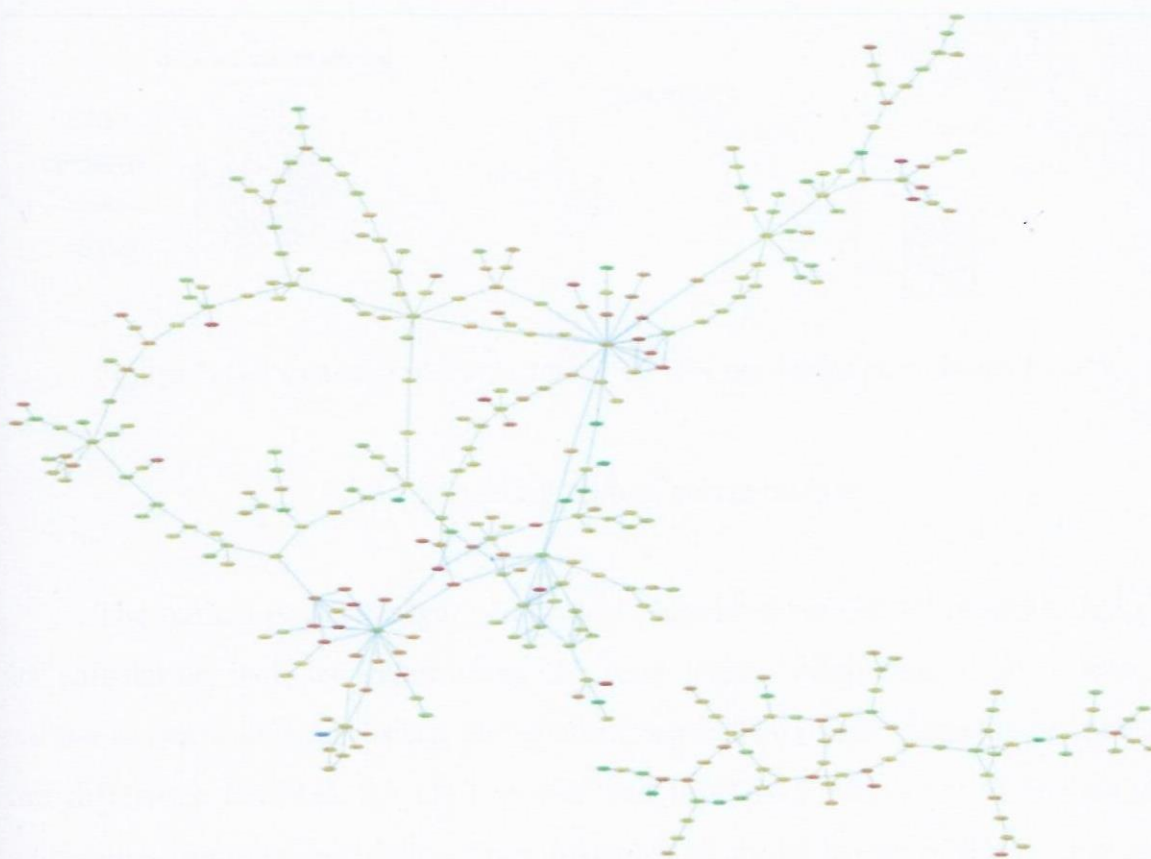


Figure 1.3 – Network analysis of telecommunications network topology

### 1.3 Spatial Analysis Methods for Telecommunications Networks

#### 1.3.1 GPON Network Components and Their Spatial Representation

GPON (Gigabit Passive Optical Network) consists of several key components, each of which has a specific spatial representation in GIS (Figure 1.4). OLT (Optical Line Terminal) is represented as a point with attributes of power, number of ports, model and status, located in central offices and communication nodes. Optical cable is represented as a line with attributes of fiber type, length, loss and year of installation, routed underground, overhead or in ducts. Splitters are represented as points with attributes of branching ratio, loss and number of ports, located in distribution cabinets and couplings. Subscriber connections are represented as points with attributes of service type, speed and contract, tied to houses, apartments and offices [5].

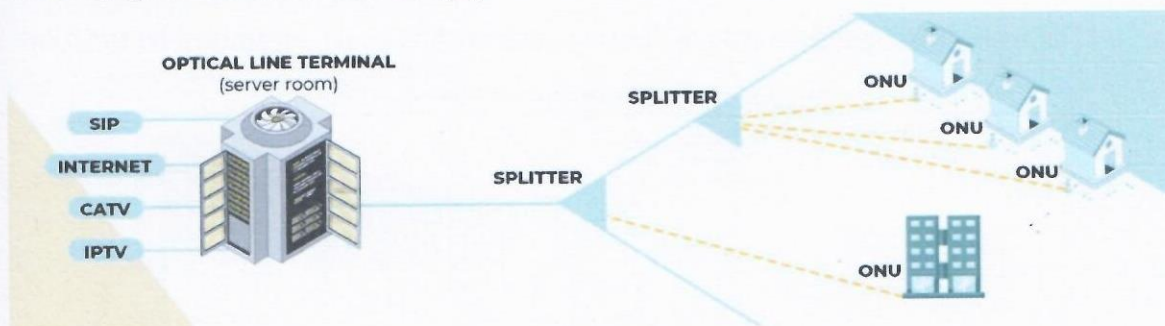


Figure 1.4 – Spatial representation of GPON network components in GIS

### 1.3.2 Modeling optical power budget

The optical power budget is a critical parameter for GPON network design. Its calculation includes determining the total losses, consisting of fiber losses, splitter losses, connector losses, and power margin. The power budget is defined as the difference between the OLT power and the ONU sensitivity. The detailed calculation involves calculating fiber losses based on the length of the section and the attenuation coefficient, splitter losses depending on the branching coefficient, connector losses taking into account their number, and adding a power margin to compensate for unexpected losses. Visualization in GIS involves color coding of sections by loss level, interactive reports on power budget violations, and generating recommendations for network optimization [6].

### 1.3.3 FPV systems and radio coverage modeling

FPV (First Person View) drones use radio channels to transmit video in real time. Their coverage modeling involves the use of various radio wave propagation models. The Free Space Path Loss model is used for idealized conditions, the Okumura-Hta model is used for urban conditions, the COST-231 model is used for suburban conditions, and the two-beam model takes into account signal reflection (Figure 1.5). Factors affecting coverage include signal frequency (1.2 GHz, 2.4 GHz, 5.8 GHz), transmitter power, receiver sensitivity, the presence of obstacles in the form of buildings, trees and terrain, as well as atmospheric conditions [7].

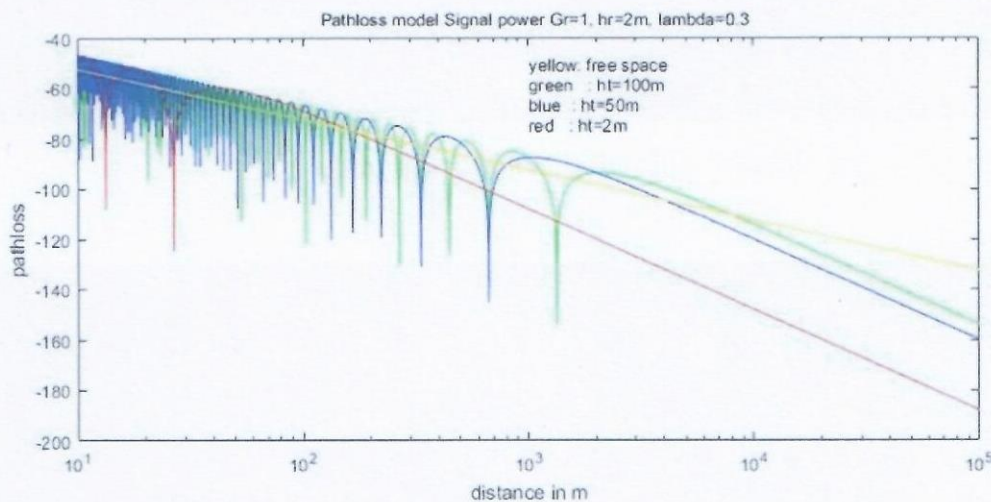


Figure 1.5 – Free Space Path Loss Model

### 1.3.4 Starlink Satellite Communication and Availability Analysis

The Starlink system uses low-orbit satellites to provide global coverage (Figure 1.6). The GIS analysis includes consideration of satellite system parameters, including an orbital altitude of 550 km, a number of satellites in the thousands, a minimum elevation angle of  $25^\circ$  for stable communication, and an azimuth as the direction to the satellite [8].

The introduction of 5G technology, especially in the high-frequency millimeter wave (mmWave, 24–100 GHz) range, fundamentally changes the approach to planning radio networks. Unlike previous generations (3G/4G), where the signal was sufficient to bypass obstacles, millimeter wave propagation is quasi-

### 1.3.5 Features of 5G network spatial planning

Figure 1.7 – Starlink satellite communication availability map

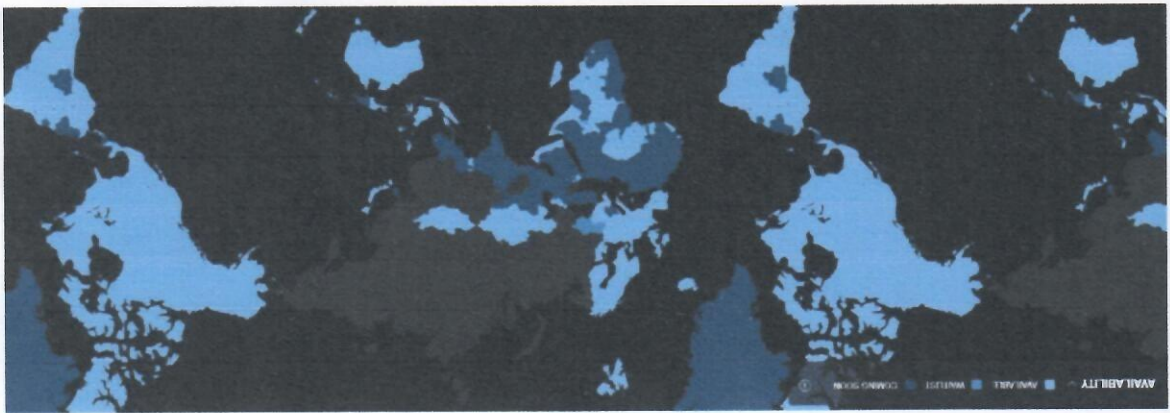
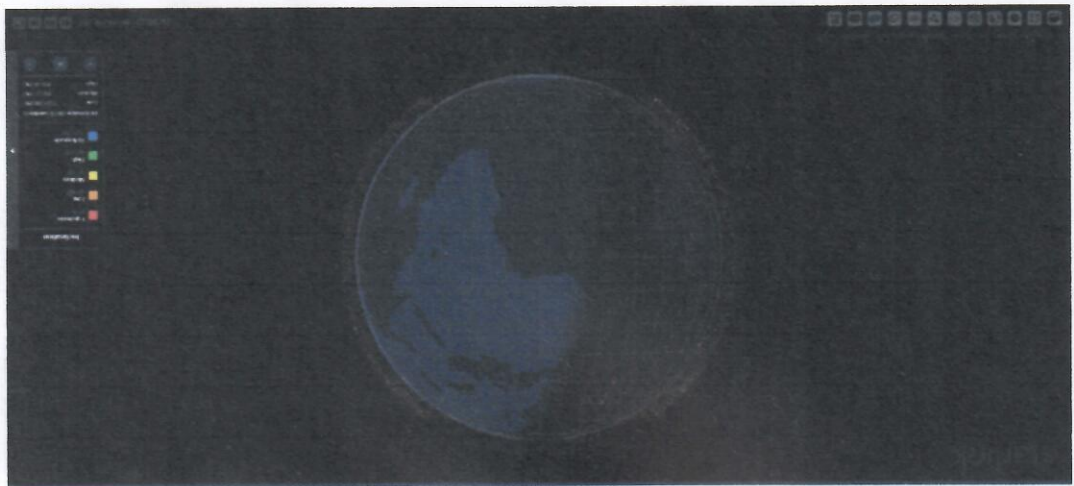


Figure 1.6 – Starlink Satellite Model



The calculation of the visibility area involves determining the positions of the satellites, calculating the elevation angle for each satellite, and identifying visible satellites that exceed the minimum elevation angle. Starlink availability factors include geographic coordinates, the presence of obstacles on the horizon, the density of the satellite constellation, and regulatory restrictions (Figure 1.7) [9].

optical and critically depends on the availability of line of sight (LoS). Any physical obstacle - a building, a tree or a billboard - can lead to a connection failure (Figure 1.8).

In a GIS environment, 5G modeling requires the use of highly accurate digital surface models (DSM) that take into account the geometry of urban development. The main analysis method is the calculation of line of sight zones (Viewshed Analysis) for the placement of Small Cell base stations. Since the range of such stations is only 100–300 meters, GIS tools are used to find optimal locations for equipment installation (lampposts, building facades) in order to minimize “dead zones” and the “urban canyon” effect [10].

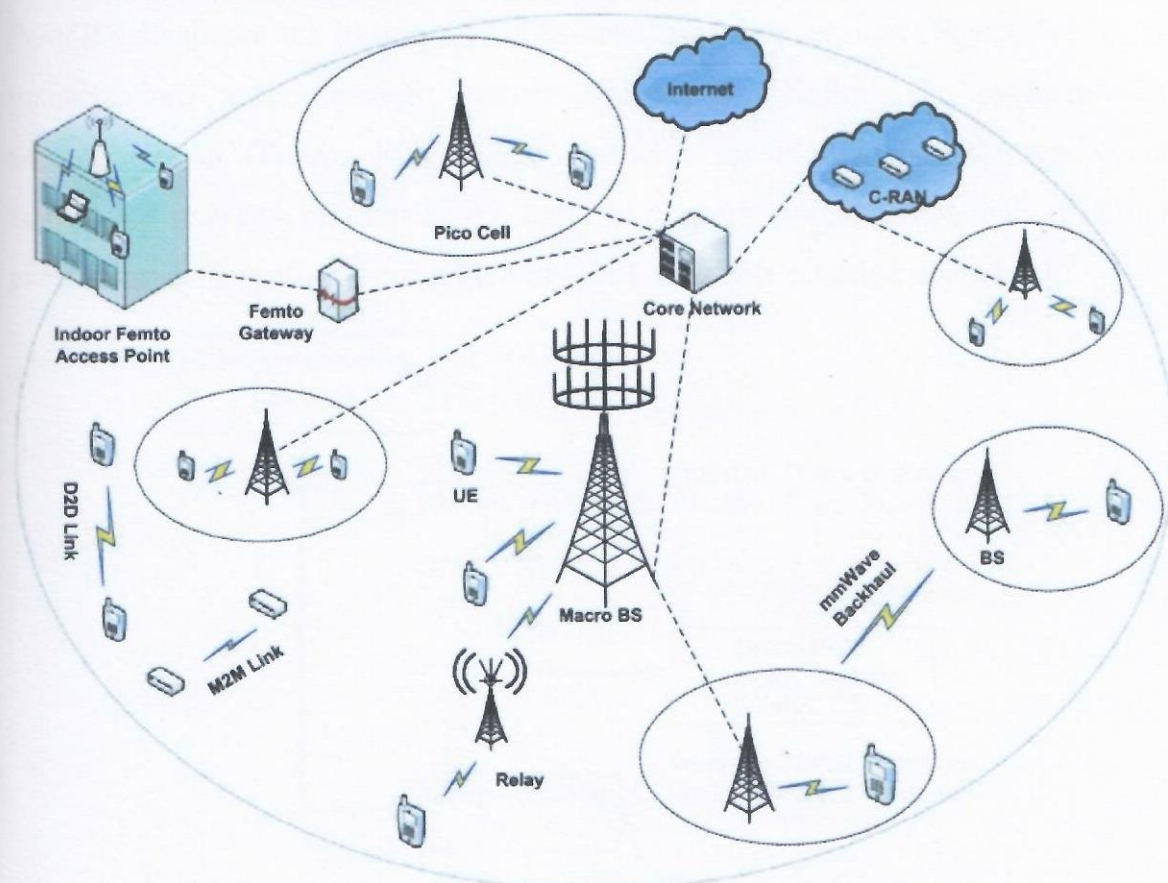


Figure 1.8 – Architecture of 5G network

#### 1.4 GIS Integration with Network Management Systems

### 1.4.1 Architecture of GIS Integration with OSS/BSS

Modern approaches to GIS integration with network management systems involve implementation at three levels. At the data level, database synchronization is carried out via API, which ensures the relevance of information in all systems. At the service level, web services are used for data exchange, which allows integrating various systems regardless of their technological platform. At the application level, a single user interface is implemented, which provides convenient work with all functions.

Integration technologies include REST API for real-time data exchange, PostGIS databases for storing spatial information, web services (WMS, WFS) for visualization, and message brokers (RabbitMQ, Kafka) for asynchronous communication (Figure 1.9). Usage scenarios include automated creation of connection requests, monitoring the status of network elements on a map, planning preventive work, and analyzing the impact of accidents on subscribers [11].

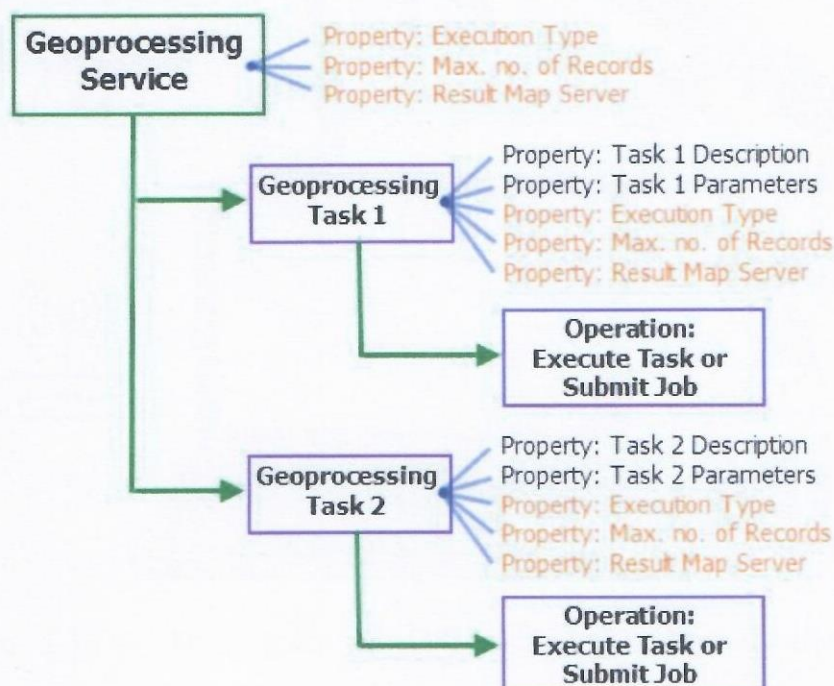


Figure 1.9 – Architecture of GIS integration with network management systems

### 1.4.2 Monitoring systems and their integration with GIS

For effective management of telecommunication networks, GIS is integrated with monitoring systems. Integration with Zabbix and Nagios allows you to display alarms on a map, correlate events with geographical location, and predict the impact of accidents on the network infrastructure. Integration with Grafana provides the creation of monitoring panels with geographical context, trend analysis taking into account spatial distribution, and generation of reports on the quality of services by region.

Integration with accounting systems includes connection to Userside for subscriber management, to 1C for accounting of equipment and materials, and to CRM for customer service. Such integration provides a comprehensive approach to telecommunications infrastructure management and increases the operator's efficiency (Figure 1.10) [12].



Figure 1.10 – Integration of GIS with telecommunications network monitoring systems

## Додаток Б

## Плакати

Міністерство освіти та науки України  
Національний університет «Полтавська політехніка імені Юрія Кондратюка»  
Кафедра автоматики, електроніки та телекомунікацій

**Інтеграція геоінформаційних систем та методів  
просторового аналізу для планування та управління  
телекомунікаційними мережами**

Кваліфікаційна робота магістра

Виконав:  
Студент групи 601ТТ

Сухоребрій О.В.

Керівник:  
Професор: д.т.н.

Косенко В.В.

1

**Актуальність теми** зумовлена цифровізацією суспільства, де стабільний зв'язок є основою економіки, медицини та освіти. Геоінформаційні системи допомагають ефективніше планувати мережі, оптимізувати витрати та швидко реагувати на проблеми.

**Мета дослідження** полягає у визначенні, як використання геоінформаційних систем допомагає краще планувати та керувати телекомунікаційними мережами. Робота спрямована на пошук простих і практичних рішень для зниження витрат та підвищення якості зв'язку.

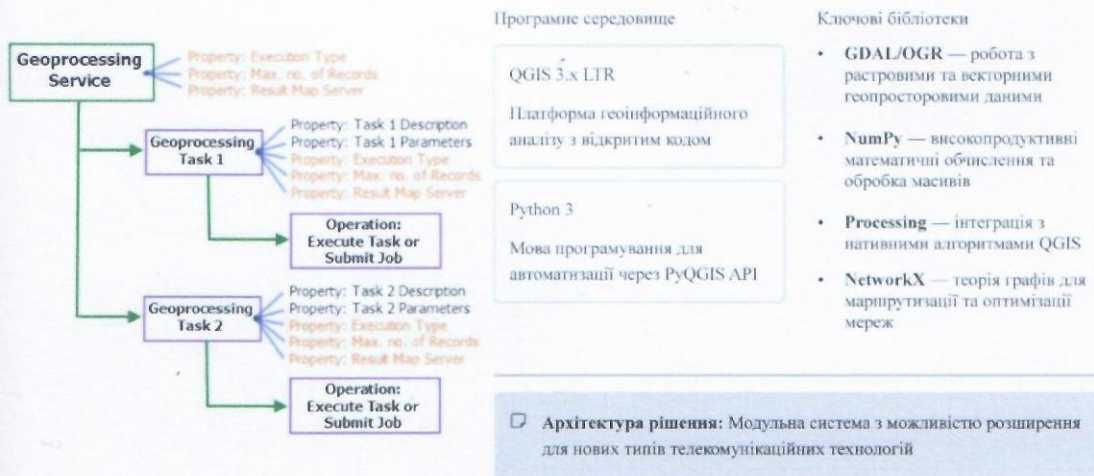
**Об'єктом дослідження** виступають процеси організації та розвитку телекомунікаційних систем, у яких інтеграція GIS забезпечує підвищення ефективності планування та керування мережами.

**Предметом дослідження** є методи геоінформаційного аналізу, алгоритми автоматизованого трасування маршрутів та моделювання радіопокриття на основі просторових даних

Для виконання поставленої мети в роботі необхідно виконати наступні **завдання**:

- Дослідити сучасні можливості геоінформаційних систем та їх застосування у сфері телекомунікацій.
- Розробити методику моделювання телекомунікаційних мереж із використанням GIS.
- Продемонструвати застосування цієї методики на прикладах різних технологій (GPON, FPV, 5G, Starlink)
- Оцінити ефективність моделювання мереж за допомогою GIS та підготувати висновки з графічними матеріалами

Інструментарій розробки



Методологія: Геометричний підхід

Ключова інновація

Замість складних та ресурсомістких радіофізичних розрахунків (Ray Tracing) запропоновано **геометричний підхід**, який забезпечує баланс між точністю та швидкістю обчислень.



- GPON**  
Контроль довжини ліній та кластеризація абонентів
- FPV / 5G**  
Аналіз прямої видимості (Line of Sight)
- Starlink**  
Аналіз затінення горизонту

10x	02	95%
Швидкість обчислень Порівняно з класичними методами	Вартість даних Використання відкритих джерел	Точність Достатня для інженерного планування

## GPON: Автоматизація топології мережі

- 1  
Імпорт геоданих  
Завантаження будівель з OpenStreetMap та їх класифікація
- 2  
Генерація топології  
Автоматичне створення вузлів (OLT, ) та трасування кабельних ліній
- 3  
Збереження результатів  
Формування структурованої бази даних у форматі GeoPackage






## Результат

Створено автоматизовану систему генерації GPON-мережі, яка усуває людський фактор при проєктуванні та забезпечує коректне трасування абонентських кабельних ліній.

Ручне трасування замінено на алгоритмічний підхід, що скорочує час проєктування з тижнів до годин.

## GPON: Оптимізація через кластеризацію

 <p><b>K-Means Clustering</b> Алгоритм з обмеженням радіуса для групування абонентів</p>	 <p><b>Автонідбір параметрів</b> Оптимальна кількість спліттерів на основі щільності забудови</p>	 <p><b>Контроль довжини</b> Моніторинг абонентських відводів (&lt;500 м)</p>
--	---	---

## Результати аналізу та оптимізації



Виявлено **20 проблемних підключень** з довжиною абонентського відводу понад 500 метрів, що потенційно призводить до погіршення якості сигналу.

Після оптимізації розташування спліттерів досягнуто зменшення витрат кабелю до **15%** та покращення показників мережі.

FPV-дрони: Виклик 3D-навігації

6

Проблема

У складних рельєфах виникають «мертві зони» — радіотіні, де повністю втрачається відеозв'язок з дроном. Планування маршрутів по традиційній 2D-карті не працює.



Вхідні дані  
SRTM DEM — цифрова модель рельєфу

Метод аналізу  
Алгоритм Viewshed через GDAL

Визначення зон покриття  
Растровий аналіз видимості

Висота 80 м  
Зона стабільного покриття визначена успішно,  
пряма видимість забезпечена

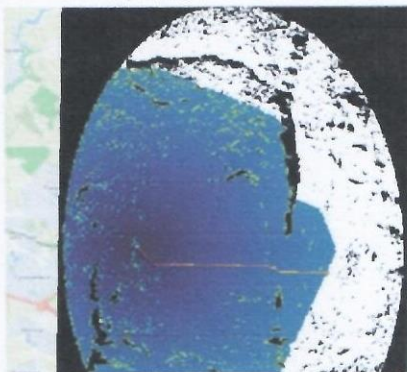
FPV: Побудова безпечного маршруту

7

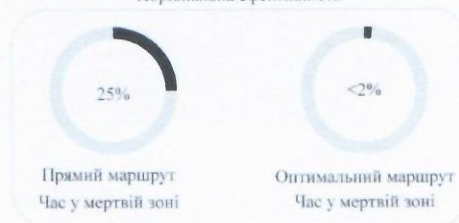
1	2	3
Визначення вартості Зона видимості = низька вартість (безпечно). Тіньова зона = висока вартість (небезпечно)	Алгоритм Дейкстри Пошук шляху найменшої вартості (Least Cost Path) з автоматичним обходом мертвих зон	Валідація маршруту Перевірка часу перебування в тіні та загальної безпеки польоту

Застосування алгоритму Least Cost Path забезпечило зменшення часу перебування в радіотіні у 12 разів, що критично важливо для стабільності керування FPV-дроном.

Оптимізований маршрут автоматично огинає небезпечні зони, зберігаючи стабільний відеозв'язок.



Порівняльна ефективність

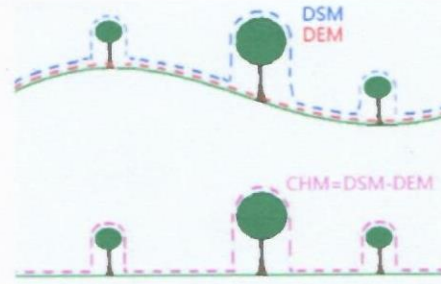


Starlink: Подолання "міських каньйонів"

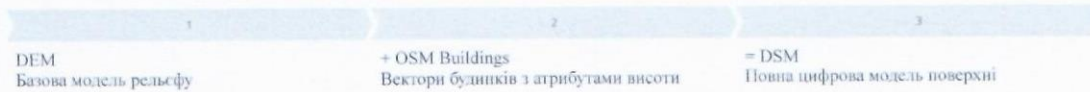
8

Для коректної роботи супутникового зв'язку Starlink необхідний чистий горизонт з кутом місця понад 25°. У щільній міській забудові високі будинки створюють значні перешкоди та блокують сигнал.

Стандартні SRTM DEM містять інформацію лише про рельєф місцевості, повністю ігноруючи штучні об'єкти — будівлі, споруди, інфраструктуру.



Інноваційне рішення: Синтетична DSM



Розроблено методіку програмного синтезу Digital Surface Model шляхом інтеграції рельєфу з векторними даними OpenStreetMap, що усуває необхідність дорогих LiDAR-зйомок та забезпечує точне моделювання міського середовища.

Starlink: Моделювання перешкод сигналу

9

Методика Hillshade Analysis

Параметри моделювання

Hillshade Analysis дозволяє визначити зони, де сигнал від супутника блокується міською забудовою. Метод базується на аналізі тіней від будівель.



92%

25°

180°

Точність моделі

Мінімальний кут

Азимут

Валідація відносно теоретичних розрахунків

Критичний параметр для Starlink

Південна орієнтація супутників

## Мережі 5G mmWave: Моделювання покриття

## Підготовка DSM

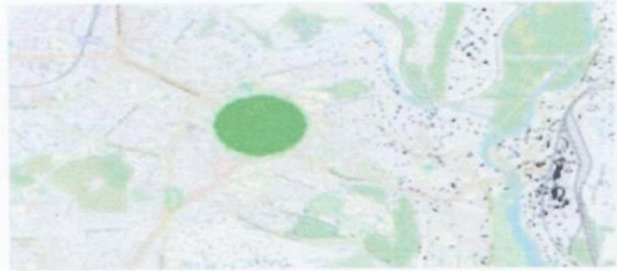
Завантаження цифрової моделі поверхні з урахуванням дерев та будівель для точного розрахунку перешкод

## Розрахунок Viewshed

Аналіз прямої видимості (LoS) для однієї базової станції з робочим радіусом 300-500 метрів

## Візуалізація результатів

Картографування зон з сигналом (зелені області) та мертвих зон (прозорі області без покриття)



**Специфіка mmWave:** Високочастотні сигнали FR2 діапазону критично залежать від прямої видимості та легко блокуються перешкодами

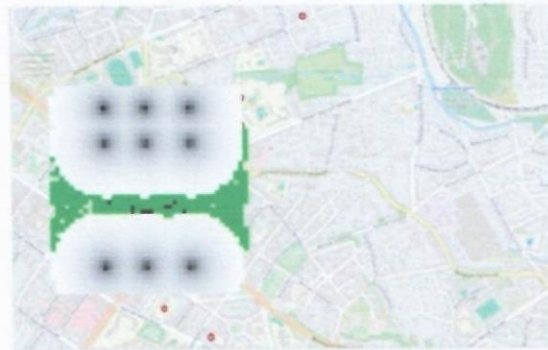
## Сценарний аналіз мережі 5G

## Методологія дослідження

Для оцінки реального покриття міської території змодельовано регулярну сітку з 9 базових станцій у конфігурації 3x3. Результати окремих розрахунків Viewshed об'єднано у єдиний інтегральний шар за допомогою технології VRT Merge.

## Ключові висновки

- Навіть при щільному розташуванні БС залишаються мертві зони
- Найбільші проблеми — внутрішні двори та простори між будівлями
- Модель точно ідентифікує локації для Small Cells



Інтегральна карта покриття демонструє необхідність додаткових вузлів у проблемних зонах

## Практичне значення та досягнення



## GPON

Автоматизація проектування оптичних мереж, скорочення часу рутинних операцій



## FPV

Зниження ризику втрати дрона у 12 разів завдяки LCP-маршрутизації



## Starlink

Оцінка доступності без виїзду на місце через синтетичну DSM



## 5G mmWave

Інструмент планування оптимального розташування базових станцій

## Універсальність рішення

Методика повністю реалізована на базі Open Source програмного забезпечення (QGIS/Python), що забезпечує доступність для будь-якого телекомунікаційного провайдера без додаткових витрат на ліцензії.

## Дякую!

Готовий відповісти на ваші запитання

Додаток В  
Програмні Скрипти

Скрипт GPON - 1 частина

```
# -*- coding: utf-8 -*-
from qgis.core import (
    QgsProject, QgsVectorLayer, QgsField, QgsFields, QgsVectorFileWriter,
    QgsWkbTypes, QgsCategorizedSymbolRenderer, QgsRendererCategory,
    QgsMarkerSymbol, QgsLineSymbol
)
from PyQt5.QtCore import QVariant

#
=====

SAVE_PATH = " C:/example/path/"
BUILDINGS_FILE_PATH = " C:/example/path/file.shp"
#
=====

if not SAVE_PATH.endswith('/'):
    SAVE_PATH += '/'
crs = QgsProject.instance().crs()
if not crs.isValid():
    crs = QgsCoordinateReferenceSystem("EPSG:4326")
print("Створюю шари проекту (v3.40)...")

# === 1. ШАР: ОПТИЧНІ КАБЕЛІ ===
cable_layer_path = f"{SAVE_PATH}1_Optical_Cables.gpkg"
```

```

cable_fields = QgsFields()
cable_fields.append(QgsField("fiber_count", QVariant.Int, "К-ть волокон"))
cable_fields.append(QgsField("install_method", QVariant.String, "Метод
прокладки"))
cable_fields.append(QgsField("length_m", QVariant.Double, "Довжина
(м)"))
writer = QgsVectorFileWriter(cable_layer_path, "UTF-8", cable_fields,
QgsWkbTypes.LineString, crs, "GPKG")
del writer
cables_layer = QgsVectorLayer(cable_layer_path, "1. Оптичні кабелі",
"ogr")
cable_styles = [
    ('4', '#DAA520', 1.0, '4F'), ('8', '#FFA500', 1.5, '8F'),
    ('12', '#32CD32', 2.0, '12F'), ('16', '#1E90FF', 2.5, '16F'),
    ('24', '#FF4500', 3.0, '24F'), ('48', '#8B0000', 3.5, '48F+')
]
categories = []
for value, color, width, label in cable_styles:
    symbol = QgsLineStyle.createSimple({'color': color, 'width':
str(width)})
    categories.append(QgsRendererCategory(value, symbol, label))
default_symbol = QgsLineStyle.createSimple({'color': '#FF00FF', 'width':
'1.0', 'style': 'dot'})
categories.append(QgsRendererCategory("", default_symbol, "Інше"))
renderer = QgsCategorizedSymbolRenderer('fiber_count', categories)
cables_layer.setRenderer(renderer)
cables_layer.startEditing()
cables_layer.addExpressionField('$length', QgsField('length_m',
QVariant.Double))
cables_layer.commitChanges()

```

```

QgsProject.instance().addMapLayer(cables_layer)
print("...Шар '1. Оптичні кабелі' створено.")

# === 2. ШАР: ВУЗЛИ МЕРЕЖІ ===
node_layer_path = f"{SAVE_PATH}2_Network_Nodes.gpkg"
node_fields = QgsFields()
node_fields.append(QgsField("node_id", QVariant.String, "ID Вузла"))
node_fields.append(QgsField("node_type", QVariant.String, "Тип вузла"))
node_fields.append(QgsField("splitter_ratio", QVariant.String,
"Подільник"))
node_fields.append(QgsField("capacity", QVariant.Int, "Ємність (порти)"))
writer = QgsVectorFileWriter(node_layer_path, "UTF-8", node_fields,
QgsWkbTypes.Point, crs, "GPKG")
del writer
nodes_layer = QgsVectorLayer(node_layer_path, "2. Вузли мережі", "ogr")
node_styles = [
    ('CO', 'star', '#FF0000', 8, 'Головний Офіс (OLT)'),
    ('Муфта', 'square', '#0000FF', 5, 'Муфта (зварка)'),
    ('Коробка SFU', 'circle', '#32CD32', 5, 'Коробка (Приватний сектор)'),
    ('Коробка MDU', 'diamond', '#FFA500', 6, 'Коробка (Багатоповерхівка)')
]
categories = []
for value, shape, color, size, label in node_styles:
    symbol = QgsMarkerSymbol.createSimple({'name': shape, 'color': color,
'size': str(size)})
    categories.append(QgsRendererCategory(value, symbol, label))
renderer = QgsCategorizedSymbolRenderer('node_type', categories)
nodes_layer.setRenderer(renderer)
QgsProject.instance().addMapLayer(nodes_layer)
print("...Шар '2. Вузли мережі' створено.")

```

```

# === 3. ШАР: АБОНЕНТСЬКІ ВІДВОДИ ===
drop_layer_path = f"{SAVE_PATH}3_Drop_Cables.gpkg"
drop_fields = QgsFields()
drop_fields.append(QgsField("drop_type",QVariant.String,"Тип
патчкорду"))
drop_fields.append(QgsField("length_m", QVariant.Double, "Довжина
(м)"))
writer = QgsVectorFileWriter(drop_layer_path, "UTF-8", drop_fields,
QgsWkbTypes.LineString, crs, "GPKG")
del writer
drop_layer = QgsVectorLayer(drop_layer_path, "3. Абонентські відводи
(Патчкорди)", "ogr")
drop_layer.startEditing()
drop_layer.addExpressionField('$length', QgsField('length_m',
QVariant.Double))
drop_layer.commitChanges()
QgsProject.instance().addMapLayer(drop_layer)
print("...Шар '3. Абонентські відводи (Патчкорди)' створено.")

# === 4. ШАР: БУДИНКИ ===
buildings_layer = QgsVectorLayer(BUILDINGS_FILE_PATH, "4.
Будинки (Абоненти)", "ogr")
if buildings_layer.isValid():
    provider = buildings_layer.dataProvider()
    existing_fields = provider.fields().names()
    buildings_layer.startEditing()
    if 'build_type' not in existing_fields:
        provider.addAttributes([[QgsField('build_type', QVariant.String, "Тип
будинку")]])

```

```

if 'patch_type' not in existing_fields:
    provider.addAttributes([QgsField('patch_type', QVariant.String, "Тип
патчкорду")])
    buildings_layer.commitChanges()
    QgsProject.instance().addMapLayer(buildings_layer)
    print("...Шар '4. Будинки' успішно завантажено.")
else:
    print(f"ПОМИЛКА: Не можу знайти ваш файл будинків тут:
{BUILDINGS_FILE_PATH}")

iface.messageBar().pushMessage("Успіх!", "Структуру проєкту створено
(v3.40).", level=Qgis.Success)

```

#### Скрипт GPON - 2 частина

```

# -*- coding: utf-8 -*-
from qgis.core import (
    QgsProject, QgsVectorLayer, QgsField, QgsFeature, QgsGeometry,
    QgsPointXY,
    QgsCoordinateReferenceSystem, QgsCoordinateTransform,
    QgsFeatureRequest,
    QgsRuleBasedRenderer, QgsSymbol, QgsLineSymbol
)
from PyQt5.QtCore import QVariant
from PyQt5.QtGui import QColor
import processing
import random
import math

```

#

---

---

SAVE\_PATH = " C:/example/path/"

BUILDINGS\_FILE\_PATH = " C:/example/path/file.shp"

TARGET\_SFU\_SUBSCRIBERS = 500

NUM\_SFU\_CLUSTERS = 12 # Збільшено кількість кластерів для

менших радіусів

POLTAVA\_CENTER\_LON = 34.5512

POLTAVA\_CENTER\_LAT = 49.5891

ANALYSIS\_RADIUS\_METERS = 1500

DROP\_CABLE\_LIMIT\_METERS = 500.0

MAX\_CLUSTER\_RADIUS = 500.0 # Максимальний радіус кластера

#

---

---

print("--- Початок Аналізу GPON ---")

# --- 1. ПЕРЕВІРКА СИСТЕМИ КООРДИНАТ ---

buildings\_layer = QgsVectorLayer(BUILDINGS\_FILE\_PATH, "Будинки  
(Оригінал)", "ogr")

if not buildings\_layer.isValid():

print(f"ПОМИЛКА: Не можу знайти {BUILDINGS\_FILE\_PATH}")

raise Exception("Шар будинків не знайдено")

crs\_latlon = buildings\_layer.crs()

crs\_utm = QgsCoordinateReferenceSystem("EPSG:32636")

transform\_to\_utm = QgsCoordinateTransform(crs\_latlon, crs\_utm,  
QgsProject.instance())

```

transform_to_latlon = QgsCoordinateTransform(crs_utm, crs_latlon,
QgsProject.instance())
print(f"...Вихідна СК: {crs_latlon.authid()}. Цільова СК для аналізу:
{crs_utm.authid()}")

```

```
# --- 2. ПОШУК РОБОЧИХ ШАРІВ ---
```

```
print("...Крок 1: Пошук робочих шарів у проєкті...")
```

```
project = QgsProject.instance()
```

```
cable_layer = project.mapLayersByName("1. Оптичні кабелі")
```

```
node_layer = project.mapLayersByName("2. Вузли мережі")
```

```
drop_layer = project.mapLayersByName("3. Абонентські відводи
(Патчкорди)")
```

```
if not all([cable_layer, node_layer, drop_layer]):
```

```
    print("ПОМИЛКА: Не можу знайти шари ('1...', '2...', '3...'). Запустіть
Скрипт 1 (Виправлений) ПЕРШ.")
```

```
    raise Exception("Робочі шари не знайдено.")
```

```
cable_layer = cable_layer[0]
```

```
node_layer = node_layer[0]
```

```
drop_layer = drop_layer[0]
```

```
cable_provider = cable_layer.dataProvider()
```

```
node_provider = node_layer.dataProvider()
```

```
drop_provider = drop_layer.dataProvider()
```

```
print("...Робочі шари успішно знайдено.")
```

```
# --- ДОДАЄМО ПОТРІБНІ ПОЛЯ ДО ШАРІВ ---
```

```
print("...Перевірка та додавання полів...")
```

```
# Додаємо поля до шару вузлів, якщо їх немає
```

```
existing_node_fields = [field.name() for field in node_provider.fields()]
```

```
if 'max_distance' not in existing_node_fields:
```

```

node_provider.addAttribute([QgsField("max_distance",
QVariant.String)])
    if 'capacity' not in existing_node_fields:
        node_provider.addAttribute([QgsField("capacity", QVariant.Int)])
    node_layer.updateFields()
    existing_drop_fields = [field.name() for field in drop_provider.fields()]
    if 'distance' not in existing_drop_fields:
        drop_provider.addAttribute([QgsField("distance", QVariant.Double)])
    if 'problem' not in existing_drop_fields:
        drop_provider.addAttribute([QgsField("problem", QVariant.String)])
    drop_layer.updateFields()

# --- ФУНКЦІЯ ДЛЯ ОПТИМАЛЬНОГО РОЗПОДІЛУ КЛАСТЕРІВ ---
def optimize_clusters_with_radius_limit(points_utm, max_clusters=15,
max_radius=200.0):
    """Оптимізація кластеризації з обмеженням радіусу"""
    best_clusters = max_clusters
    best_result = None

    for cluster_count in range(8, max_clusters + 1): # Перебираємо різну
    кількість кластерів
        kmeans_params = {
            'INPUT': points_utm,
            'CLUSTERS': cluster_count,
            'OUTPUT': 'memory:temp_clusters'
        }
        try:
            kmeans_result = processing.run("native:kmeansclustering",
kmeans_params)
            cluster_layer = kmeans_result['OUTPUT']

```

```

max_cluster_radius = 0
valid = True

for feat in cluster_layer.getFeatures():
    cluster_id = feat['CLUSTER_ID']
    cluster_points = []
    for f in
cluster_layer.getFeatures(QgsFeatureRequest().setFilterExpression(f"CLUSTER_I
D" = {cluster_id}')):
        cluster_points.append(f.geometry().asPoint())

if len(cluster_points) > 1:
    # Обчислюємо центр та максимальну відстань
    center_x = sum(p.x() for p in cluster_points) / len(cluster_points)
    center_y = sum(p.y() for p in cluster_points) / len(cluster_points)
    center = QgsPointXY(center_x, center_y)

    cluster_radius = max(center.distance(p) for p in cluster_points)
    max_cluster_radius = max(max_cluster_radius, cluster_radius)

    if cluster_radius > max_radius:
        valid = False
        break

if valid and max_cluster_radius <= max_radius:
    best_clusters = cluster_count
    best_result = cluster_layer
    print(f"...Знайдено оптимальну конфігурацію: {cluster_count}
кластерів, макс. радіус: {max_cluster_radius:.1f}м")
    break

```

```
except Exception as e:
    continue

if best_result is None:
    # Якщо не знайшли ідеальну конфігурацію, беремо останню
    kmeans_params = {
        'INPUT': points_utm,
        'CLUSTERS': max_clusters,
        'OUTPUT': 'memory:temp_clusters'
    }
    best_result = processing.run("native:kmeansclustering",
kmeans_params)['OUTPUT']
    print(f"...Використовую максимальну кількість кластерів:
    {max_clusters}")

    return best_result

# --- ПЕРЕВІРКА, ЧИ ВЖЕ Є ДАНІ ---
if node_layer.featureCount() > 0:
    print("...Знайдено існуючі дані у шарах. Переходжу одразу до
    стилізації...")
else:
    print("...Шари порожні. Починаю генерацію даних...")

# --- ФІЛЬТР ТА РОЗДІЛЕННЯ АБОНЕНТІВ ---
print(f"...Крок 2: Фільтрація та розділення MDU/SFU в радіусі
    {ANALYSIS_RADIUS_METERS}м...")
```

```

center_point_latlon =
QgsGeometry.fromPointXY(QgsPointXY(POLTAVA_CENTER_LON,
POLTAVA_CENTER_LAT))
center_point_utm = QgsGeometry(center_point_latlon)
center_point_utm.transform(transform_to_utm)
analysis_buffer_geom =
center_point_utm.buffer(ANALYSIS_RADIUS_METERS, 5)

```

```

mdu_features = []
sfu_features = []
all_filtered_features = []
field_index = buildings_layer.fields().indexOfName('building')

if field_index == -1:
    print("ПОПЕРЕДЖЕННЯ: У файлі 1.shp немає колонки 'building'.
Всі будинки будуть вважатися SFU.")

```

```

for feat in buildings_layer.getFeatures():
    building_geom_latlon = feat.geometry()
    building_geom_utm = QgsGeometry(building_geom_latlon)
    building_geom_utm.transform(transform_to_utm)
    if building_geom_utm.centroid().intersects(analysis_buffer_geom):
        all_filtered_features.append(feat)
        building_type = None
        if field_index != -1:
            try:
                building_type = feat['building']
            except:
                building_type = None
        if building_type == 'apartments':

```

```
        mdu_features.append(feat)
        elif building_type in ['house', 'detached', 'yes', 'residential'] or
building_type is None:
            sfu_features.append(feat)

print(f"...Знайдено {len(mdu_features)} MDU (багатоповерхівок).")
print(f"...Знайдено {len(sfu_features)} SFU (приватний сектор).")

if len(sfu_features) > TARGET_SFU_SUBSCRIBERS:
    sfu_features = random.sample(sfu_features,
TARGET_SFU_SUBSCRIBERS)
    print(f"...Вибрано {len(sfu_features)} SFU для кластеризації.")

# --- СТВОРЕННЯ ГОЛОВНОГО ОФІСУ (СО) ---
if not all_filtered_features:
    print("ПОМИЛКА: Не знайдено ЖОДНОГО будинку в зоні
аналізу.")
    raise Exception("Будинки не знайдено")

sum_x_utm, sum_y_utm = 0.0, 0.0
for feat in all_filtered_features:
    centroid_utm = QgsGeometry(feat.geometry())
    centroid_utm.transform(transform_to_utm)
    point = centroid_utm.centroid().asPoint()
    sum_x_utm += point.x()
    sum_y_utm += point.y()

co_x_utm = sum_x_utm / len(all_filtered_features)
co_y_utm = sum_y_utm / len(all_filtered_features)
```

```

co_point_utm = QgsGeometry.fromPointXY(QgsPointXY(co_x_utm,
co_y_utm))
co_point_latlon = QgsGeometry(co_point_utm)
co_point_latlon.transform(transform_to_latlon)

olt_feat = QgsFeature(node_layer.fields()) # Використовуємо оновлені
ПОЛЯ
olt_feat.setGeometry(co_point_latlon)
olt_feat.setAttribute('node_id', 'CO-01')
olt_feat.setAttribute('node_type', 'CO')
node_provider.addFeatures([olt_feat])
co_point_for_cables = olt_feat.geometry().asPoint()
print("...ГОЛОВНИЙ Офіс (CO) створено в центрі активності.")

# --- ОБРОБКА MDU (ОПТИМІЗОВАНА) ---
print(f"...Крок 3: Обробка {len(mdu_features)} MDU...")
mdu_boxes_to_add = []
mdu_cables_to_add = []

for i, mdu_feat in enumerate(mdu_features):
    mdu_box_geom = mdu_feat.geometry().centroid()
    mdu_box_point = mdu_box_geom.asPoint()

    box_feat = QgsFeature(node_layer.fields())
    box_feat.setGeometry(mdu_box_geom)
    box_feat.setAttribute('node_id', f'BOX-MDU-{i+1:03d}')
    box_feat.setAttribute('node_type', 'Коробка MDU')
    box_feat.setAttribute('splitter_ratio', '1:8') # Зменшено для економії
ВОЛОКОН
    mdu_boxes_to_add.append(box_feat)

```

```

        line_geom = QgsGeometry.fromPolylineXY([co_point_for_cables,
mdu_box_point])
        cable_feat = QgsFeature(cable_provider.fields())
        cable_feat.setGeometry(line_geom)
        cable_feat.setAttribute('fiber_count', 4) # Зменшено з 8 до 4 волокон
        cable_feat.setAttribute('install_method', 'Підвіс до MDU')
        mdu_cables_to_add.append(cable_feat)

node_provider.addFeatures(mdu_boxes_to_add)
cable_provider.addFeatures(mdu_cables_to_add)
print("...MDU коробки та 4F кабелі до них створено.")

# --- ОБРОБКА SFU (ОПТИМІЗОВАНА КЛАСТЕРИЗАЦІЯ) ---
print(f"...Крок 4: Оптимізована кластеризація для {len(sfu_features)}
SFU...")
if len(sfu_features) > 0:
    temp_points_layer = QgsVectorLayer(f"Point?crs={crs_utm.authid()}",
"sfu_points_utm", "memory")
    temp_provider = temp_points_layer.dataProvider()
    temp_provider.addAttributes([QgsField("orig_fid", QVariant.Int)])
    temp_points_layer.updateFields()
    temp_features_list = []
    original_sfu_features_map = {}

    for feat in sfu_features:
        orig_fid = feat.id()
        centroid_latlon = feat.geometry().centroid()
        centroid_utm = QgsGeometry(centroid_latlon)
        centroid_utm.transform(transform_to_utm)

```

```
temp_feat = QgsFeature()
temp_feat.setGeometry(centroid_utm)
temp_feat.setAttributes([orig_fid])
temp_features_list.append(temp_feat)
original_sfu_features_map[orig_fid] = feat

temp_provider.addFeatures(temp_features_list)
kmeans_output_layer = optimize_clusters_with_radius_limit(
    temp_points_layer,
    max_clusters=15,
    max_radius=MAX_CLUSTER_RADIUS
)

print("...Оптимізовану кластеризацію SFU завершено.")

cluster_data = {}
sfu_boxes_by_cluster = {}
sfu_boxes_to_add = []
sfu_cables_to_add = []
for feat in kmeans_output_layer.getFeatures():
    cluster_id = feat['CLUSTER_ID']
    point = feat.geometry().asPoint()
    if cluster_id not in cluster_data:
        cluster_data[cluster_id] = {
            'sum_x': 0.0,
            'sum_y': 0.0,
            'count': 0,
            'features': [],
            'points': []
        }
```

```

cluster_data[cluster_id]['sum_x'] += point.x()
cluster_data[cluster_id]['sum_y'] += point.y()
cluster_data[cluster_id]['count'] += 1
cluster_data[cluster_id]['features'].append(feats)
cluster_data[cluster_id]['points'].append(point)
for cluster_id, data in cluster_data.items():
    count = data['count']
    if count == 0:
        continue
    center_x_utm = data['sum_x'] / count
    center_y_utm = data['sum_y'] / count

    # Перевіряємо максимальну відстань у кластері
    cluster_center = QgsPointXY(center_x_utm, center_y_utm)
    max_distance = max(cluster_center.distance(p) for p in data['points'])

    box_point_utm =
    QgsGeometry.fromPointXY(QgsPointXY(center_x_utm, center_y_utm))
    box_point_latlon = QgsGeometry(box_point_utm)
    box_point_latlon.transform(transform_to_latlon)
    box_point_for_cables = box_point_latlon.asPoint()

    box_feat = QgsFeature(node_layer.fields())
    box_feat.setGeometry(box_point_latlon)
    box_feat.setAttribute('node_id', f'BOX-SFU-{cluster_id}')
    box_feat.setAttribute('node_type', 'Коробка SFU')

    # Оптимізований спліттер залежно від кількості абонентів
    if count <= 8:
        splitter_ratio = '1:8'

```

```

        fiber_count = 4
    elif count <= 16:
        splitter_ratio = '1:16'
        fiber_count = 4
    elif count <= 32:
        splitter_ratio = '1:32'
        fiber_count = 8
    else:
        splitter_ratio = '1:64'
        fiber_count = 8

    box_feat.setAttribute('splitter_ratio', splitter_ratio)
    box_feat.setAttribute('capacity', count)
    box_feat.setAttribute('max_distance', f'{max_distance:.1f}м')
    sfu_boxes_by_cluster[cluster_id] = box_feat
    sfu_boxes_to_add.append(box_feat)
    cable_feat = QgsFeature(cable_provider.fields())
    cable_feat.setGeometry(QgsGeometry.fromPolylineXY([co_point_f
or_cables, box_point_for_cables]))
    cable_feat.setAttribute('fiber_count', fiber_count)
    cable_feat.setAttribute('install_method', f'Магістраль SFU-
{cluster_id}')
    sfu_cables_to_add.append(cable_feat)

    node_provider.addFeatures(sfu_boxes_to_add)
    cable_provider.addFeatures(sfu_cables_to_add)
    print(f"...Створено {len(sfu_boxes_to_add)} SFU коробок та
{len(sfu_cables_to_add)} кабелів (4-8F).")
    sfu_drops_to_add = []
    problem_drops_count = 0

```

```

for cluster_id, data in cluster_data.items():
    box_point = sfu_boxes_by_cluster[cluster_id].geometry().asPoint()
    box_point_utm = QgsGeometry.fromPointXY(box_point)
    box_point_utm.transform(transform_to_utm)

    for feat in data['features']:
        orig_fid = feat['orig_fid']
        sub_geom = original_sfu_features_map[orig_fid].geometry().centroid()
        sub_point = sub_geom.asPoint()
        sub_point_utm = QgsGeometry.fromPointXY(sub_point)
        sub_point_utm.transform(transform_to_utm)

        distance = box_point_utm.distance(sub_point_utm)

        line_geom = QgsGeometry.fromPolylineXY([box_point,
sub_point])
        drop_feat = QgsFeature(drop_layer.fields()) # Використовуємо
оновлені поля
        drop_feat.setGeometry(line_geom)
        drop_feat.setAttribute('drop_type', 'Зовнішній')
        drop_feat.setAttribute('distance', distance)

        if distance > DROP_CABLE_LIMIT_METERS:
            problem_drops_count += 1
            drop_feat.setAttribute('problem', 'Так')
        else:
            drop_feat.setAttribute('problem', 'Hi')

```

```

        sfu_drops_to_add.append(drop_feat)

        drop_provider.addFeatures(sfu_drops_to_add)
        print(f"...Створено {len(sfu_drops_to_add)} SFU патчкордів.")
        print(f"...Проблемних відводів (>500м): {problem_drops_count}")

    else:
        print("...SFU будинки для кластеризації не знайдено. Пропускаю
Крок 4.")
        print(f"...Крок 5: Налаштування стилів для ліміту
{DROP_CABLE_LIMIT_METERS}м...")

        if cable_layer.isEditable():
            cable_layer.commitChanges()
        if drop_layer.isEditable():
            drop_layer.commitChanges()
        symbol_ok = QgsLineSymbol.createSimple({'color': '#555555', 'width':
'0.5'})
        symbol_bad = QgsLineSymbol.createSimple({'color': '#FF0000', 'width':
'1.2'})
        root_rule = QgsRuleBasedRenderer.Rule(None)

        rule_ok = QgsRuleBasedRenderer.Rule(symbol_ok)
        rule_ok.setLabel(f"OK (<= {DROP_CABLE_LIMIT_METERS}м)")
        rule_ok.setFilterExpression(f"$length <=
{DROP_CABLE_LIMIT_METERS}")

        rule_bad = QgsRuleBasedRenderer.Rule(symbol_bad)
        rule_bad.setLabel(f"Проблема (> {DROP_CABLE_LIMIT_METERS}м)")

```

```

rule_bad.setFilterExpression(f"$length
{DROP_CABLE_LIMIT_METERS}")

root_rule.appendChild(rule_ok)
root_rule.appendChild(rule_bad)

renderer = QgsRuleBasedRenderer(root_rule)
drop_layer.setRenderer(renderer)
drop_layer.triggerRepaint()

print("...Стилі для ліміту 500м застосовано.")

# --- ЗАВЕРШЕННЯ ---
iface.mapCanvas().refreshAllLayers()
iface.messageBar().pushMessage(
    "Інженерний Аналіз Завершено!",
    f"Оптимізована мережа: менше волокон, більше кластерів, ліміт
500м.",
    level=3,
    duration=7
)

print("\n===ОПТИМІЗОВАНИЙ АНАЛІЗ ЗАВЕРШЕНО===")

```

#### Скрипт FPV - 1 частина

```

# -*- coding: utf-8 -*-
import processing
from qgis.core import (
    QgsProject,    QgsVectorLayer,    QgsRasterLayer,
    QgsCoordinateReferenceSystem,

```

QgsCoordinateTransform, QgsPointXY, QgsGeometry

)

---



---

# НАЛІАШТУВАННЯ АНАЛІЗУ FPV

---



---

Downloader)

DEM\_LAYER\_NAME = "N49E034\_N49E034"

OBSERVER\_LON = 34.5512

OBSERVER\_LAT = 49.5891

OBSERVER\_HEIGHT\_METERS = 300

ANALYSIS\_RADIUS\_METERS = 10000.0 # 10 км

OUTPUT\_PATH = " C:/example/path/FPV\_Coverage\_GDAL.tif"

#

---



---

print("--- Початок 3D-аналізу зони видимості (FPV) ---")

# 1. Знаходимо DEM-шар у проекті

dem\_layer =

QgsProject.instance().mapLayersByName(DEM\_LAYER\_NAME)

if not dem\_layer:

print(f"ПОМИЛКА: Не можу знайти шар з назвою  
'{DEM\_LAYER\_NAME}'. Перевірте назву.")

raise Exception("DEM не знайдено.")

dem\_layer = dem\_layer[0]

crs\_dem = dem\_layer.crs()

print(f"...DEM шар '{DEM\_LAYER\_NAME}' знайдено. Його 'CRS':  
{crs\_dem.authid()}")

# 2. Налаштування координат та радіусу

```
observer_coords_str = f"{OBSERVER_LON},{OBSERVER_LAT}"
radius_in_degrees = ANALYSIS_RADIUS_METERS / 111000.0
```

```
print(f"...Точка спостерігача: {observer_coords_str} (у градусах)")
```

```
print(f"...Висота спостерігача: {OBSERVER_HEIGHT_METERS} м")
```

```
print(f"...Радіус аналізу: {ANALYSIS_RADIUS_METERS} м
```

```
(приблизно {radius_in_degrees:.5f} градусів)")
```

# 3. Запускаємо алгоритм GDAL 'Viewshed'

```
print("...Запуск алгоритму GDAL 'Viewshed'. Це може зайняти  
хвилину...")
```

```
viewshed_params = {
```

```
    'INPUT': dem_layer,
```

```
    'OBSERVER': observer_coords_str,
```

```
    'OBSERVER_HEIGHT': OBSERVER_HEIGHT_METERS,
```

```
    'TARGET_HEIGHT': 0.0,
```

```
    'MAX_DISTANCE': radius_in_degrees,
```

```
    'MODE': 0,
```

```
    'OUTPUT': OUTPUT_PATH
```

```
}
```

```
try:
```

```
    processing.run("gdal:viewshed", viewshed_params)
```

```
    print("...Аналіз зони видимості успішно завершено.")
```

```
except Exception as e:
```

```
    print(f"ПОМИЛКА при виконанні GDAL: {e}")
```

```
    raise e
```

```

# 4. Додаємо результат на карту
result_layer = QgsRasterLayer(OUTPUT_PATH, "FPV Зона Покриття
(GDAL)")
if not result_layer.isValid():
    print(f"ПОМИЛКА: Не вдалося завантажити результат з
{OUTPUT_PATH}")
else:
    QgsProject.instance().addMapLayer(result_layer)
    print(f"...Шар 'FPV Зона Покриття (GDAL)' додано на карту.")

print("--- 3D-аналіз завершено! ---")

```

#### Скрипт FPV - 2 частина

```

# -*- coding: utf-8 -*-
import processing
from qgis.core import (
    QgsProject,    QgsVectorLayer,    QgsRasterLayer,
    QgsCoordinateReferenceSystem,
    QgsCoordinateTransform, QgsPointXY, QgsGeometry, QgsFeature,
    QgsVectorFileWriter,
    QgsFields, QgsField, QgsWkbTypes
)
from qgis.PyQt.QtCore import QVariant
import os

#
=====
=====

# НАЛАШТУВАННЯ АНАЛІЗУ FPV

```

#

# 1. Назва вашого шару ЗОНИ ВИДИМОСТІ

VIEWSHED\_LAYER\_NAME = "FPV Зона Покриття (GDAL)"

# 2. Координати точок (WGS84)

START\_LON = 34.50 # ЗМІНІТЬ

START\_LAT = 49.59 # ЗМІНІТЬ

END\_LON = 34.60 # ЗМІНІТЬ

END\_LAT = 49.56 # ЗМІНІТЬ

# 3. Шлях для збереження

SAVE\_PATH = " C:/example/path/"

#

# АВТОМАТИЧНІ НАЛАШТУВАННЯ

#

```
VIEWSHED_UTM_PATH = f"{SAVE_PATH}FPV_Viewshed_UTM.tif"
COST_SURFACE_PATH = f"{SAVE_PATH}FPV_Cost_Surface.tif"
ACCESSIBLE_AREA_PATH
f"{SAVE_PATH}FPV_Accessible_Area.shp"
FINAL_PATH_SHP = f"{SAVE_PATH}FPV_Optimal_Path.shp"
```

```
RELAY_POINTS_SHP = f"{SAVE_PATH}FPV_Relay_Points.shp"
DIRECT_PATH_SHP = f"{SAVE_PATH}FPV_Direct_Path.shp"
```

```
print("--- СКРИПТ ДЛЯ ПОШУКУ ШЛЯХУ ДРОНА ЧЕРЕЗ ЗОНИ  
ПОКРИТТЯ ---")
```

```
# Створення папки
```

```
os.makedirs(SAVE_PATH, exist_ok=True)
```

```
# --- 1. Знаходимо шар Viewshed ---
```

```
viewshed_layer =
```

```
QgsProject.instance().mapLayersByName(VIEWSHED_LAYER_NAME)
```

```
if not viewshed_layer:
```

```
    raise Exception(f"Шар '{VIEWSHED_LAYER_NAME}' не знайдено!")
```

```
viewshed_layer = viewshed_layer[0]
```

```
print(f"Шар '{VIEWSHED_LAYER_NAME}' знайдено")
```

```
# --- 2. Координатні системи ---
```

```
crs_latlon = QgsCoordinateReferenceSystem("EPSG:4326")
```

```
crs_utm = QgsCoordinateReferenceSystem("EPSG:32636")
```

```
transform_to_utm = QgsCoordinateTransform(crs_latlon, crs_utm,
```

```
QgsProject.instance())
```

```
# --- 3. Перепроєктування в UTM ---
```

```
if not os.path.exists(VIEWSHED_UTM_PATH):
```

```
    print("Перепроєктування viewshed в UTM...")
```

```
    processing.run("gdal:warp", {
```

```
        'INPUT': viewshed_layer,
```

```
        'SOURCE_CRS': crs_latlon,
```

```
'TARGET_CRS': crs_utm,
'RESAMPLING': 0,
'NODATA': 0,
'OUTPUT': VIEWSHED_UTM_PATH
})
```

```
viewshed_utm_layer = QgsRasterLayer(VIEWSHED_UTM_PATH,
"Viewshed UTM")
```

```
if not viewshed_utm_layer.isValid():
```

```
    raise Exception("Помилка створення Viewshed UTM")
```

```
print(f"Розмір
```

растру:

```
{viewshed_utm_layer.width()}x{viewshed_utm_layer.height()}")
```

```
# --- 4. Створення Cost Surface ---
```

```
print("Створення карти вартості...")
```

```
if not os.path.exists(COST_SURFACE_PATH):
```

```
    # Формула: низька вартість = хороше покриття, висока = погане
```

```
    processing.run("gdal:rastercalculator", {
```

```
        'INPUT_A': viewshed_utm_layer,
```

```
        'BAND_A': 1,
```

```
        'FORMULA': '(A==255)*1 + (A==0)*1000', # 1 - де видно, 1000 - де
```

не видно

```
        'OUTPUT': COST_SURFACE_PATH
```

```
    })
```

```
cost_surface_layer = QgsRasterLayer(COST_SURFACE_PATH, "Cost
Surface")
```

```
print("Карта вартості створена")
```

```
# --- 5. Трансформація координат ---
def transform_coordinates(lon, lat):
    point = QgsGeometry.fromPointXY(QgsPointXY(lon, lat))
    point.transform(transform_to_utm)
    return point.asPoint()

start_point_utm = transform_coordinates(START_LON, START_LAT)
end_point_utm = transform_coordinates(END_LON, END_LAT)

print(f"Старт UTM: {start_point_utm.x():.2f}, {start_point_utm.y():.2f}")
print(f"Фініш UTM: {end_point_utm.x():.2f}, {end_point_utm.y():.2f}")

# --- 6. Створюємо прямий шлях для аналізу ---
print("Створення прямого шляху для аналізу...")
fields = QgsFields()
fields.append(QgsField("id", QVariant.Int))
fields.append(QgsField("type", QVariant.String))

writer = QgsVectorFileWriter(
    DIRECT_PATH_SHP,
    "UTF-8",
    fields,
    QgsWkbTypes.LineString,
    crs_utm,
    "ESRI Shapefile"
)

line_geometry = QgsGeometry.fromPolylineXY([start_point_utm,
end_point_utm])
feature = QgsFeature()
```

```
feature.setGeometry(line_geometry)
feature.setAttributes([1, "direct_path"])
writer.addFeature(feature)
del writer

direct_path_layer = QgsVectorLayer(DIRECT_PATH_SHP, "Прямий
шлях", "ogr")
QgsProject.instance().addMapLayer(direct_path_layer)
print("Прямий шлях створений")

# --- 7. Аналіз перетину з чорними зонами ---
print("Аналіз перетину з зонами без покриття...")

# Створюємо вектор зон
if not os.path.exists(ACCESSIBLE_AREA_PATH):
    processing.run("gdal:polygonize", {
        'INPUT': viewshed_utm_layer,
        'BAND': 1,
        'FIELD': 'value',
        'OUTPUT': ACCESSIBLE_AREA_PATH
    })

zones_layer = QgsVectorLayer(ACCESSIBLE_AREA_PATH, "Зони
покриття", "ogr")

# Знаходимо чорні зони (де value = 0)
black_zones_count = 0
relay_points = []

for feature in zones_layer.getFeatures():
```

```
if feature['value'] == 0: # Чорна зона - без покриття
    black_zones_count += 1
    black_zone_geom = feature.geometry()

# Перевіряємо перетин з нашим шляхом
if line_geometry.intersects(black_zone_geom):
    intersection = line_geometry.intersection(black_zone_geom)
    if intersection and not intersection.isEmpty():
        # Додаємо точку ретранслятора в центр перетину
        relay_points.append(intersection.centroid())

print(f"Знайдено {black_zones_count} чорних зон (без покриття)")
print(f"Прямий шлях перетинає {len(relay_points)} зон без покриття")

# --- 8. Створюємо точки ретрансляції ---
if relay_points:
    print("Створення точок ретрансляції...")

fields = QgsFields()
fields.append(QgsField("id", QVariant.Int))
fields.append(QgsField("type", QVariant.String))
fields.append(QgsField("description", QVariant.String))

writer = QgsVectorFileWriter(
    RELAY_POINTS_SHP,
    "UTF-8",
    fields,
    QgsWkbTypes.Point,
    crs_utm,
    "ESRI Shapefile"
```

```

)

for i, point_geom in enumerate(relay_points):
    feature = QgsFeature()
    feature.setGeometry(point_geom)
    feature.setAttributes([i+1, "ретранслятор", f"Точка {i+1} для
покриття"])
    writer.addFeature(feature)
    point = point_geom.asPoint()
    print(f" Ретранслятор {i+1}: {point.x():.2f}, {point.y():.2f}")

del writer
    relay_layer = QgsVectorLayer(RELAY_POINTS_SHP, "Точки
ретрансляції FPV", "ogr")
    QgsProject.instance().addMapLayer(relay_layer)
    print("Точки ретрансляції створені")

# --- 9. Пробуємо знайти оптимальний шлях через GRASS ---
print("Спроба побудови оптимального шляху...")

try:
    # Використовуємо r.walk для пошуку шляху
    processing.run("grass7:r.walk", {
        'elevation': cost_surface_layer,
        'friction': cost_surface_layer,
        'start_points':
f'Point?crs=EPSG:32636&field=id:integer&point={start_point_utm.x()}%20{start
_point_utm.y()}',

```

```

'output': f'{SAVE_PATH}walk_output.tif',
'GRASS_REGION_PARAMETER': viewshed_utm_layer.extent(),
'GRASS_REGION_CELLSIZE_PARAMETER': 0,
})

# Потім r.drain для виділення шляху
processing.run("grass7:r.drain", {
    'input': f'{SAVE_PATH}walk_output.tif',
    'start_points':
f'Point?crs=EPSG:32636&field=id:integer&point={end_point_utm.x()}%20{end_
point_utm.y()}',
    'output': FINAL_PATH_SHP,
    'GRASS_REGION_PARAMETER': viewshed_utm_layer.extent(),
})

    optimal_path_layer = QgsVectorLayer(FINAL_PATH_SHP,
"Оптимальний шлях FPV", "ogr")
    if optimal_path_layer.isValid() and optimal_path_layer.featureCount() > 0:
        QgsProject.instance().addMapLayer(optimal_path_layer)
        print("Оптимальний шлях знайдений!")
    else:
        raise Exception("Шлях не знайдено")

except Exception as e:
    print(f"GRASS не вдалось знайти шлях: {e}")
    print("Використовуюю прямий шлях з аналізом...")

```

```
# Копіюємо прямий шлях як оптимальний
    optimal_path_layer = QgsVectorLayer(DIRECT_PATH_SHP, "Прямий
шлях (потрібні ретранслятори)", "ogr")
    QgsProject.instance().addMapLayer(optimal_path_layer)

# --- 10. Застосовуємо стилі ---
print("Застосування стилів...")

# Стиль для прямого шляху
try:
    from qgis.core import QgsLineSymbol, QgsSingleSymbolRenderer
    symbol_red = QgsLineSymbol.createSimple({'color': 'red', 'width': '1.0'})
    direct_path_layer.renderer().setSymbol(symbol_red)
    direct_path_layer.triggerRepaint()
except:
    pass

# Стиль для оптимального шляху
try:
    symbol_green = QgsLineSymbol.createSimple({'color': 'green', 'width':
'2.0'})
    if 'optimal_path_layer' in locals() and optimal_path_layer.isValid():
        optimal_path_layer.renderer().setSymbol(symbol_green)
        optimal_path_layer.triggerRepaint()
except:
    pass

# --- 11. Фінальний звіт ---
print("\n" + "="*50)
print("ФІНАЛЬНИЙ ЗВІТ АНАЛІЗУ FPV ПОКРИТТЯ")
```

```

print("="*50)
print(f"Стартова точка: {START_LON}, {START_LAT}")
print(f"Кінцева точка: {END_LON}, {END_LAT}")
print(f"Довжина прямого шляху: {line_geometry.length():.2f} метрів")

if relay_points:
    print(f" УВАГА: Шлях перетинає {len(relay_points)} зон без FPV
покриття!")

# Оновлюємо карту
iface.mapCanvas().refreshAllLayers()


```

Скрипт Starlink - 1 частина

```

# -*- coding: utf-8 -*-
import processing
from qgis.core import (
    QgsProject,    QgsVectorLayer,    QgsRasterLayer,
    QgsCoordinateReferenceSystem,
    QgsCoordinateTransform, QgsFeature, QgsVectorFileWriter, QgsFields,
    QgsField, QgsWkbTypes
)
from PyQt5.QtCore import QVariant
import os

# --- НАЛАШТУВАННЯ ---
DEM_LAYER_NAME = "N49E034_N49E034"
BUILDINGS_PATH = " C:/example/path/file.shp"
SAVE_PATH = " C:/example/path/"

# --- ПАРАМЕТРИ ---

```

```
HOUSE_HEIGHT = 8.0
APARTMENT_HEIGHT = 30.0
SAT_ELEVATION = 25.0
SAT_AZIMUTH = 180.0
```

```
print("--- STARLINK v7.0 (STRICT UTM) ---")
if not os.path.exists(SAVE_PATH): os.makedirs(SAVE_PATH)
```

```
# 1. Завантаження
```

```
dem_layer =
QgsProject.instance().mapLayersByName(DEM_LAYER_NAME)[0]
buildings_layer = QgsVectorLayer(BUILDINGS_PATH, "Buildings", "ogr")
crs_utm = QgsCoordinateReferenceSystem("EPSG:32636")
```

```
# 2. Перепроєктування DEM (Градуси -> Метри)
```

```
print("...1. Конвертація DEM в метри...")
dem_utm_path = f"{SAVE_PATH}1_DEM_UTM.tif"
processing.run("gdal:warp", {
    'INPUT': dem_layer, 'TARGET_CRS': crs_utm, 'RESAMPLING': 1,
    'OUTPUT': dem_utm_path
})
```

```
# ВАЖЛИВО: Завантажуємо UTM шар назад, щоб отримати його точні
розміри в метрах
```

```
dem_utm = QgsRasterLayer(dem_utm_path, "DEM UTM")
if not dem_utm.isValid(): raise Exception("Помилка створення DEM
UTM")
```

```
# Отримуємо розмір пікселя та межі SAME з МЕТРИЧНОГО ШАРУ
```

```
pixel_size_x = dem_utm.rasterUnitsPerPixelX()
pixel_size_y = dem_utm.rasterUnitsPerPixelY()
```

```

extent = dem_utm.extent()
print(f"...Розмір пікселя: {pixel_size_x:.2f} x {pixel_size_y:.2f} метрів")

# 3. Підготовка векторів (Додавання висоти Z)
print("...2. Призначення висот будівлям...")
buildings_z_path = f"{SAVE_PATH}2_Buildings_Z.shp"
fields = QgsFields(); fields.append(QgsField("Z_VALUE",
QVariant.Double))
# Створюємо шейп-файл одразу в UTM
writer = QgsVectorFileWriter(buildings_z_path, "UTF-8", fields,
QgsWkbTypes.Polygon, crs_utm, "ESRI Shapefile")

# Трансформація: Шар Будинків (Градуси) -> UTM (Метри)
tr = QgsCoordinateTransform(buildings_layer.crs(), crs_utm,
QgsProject.instance())

for feat in buildings_layer.getFeatures():
    new_feat = QgsFeature(fields)
    geom = feat.geometry()
    geom.transform(tr) # Конвертуємо геометрію в метри
    new_feat.setGeometry(geom)

    h = HOUSE_HEIGHT
    try:
        if feat['building'] == 'apartments': h = APARTMENT_HEIGHT
        elif feat['building'] == 'industrial': h = 15.0
    except: pass
    new_feat.setAttributes([h])
    writer.addFeature(new_feat)
del writer

```

# 4. Растеризація (Вектор -> Растр висот)

# Тепер ми растеризуємо Метричний вектор на Метричну сітку. Це має працювати.

```
print("...3. Растеризація будівель...")
buildings_raster_path = f"{SAVE_PATH}3_Buildings_Raster.tif"
processing.run("gdal:rasterize", {
    'INPUT': buildings_z_path,
    'FIELD': 'Z_VALUE',
    'UNITS': 1, # Georeferenced units (тепер це точно метри)
    'WIDTH': pixel_size_x, 'HEIGHT': pixel_size_y,
    'EXTENT': extent,
    'NODATA': 0,
    'OUTPUT': buildings_raster_path
})
```

# 5. Створення DSM (DEM + Будинки)

```
print("...4. Створення DSM...")
dsm_path = f"{SAVE_PATH}4_Starlink_DSM.tif"
processing.run("gdal:rastercalculator", {
    'INPUT_A': dem_utm_path, 'BAND_A': 1,
    'INPUT_B': buildings_raster_path, 'BAND_B': 1,
    'FORMULA': 'A + B',
    'OUTPUT': dsm_path
})
```

# 6. Hillshade

```
print("...5. Аналіз тіней...")
obstructions_path = f"{SAVE_PATH}5_Starlink_Obstructions.tif"
processing.run("gdal:hillshade", {
```

```

    'INPUT': dsm_path,
    'AZIMUTH': SAT_AZIMUTH, 'ALTITUDE': SAT_ELEVATION,
    'Z_FACTOR': 1.0,
    'OUTPUT': obstructions_path
})

```

```
# 7. Результат
```

```
result = QgsRasterLayer(obstructions_path, "Starlink Результат")
```

```
if result.isValid():
```

```
    QgsProject.instance().addMapLayer(result)
```

```
    print("Карта створена.")
```

```
else:
```

```
    print("Помилка. Перевірте папку результатів.")
```

#### Скрипт Starlink - 2 частина

```

# -*- coding: utf-8 -*-
import processing
from qgis.core import (
    QgsProject,    QgsVectorLayer,    QgsRasterLayer,
    QgsCoordinateReferenceSystem,
    QgsFeature, QgsVectorFileWriter, QgsFields, QgsField, QgsWkbTypes,
    QgsCoordinateTransform
)
from PyQt5.QtCore import QVariant
import os

# --- НАЛАШТУВАННЯ ---
DEM_LAYER_NAME = "N49E034_N49E034"
BUILDINGS_PATH = "C:/example/path/file.shp"

```

```

SAVE_PATH = " C:/example/path/"

# --- ПАРАМЕТРИ МОДЕЛЮВАННЯ ---
HOUSE_HEIGHT = 8.0    # Висота приватного будинку
APARTMENT_HEIGHT = 30.0 # Висота 9-поверхівки
SAT_AZIMUTH = 180.0    # Супутники на Півдні (антена дивиться на
Північ)
SAT_ELEVATION = 25.0    # Мінімальний кут сигналу (все що нижче -
перешкода)

print("--- СТАРТ МОДЕЛЮВАННЯ STARLINK (v7.0) ---")

if not os.path.exists(SAVE_PATH): os.makedirs(SAVE_PATH)

# 1. Завантаження шарів
dem_layer = QgsProject.instance().mapLayersByName(DEM_LAYER_NAME)
if not dem_layer: raise Exception("Шар DEM не знайдено на карті!")
dem_layer = dem_layer[0]

buildings_layer = QgsVectorLayer(BUILDINGS_PATH, "Buildings", "ogr")
if not buildings_layer.isValid():
    # Спроба знайти в альтернативному місці
    BUILDINGS_PATH = " C:/example/path/file.shp"
    buildings_layer = QgsVectorLayer(BUILDINGS_PATH, "Buildings",
"ogr")
if not buildings_layer.isValid():
    raise Exception("Не можу знайти файл 1.shp. Перевірте шлях у
скрипті!")

```

```

crs_utm = QgsCoordinateReferenceSystem("EPSG:32636") # Метрична
система

# 2. Перепроєктування DEM в метри
print("...1. Конвертація рельєфу в метри...")
dem_utm_path = f"{SAVE_PATH}1_DEM_UTM.tif"
processing.run("gdal:warp", {
    'INPUT': dem_layer, 'TARGET_CRS': crs_utm, 'RESAMPLING': 1,
    'OUTPUT': dem_utm_path
})
dem_utm = QgsRasterLayer(dem_utm_path, "DEM UTM")
if not dem_utm.isValid(): raise Exception("Помилка DEM UTM")
# Отримуємо точні розміри пікселя з метричного файлу
pixel_size_x = dem_utm.rasterUnitsPerPixelX()
pixel_size_y = dem_utm.rasterUnitsPerPixelY()
extent = dem_utm.extent()

# 3. Підготовка векторів (Додавання висоти Z)
print("...2. Призначення висот будівлям...")
buildings_z_path = f"{SAVE_PATH}2_Buildings_Z.shp"
fields = QgsFields(); fields.append(QgsField("Z_VALUE",
QVariant.Double))
writer = QgsVectorFileWriter(buildings_z_path, "UTF-8", fields,
QgsWkbTypes.Polygon, crs_utm, "ESRI Shapefile")

tr = QgsCoordinateTransform(buildings_layer.crs(), crs_utm,
QgsProject.instance())

for feat in buildings_layer.getFeatures():
    new_feat = QgsFeature(fields)

```

```

geom = feat.geometry(); geom.transform(tr)
new_feat.setGeometry(geom)

h = HOUSE_HEIGHT
try:
    if feat['building'] == 'apartments': h = APARTMENT_HEIGHT
    elif feat['building'] == 'industrial': h = 15.0
except: pass
new_feat.setAttributes([h]); writer.addFeature(new_feat)
del writer

```

# 4. Растеризація (Вектор -> Растр висот)

```

print("...3. Растеризація будівель...")
buildings_raster_path = f"{SAVE_PATH}3_Buildings_Raster.tif"
processing.run("gdal:rasterize", {
    'INPUT': buildings_z_path,
    'FIELD': 'Z_VALUE',
    'UNITS': 1, # Georeferenced units (метри)
    'WIDTH': pixel_size_x, 'HEIGHT': pixel_size_y,
    'EXTENT': extent,
    'NODATA': 0, 'OUTPUT': buildings_raster_path
})

```

# 5. Створення DSM (Алгебра карт: Рельєф + Будинки)

```

print("...4. Створення DSM (Рельєф + Будинки)...")
dsm_path = f"{SAVE_PATH}4_Starlink_DSM.tif"
processing.run("gdal:rastercalculator", {
    'INPUT_A': dem_utm_path, 'BAND_A': 1,
    'INPUT_B': buildings_raster_path, 'BAND_B': 1,
    'FORMULA': 'A + B',

```

```

'OUTPUT': dsm_path
})

# 6. Hillshade (Аналіз перешкод)
print("...5. Аналіз перешкод (Hillshade)...")
obstructions_path = f"{SAVE_PATH}5_Starlink_Obstructions.tif"
processing.run("gdal:hillshade", {
    'INPUT': dsm_path,
    'AZIMUTH': SAT_AZIMUTH, 'ALTITUDE': SAT_ELEVATION,
    'Z_FACTOR': 1.0, 'OUTPUT': obstructions_path
})

```

```

# 7. Результат
result = QgsRasterLayer(obstructions_path, "Starlink Перешкоди")
if result.isValid():
    QgsProject.instance().addMapLayer(result)
    print("Карта створена.")
    print("Чорні зони - це тіні від будинків (мертві зони).")
else:
    print("Помилка відображення.")

```

#### Скрипт 5G - 1 частина

```

# -*- coding: utf-8 -*-
import processing
from qgis.core import (
    QgsProject, QgsVectorLayer, QgsRasterLayer, QgsPointXY,
    QgsGeometry,
    QgsCoordinateReferenceSystem, QgsCoordinateTransform
)
import os

```

```
# --- НАЛАШТУВАННЯ ---
# Шлях до файлу DSM
DSM_PATH = " C:/example/path/4_Starlink_DSM.tif"
SAVE_PATH = " C:/example/path/"
OUTPUT_PATH = f"{SAVE_PATH}5G_Coverage.tif"

# --- ПАРАМЕТРИ 5G ---
# Координати (Градуси) - ми їх переведемо в метри автоматично
BS_LON = 34.5512
BS_LAT = 49.5891

BS_HEIGHT = 25.0 # Антена на даху (5м над поверхнею)
RADIUS = 500.0 # Радіус дії (метри)

print("--- МОДЕЛЮВАННЯ 5G (v1.2 Coord Fix) ---")

if not os.path.exists(SAVE_PATH): os.makedirs(SAVE_PATH)

# 1. Завантаження DSM
if not os.path.exists(DSM_PATH):
    raise Exception(f"Файл DSM не знайдено: {DSM_PATH}\nСпочатку
виконайте моделювання Starlink!")

dsm_layer = QgsRasterLayer(DSM_PATH, "DSM (для 5G)")
if not dsm_layer.isValid(): raise Exception("Шар DSM невалідний.")
QgsProject.instance().addMapLayer(dsm_layer)

# 2. Переведен координат (Градуси -> Метри)
print("...Конвертація координат бази...")
```

```
crs_src = QgsCoordinateReferenceSystem("EPSG:4326") # WGS 84
(Градуси)
crs_dest = QgsCoordinateReferenceSystem("EPSG:32636") # UTM 36N
(Метри)
tr = QgsCoordinateTransform(crs_src, crs_dest, QgsProject.instance())

pt = QgsPointXY(BS_LON, BS_LAT)
pt_utm = tr.transform(pt) # Переводимо точку в метри

# Формуємо рядок для GDAL: "X,Y"
observer_coords = f"{pt_utm.x()},{pt_utm.y()}"

print(f"...Координати UTM: {observer_coords}")
print(f"...Висота антени: {BS_HEIGHT}м")

# 3. Аналіз (Viewshed)
print("...Запуск розрахунку зони покриття...")
processing.run("gdal:viewshed", {
    'INPUT': dsm_layer,
    'OBSERVER': observer_coords, # Тепер тут правильні метри!
    'OBSERVER_HEIGHT': BS_HEIGHT,
    'TARGET_HEIGHT': 1.5,
    'MAX_DISTANCE': RADIUS, # Радіус теж в метрах
    'MODE': 0,
    'OUTPUT': OUTPUT_PATH
})

# 4. Результат
res = QgsRasterLayer(OUTPUT_PATH, "5G Покриття (mmWave)")
if res.isValid():
```

```

QgsProject.instance().addMapLayer(res)
print("Білі зони = 5G Сигнал.")

# Фарбуємо в зелений для краси
try:
    from qgis.core import QgsColorRampShader, QgsRasterShader,
QgsSingleBandPseudoColorRenderer
    from PyQt5.QtGui import QColor
    fcn = QgsColorRampShader()
    fcn.setColorRampType(QgsColorRampShader.Discrete)
    lst = [
        QgsColorRampShader.ColorRampItem(0, QColor(0,0,0,0), 'Тінь'),
        QgsColorRampShader.ColorRampItem(255, QColor(0,255,0),
'Sигнал')
    ]
    fcn.setColorRampItemList(lst)
    shader = QgsRasterShader()
    shader.setRasterShaderFunction(fcn)
    renderer = QgsSingleBandPseudoColorRenderer(res.dataProvider(), 1,
shader)
    res.setRenderer(renderer)
    res.triggerRepaint()
except: pass
else:
    print("Помилка створення шару.")

```

Скрипт 5G - 2 частина

```

# -*- coding: utf-8 -*-
import processing

```

```

from qgis.core import (
    QgsProject, QgsVectorLayer, QgsRasterLayer, QgsPointXY,
    QgsGeometry,
    QgsCoordinateReferenceSystem, QgsCoordinateTransform
)
import os

```

```
# --- НАЛАШТУВАННЯ ---
```

```
DSM_PATH = " C:/example/path/4_Starlink_DSM.tif"
```

```
SAVE_PATH = " C:/example/path/"
```

```
OUTPUT_PATH = f"{SAVE_PATH}5G_Coverage_GRID_VRT.tif" #
```

Змінено назву, щоб уникнути кешування

```
# --- ПАРАМЕТРИ 5G ---
```

```
BS_LON = 34.5512
```

```
BS_LAT = 49.5891
```

```
BS_HEIGHT = 25.0
```

```
RADIUS = 300.0 # Зберігаємо 300м для перекриття
```

```
STEP = 250.0
```

```
GRID_SIZE = 3
```

```
print("--- МОДЕЛЮВАННЯ 5G (v2.1 VRT Fix) ---")
```

```
if not os.path.exists(SAVE_PATH): os.makedirs(SAVE_PATH)
```

```
# 1. Завантаження DSM (як у попередній версії)
```

```
if not os.path.exists(DSM_PATH):
```

```
    raise Exception(f"Файл DSM не знайдено: {DSM_PATH}\nСпочатку  
виконайте моделювання Starlink!")
```

```
dsm_layer = QgsRasterLayer(DSM_PATH, "DSM (для 5G)")  
if not dsm_layer.isValid(): raise Exception("Шар DSM невалідний.")
```

```
# 2. Налаштування трансформації координат
```

```
crs_src = QgsCoordinateReferenceSystem("EPSG:4326")  
crs_dest = QgsCoordinateReferenceSystem("EPSG:32636")  
tr = QgsCoordinateTransform(crs_src, crs_dest, QgsProject.instance())
```

```
pt = QgsPointXY(BS_LON, BS_LAT)  
center_utm = tr.transform(pt)
```

```
# 3. Генерація точок та Viewshed-аналіз
```

```
print("...Запуск розрахунку зони покриття для 9 станцій...")  
viewshed_outputs = []  
offset_range = range(-int(GRID_SIZE / 2), int(GRID_SIZE / 2) + 1)  
run_count = 0
```

```
for dx in offset_range:
```

```
    for dy in offset_range:
```

```
        run_count += 1
```

```
        new_x = center_utm.x() + dx * STEP
```

```
        new_y = center_utm.y() + dy * STEP
```

```
        observer_coords = f"{new_x},{new_y}"
```

```
# Шлях для тимчасового файлу Viewshed
```

```
temp_output_path = f"{SAVE_PATH}temp_vs_{run_count}.tif"
```

```
viewshed_outputs.append(temp_output_path)
```

```
# Запуск Viewshed
```

```
processing.run("gdal:viewshed", {
```

```

'INPUT': dsm_layer,
'OBSERVER': observer_coords,
'OBSERVER_HEIGHT': BS_HEIGHT,
'TARGET_HEIGHT': 1.5,
'MAX_DISTANCE': RADIUS,
'MODE': 0,
'OUTPUT': temp_output_path
})
print(f"...Станція {run_count} розрахована.")

```

# 4. Об'єднання результатів за допомогою VRT

```

print("...Об'єднання 9 результатів в єдиний ВІРТУАЛЬНИЙ растр...")
processing.run("gdal:buildvirtualraster", {
    'INPUT': viewshed_outputs,
    'OUTPUT': f"{SAVE_PATH}merged_vrt.vrt", # Створюємо VRT файл
    'RESOLVE': False,
    'SEPARATE': False,
    'ALLOW_OTHER_DS': False,
    'PROJECT_EXTENT': False,
    'SRC_NODATA': '0' # Важливо: Ігноруємо '0' (чорний) для
    правильного накладання
})

```

# 5. Результат та очищення

```

res = QgsRasterLayer(f"{SAVE_PATH}merged_vrt.vrt", "5G Покриття
СІТКА VRT")

```

```

if res.isValid():

```

```

    QgsProject.instance().addMapLayer(res)
    print(f"Моделювання сітки 3x3 завершено.")

```

```

# Фарбування в зелений/чорний
try:
    from qgis.core import QgsColorRampShader, QgsRasterShader,
QgsSingleBandPseudoColorRenderer
    from PyQt5.QtGui import QColor
    fcn = QgsColorRampShader()
    fcn.setColorRampType(QgsColorRampShader.Discrete)
    lst = [
        # Встановлюємо 0 як прозорий або дуже темний, щоб бачити, де
нема покриття
        QgsColorRampShader.ColorRampItem(0, QColor(0,0,0), 'Тінь
(Немає LoS)'),
        QgsColorRampShader.ColorRampItem(255, QColor(0,255,0),
'Sигнал (Є LoS)')
    ]
    fcn.setColorRampItemList(lst)
    shader = QgsRasterShader()
    shader.setRasterShaderFunction(fcn)
    renderer = QgsSingleBandPseudoColorRenderer(res.dataProvider(), 1,
shader)
    res.setRenderer(renderer)
    res.triggerRepaint()
except Exception as e:
    print(f"Помилка застосування стилю: {e}")

# Видалення тимчасових файлів
# (Цей крок краще виконати вручну після перевірки, щоб не
видалити VRT, але в автоматичному режимі варто залишити)
# for temp_file in viewshed_outputs:

```

```
# if os.path.exists(temp_file):  
#     os.remove(temp_file)
```

```
else:
```

```
    print("Помилка створення об'єданого шару.")
```

## Додаток Г

Тези на тему “Геоінформаційні системи в телекомунікаційних мережах”. Збірник тез за матеріалами II Міжнародної науково-практичної конференції студентів, аспірантів та молодих вчених – Полтава: НУШ імені Юрія Кондратюка, 2025.

Міністерство освіти і науки України  
 Національний університет «Полтавська політехніка імені Юрія Кондратюка»  
 "1 DECEMBRIE 1918" University of Alba Iulia (Румунія)  
 Білостоцький технологічний університет (Польща)  
 Вільнюський університет прикладних наук (VIKO) (Литва)  
 Університет Флорида (США)  
 London Metropolitan University (Велика Британія)  
 Словацький технологічний університет (Словаччина)  
 Гірничо-металургійна академія імені Станіслава Станіни (Польща)  
 Національний університет «Запорізька політехніка»  
 Національний університет «Дніпровська політехніка»  
 Національний університет «Одеська політехніка»  
 Ізмаїльський державний гуманітарний університет  
 Глухівський національний університет  
 Сумський національний аграрний університет  
 Національний технічний університет України  
 «Київський політехнічний інститут імені Ігоря Сікорського»  
 Харківський національний педагогічний університет імені Г.С. Сковороди  
 Чернівецький національний університет імені Юрія Фельковича  
 Національна академія сухопутних військ імені гетьмана Петра Сагайдачного  
 Київський національний університет будівництва та архітектури  
 Національний технічний університет «Харківський політехнічний інститут»  
 Одеський національний університет імені І.І. Мечникова

## МОЛОДІЖНА НАУКА: ІННОВАЦІЇ ТА ГЛОБАЛЬНІ ВИКЛИКИ

ЗБІРНИК ТЕЗ  
 II МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ  
 КОНФЕРЕНЦІЇ СТУДЕНТІВ, АСПІРАНТІВ ТА МОЛОДИХ  
 ВЧЕНИХ



Полтава, 07 листопада 2025 року

УДК 004.9:621.391

*Сухорєбрий Олександр Володимирович*

студент групи 601ТТ,

Національний університет «Полтавська політехніка імені Юрія  
Кондратюка»

### **ГЕОІНФОРМАЦІЙНІ СИСТЕМИ В ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖАХ**

Сучасний етап розвитку телекомунікаційної галузі характеризується швидким зростанням вимог до пропускної здатності мереж (FTTH, 5G) та їх складності. Традиційні методи проєктування та управління інфраструктурою, засновані на статичних кресленнях САПР та розрізних базах даних, демонструють недостатню ефективність [1]. Це визначає

стратегічну потребу в автоматизації та оптимізації процесів проектування, що вимагає впровадження інтегрованих аналітичних платформ.

Геоінформаційні системи (ГІС) виступають ключовою інтеграційною платформою, яка консолідує різноманітні дані: інженерно-технічні (специфікації обладнання), просторові (топологія мережі, адреси) та комерційні (дані абонентів) [2]. ГІС дозволяє створити єдине джерело перевірених даних для комплексного аналізу та ефективного управління мережевими активами (рис. 1).

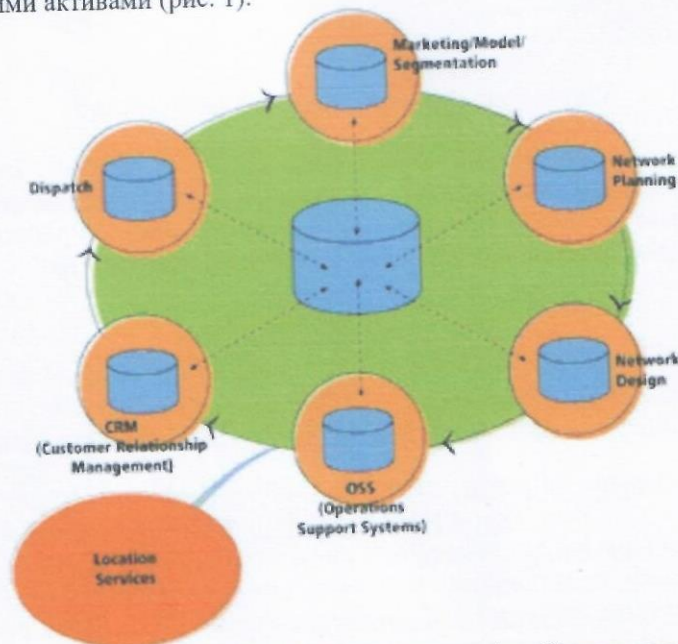


Рис. 1. Архітектура інтеграції ГІС з телекомунікаційними системами

Інтеграція ГІС забезпечує доступ до інструментів просторового аналізу, що дозволяють автоматизувати та оптимізувати планування. Наприклад, методи 2D-аналізу використовуються для наземних мереж (GPON). Просторова кластеризація (наприклад, K-Means) дозволяє автоматично аналізувати щільність абонентів та визначати оптимальні центроїди для розміщення розподільчих вузлів. Додаткове застосування прикладного аналізу (зокрема, алгоритмів «відстані до найближчого вузла») дозволяє автоматично розраховувати з'єднання та перевіряти їх відповідність технічним стандартам (рис. 2) [3].

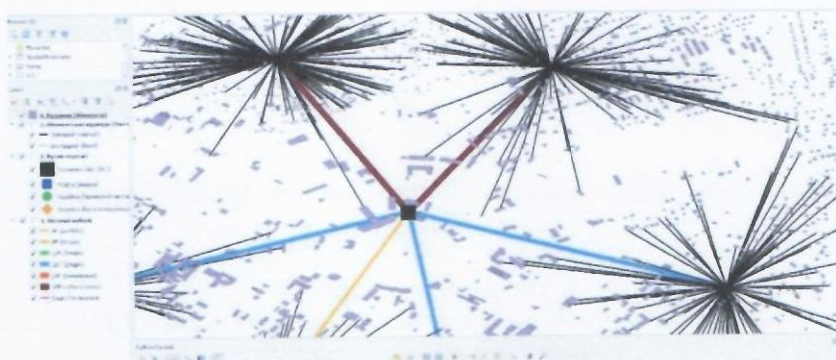


Рис. 2. 2D-аналіз GPON (кластеризація вузлів)

Отже, ГІС є фундаментальним інструментом сучасних телекомунікацій. Інтеграція методів просторового аналізу дозволяє приймати обґрунтовані інженерні рішення на всіх етапах життєвого циклу мережі – від стратегічного планування до управління операційними активами.

#### Список використаних джерел

1. van Wegen W. How geospatial data enables the rollout of fast internet. *GIM International*. 2024. URL: <https://www.gim-international.com/content/article/how-geospatial-data-enables-the-rollout-of-fast-internet>
2. IDATE. GIS Planning Tools for FTTH. 2024. URL: [https://idate.fr/wp-content/uploads/2024/05/M00103MRA\\_SAMPLE.pdf](https://idate.fr/wp-content/uploads/2024/05/M00103MRA_SAMPLE.pdf)
3. Vasdev K. GIS for 5G Network Deployment: Optimizing Coverage and Capacity with Spatial Analysis. *Journal of Artificial Intelligence & Cloud Computing*. 2022. URL: <https://onlinescientificresearch.com/articles/gis-for-5g-network-deployment-optimizing-coverage-and-capacity--with-spatial-analysis.pdf>

Наукове видання

**Молодіжна наука: інновації та глобальні виклики**

Збірник тез за матеріалами  
II Міжнародної науково-практичної конференції студентів,  
аспірантів та молодих вчених

**Редакційна колегія:**

МАСЛІЙ Олександра, ЛЕВЧЕНКО Ірина, ПАЛІЙ Катерина,  
БОРЯК Богдан, КУДІНОВА Аліна, МАКСЮТА Наталія, ЛЕВЧЕНКО  
Валерій, УСЕНКО Дмитро, РІЗНИК Юлія, ЛАКТІОНОВ Олександр

**Комп'ютерна верстка:**

КУДІНОВА Аліна, УСЕНКО Дмитро, РІЗНИК Юлія, БОРЯК Богдан,  
ЛЕВЧЕНКО Ірина

**Видавництво:**

Національний університет «Полтавська політехніка  
імені Юрія Кондратюка»  
пр. Віталія Грицаєнка, 24, м. Полтава, 36011, Україна