DOI: **10.37943/25ONQV4873**

Manara Seksembayeva
Senior lecturer, Department of Information Systems and Technologies
anuarkizi@gmail.com, orcid.org/0000-0003-0889-1058
Esil University, Kazakhstan
Zhaksylyk Amangaliyev
Bachelor student, Department of Information Systems and Technologies
akbotafort999@gmail.com, orcid.org/0009-0005-3164-7696
Esil University, Kazakhstan
Miras Anuarbekov
Bachelor student, Department of Information Systems and Technologies
anuarbekovmiras8@gmail.com, orcid.org/0009-0008-5674-9623
Esil University, Kazakhstan

INTELLIGENT DRONE ROUTE PLANNING IN URBAN ENVIRONMENTS: OPTIMIZATION AND SAFETY

Abstract: This study develops and evaluates an intelligent route-planning model for unmanned aerial vehicles operating in urban environments with the objective of minimizing total flight time while maintaining flight safety. The proposed approach addresses a practical last-mile delivery scenario in a smart-city context and is implemented using Google OR-Tools within a constrained Traveling Salesman Problem framework.

Open geospatial data from OpenStreetMap are used to obtain building geometries, street topology, and available height-related attributes, while meteorological data from OpenWeatherMap are used to account for wind conditions in flight-time estimation. Safe cruising altitude is determined from the maximum surrounding obstacle height with an additional safety margin. If explicit building-height data are unavailable, height is approximated from the number of building levels.

A prototype system was developed to visualize routes, estimate route distance, flight time, and delivery cost, and export missions in MAVLink/QGroundControl-compatible format. Experimental scenarios for real delivery points in the Astana metropolitan area, including Astana, Kosshy, and Koyandy demonstrate the practical feasibility of the proposed open-data-driven routing approach and show improvements over a baseline sequential routing strategy.

The scientific contribution of the study lies in the integration of open geospatial building data, weather-aware flight-time estimation, and safety-oriented altitude selection into a reproducible urban UAV route-planning framework. The proposed method can support the development of safe and cost-efficient drone delivery services in dense urban environments.

Experimental scenarios with increasing numbers of delivery points demonstrated that the proposed method remains computationally efficient, with sub-second solver runtime for the tested cases for practical smart-city logistics applications.

Keywords: drone delivery; route optimization; Traveling Salesman Problem; OpenStreetMap; wind conditions; building height estimation; urban route planning; Python.

Introduction

In recent years, due to the high mobility and relatively low cost of drones, drone-based delivery systems have demonstrated significant potential for providing flexible and reliable parcel transportation. However, for commercial deployment, a robust route planning mechanism is required that takes into account the following factors:

- battery energy capacity,

Received: 10.11.2025 Accepted: 25.02.2026 Published: 30.03.2026

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 25, MARCH 2026

- obstacles such as buildings and trees,
- weather conditions, including wind,
- and overall flight cost.

Most industrial solutions rely on a precomputed *horizontal* trajectory with a fixed safety altitude. In this work, we extend the classical Traveling Salesman Problem (TSP) approach by incorporating the actual building heights retrieved from OpenStreetMap (OSM) and the wind coefficient, which allows for a more accurate estimation of flight time and, consequently, delivery cost.

Route planning algorithms play a crucial role in ensuring safety, optimality, and cost efficiency.

The research objectives are as follows:

- To analyze existing methods for drone route planning;
- To develop a routing model that considers urban building heights and wind parameters;
- To implement a system prototype in Python using Google OR-Tools;
- To test the model on real delivery addresses in Astana and the adjacent urban area, including Kosshy and Koyandy, and evaluate performance metrics such as route length, estimated flight time, delivery cost, and solver runtime.

Reviews on drone applications in smart cities emphasize their contribution to urban services and quality of life [1]. Systematic surveys of last-mile delivery summarize operational efficiency and sustainability aspects [2]. Energy management and battery-related constraints are discussed in dedicated review studies [3], while practical implementation concepts for smart drone delivery systems are presented in applied engineering work [4].

Furthermore, studies [5-7] present reviews of artificial intelligence applications in logistics and autonomous delivery, focusing on the role of AI in dynamic route optimization, the use of drones and autonomous trucks in logistics chains, and the current state of implementation of these technologies in intelligent transportation systems.

In work, a parcel delivery framework is proposed as follows: after a customer places an order, a courier collects the goods and deposits them at the nearest delivery station [8]. The drone then picks up the parcel and transports it to another station, where customers can retrieve their orders by scanning a QR code with their smartphones (Figure 1).

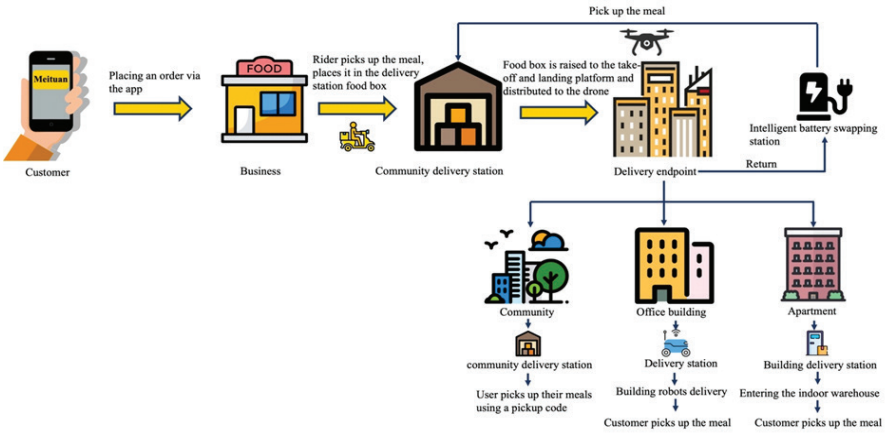

Figure 1. Parcel delivery scheme using UAVs

A multi-agent optimization method for urban air mobility (UAM) route planning considering wind and weather factors was proposed in study [9]. However, that research focuses on air-taxi management rather than multi-point logistics for drone delivery. It employs a hybrid PSO–A* algorithm but lacks integration with open geospatial data and reproducible datasets such as OpenStreetMap (OSM).

DOI: 10.37943/25ONQV4873

Study addresses the reduction of drone noise through acoustic ray tracing combined with the Deep Deterministic Policy Gradient (DDPG) algorithm [10]. Main emphasis of that work is on minimizing acoustic impact rather than optimizing flight routes in terms of time and safety.

Similarly, study focuses on reducing the acoustic footprint of UAVs in urban environments using reinforcement learning techniques (DQN/DDPG). Although highly relevant for smart-city applications, the approach does not consider logistical constraints, wind conditions, or the topology of urban structures [11].

Article presents a hybrid 3D path-planning model for drones operating in urban "canyons," employing physics-informed turbulence models and reinforcement learning algorithms [12]. However, as in previous studies, the problem is limited to single-drone trajectory generation within a simulated environment, without using open geodata, real addresses, or delivery-optimization tasks.

Unlike many existing UAV routing studies that rely on simulated environments, reinforcement learning, or specialized datasets, the proposed approach focuses on a practical last-mile urban delivery scenario based on open and reproducible data sources. The novelty of this study lies in: (1) the use of real open geospatial data from OpenStreetMap for extracting building geometries and height-related attributes; (2) safe-altitude estimation based on the maximum detected obstacle height with fallback approximation from building levels when explicit heights are unavailable; (3) the incorporation of wind conditions into flight-time estimation; (4) the formulation of the routing task as a constrained single-vehicle TSP implemented in Google OR-Tools; and (5) the development of a reproducible prototype that supports route visualization and mission export. Therefore, the contribution of the study is not limited to software implementation, but includes a unified methodology for integrating spatial, environmental, and operational constraints in urban UAV delivery planning.

Methods and Materials

For the calculation and implementation of route design, the following algorithmic steps are performed:

1. Mathematical Formulation of the UAV Routing Problem

Let the delivery system be represented by a complete directed graph

$$G = (V, E),$$

where

$$V = \{0,1,2, \dots, n\}$$

is the set of nodes, 0denotes the depot (warehouse), and $1, \dots, n$ denote customer delivery points. Each arc $(i, j) \in E$is associated with a horizontal distance d_{ij}, estimated flight time t_{ij}, and route feasibility conditions [13-14].

The binary decision variable is defined as

$$x_{ij} = \begin{cases} 1, & \text{if the UAV travels directly from node } i \text{ to node } j, \\ 0, & \text{otherwise.} \end{cases}$$

The optimization objective is to minimize the total flight time (1):

$$min \sum_{i \in V} \sum_{j \in V, j \neq i} t_{ij} x_{ij} \qquad (1)$$

subject to the following constraints.

2. Visit constraints

Each customer must be visited exactly once (2,3):

$$\sum_{j \in V,\, j \neq i} x_{ij} = 1, \forall i \in V \setminus \{0\} \qquad (2)$$

$$\sum_{i \in V,\, i \neq j} x_{ij} = 1, \forall j \in V \setminus \{0\} \qquad (3)$$

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 25, MARCH 2026

3. Depot departure/return constraints

The UAV starts from the depot and returns to the depot (4,5):

$$\sum_{j \in V \setminus \{0\}} x_{0j} = 1 \qquad (4)$$

$$\sum_{i \in V \setminus \{0\}} x_{i0} = 1 \qquad (5)$$

4. Battery/range constraint

The total route length must not exceed the maximum available flight range (6):

$$\sum_{i \in V} \sum_{j \in V,\, j \neq i} d_{ij}\, x_{ij} \leq D_{\max} \qquad (6)$$

where $D_{\max}$is the maximum route distance allowed by the UAV battery capacity.

5. Safe altitude constraint

To ensure obstacle clearance, the cruising altitude is selected as (7)

$$h_{\text{cruise}} = h_{\text{obs}}^{\max} + \Delta h \qquad (7)$$

where $h_{\text{obs}}^{\max}$is the maximum obstacle height extracted from OSM data in the routing area, and $\Delta h$is a safety margin (8). In this study,

$$\Delta h = 20 \text{ m} \qquad (8)$$

6. Distance model

The horizontal distance between two waypoints is calculated using the haversine formula (9):

$$d_{ij} = 2R \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_j - \varphi_i}{2}\right) + \cos(\varphi_i)\cos(\varphi_j)\sin^2\left(\frac{\lambda_j - \lambda_i}{2}\right)}\right) \qquad (9)$$

where $R$is the Earth's radius, $\varphi$is latitude, and $\lambda$is longitude.

7. Flight-time model

The horizontal flight time is estimated as (10)

$$t_{ij}^{\text{hor}} = \frac{d_{ij}}{v \cdot k_{\text{wind}}} \qquad (10)$$

where v is the nominal cruise speed of the UAV and $k_{\text{wind}}$is the weather correction coefficient determined from meteorological data.

The vertical ascent/descent time is estimated separately as (11)

$$t_{ij}^{\text{vert}} = \frac{|\, h_j - h_i \,|}{v_z} \qquad (11)$$

where $h_i$and $h_j$are waypoint altitudes, and $v_z$is the vertical speed of the UAV.

DOI: 10.37943/25ONQV4873

Thus, the total flight time between nodes is (12)

$$t_{ij} = t_{ij}^{\text{hor}} + t_{ij}^{\text{vert}} \qquad (12)$$

and the objective function in (1) minimizes the total mission time over the entire route [15-17].

8. *Solving the TSP with battery constraint***.**

The routing problem is formulated as a single-vehicle Traveling Salesman Problem (TSP) of the form *warehouse → all clients → warehouse*, subject to the maximum total distance D_{max}, which represents the drone's battery capacity limit. The model is implemented using Google OR-Tools**,** where the optimization objective is to minimize the total flight time or distance while satisfying the energy constraint and ensuring route feasibility within the defined safety and environmental parameters:

```
manager = RoutingIndexManager(n, 1, 0)
routing = RoutingModel(manager)
# Стоимость = время
tidx = routing.RegisterTransitCallback(lambda i,j: int(t_mat[...] *1000))
routing.SetArcCostEvaluatorOfAllVehicles(tidx)
# Ограничение по дистанции
didx = routing.RegisterTransitCallback(lambda i,j: int(d_mat[...] *1000))
routing.AddDimension(didx, 0, int(D_max*1000), True, "Distance")
```

Google's OR-Tools are an open-source suite designed for optimization, providing a plethora of algorithms for routing, flow, graph, linear programming, and constraint problems. A detailed overview of various local search and meta-heuristic methods implemented in OR-Tools is presented in [18].

9. Workflow of the Proposed Method

The proposed route-planning framework consists of the following sequential stages:

- **Geocoding of input locations.** The warehouse and customer addresses are converted into geographic coordinates using Nominatim/OpenStreetMap services. The resulting coordinates form the node set of the routing graph.
- **Extraction of urban obstacle data.** A bounding box enclosing all route points is generated. Building data are retrieved from OpenStreetMap through Overpass API queries. For each building, the tags height and building:levels are analyzed. If the height tag is unavailable, building height is approximated as (13):

$$h_b = 3L, \qquad (13)$$

where L is the number of building levels and 3 m is the assumed average floor height.

- **Determination of safe cruising altitude.** The maximum obstacle height within the area is identified, and the cruising altitude is computed according to Eq. (7), ensuring a safety clearance above surrounding buildings.
- **Computation of pairwise distance and time matrices.** The pairwise distance matrix is constructed using the haversine formula. The time matrix is then calculated by incorporating horizontal movement, vertical ascent/descent, and wind correction.
- **Optimization of the route.** The routing task is solved as a constrained single-vehicle TSP using Google OR-Tools. The objective is to minimize the total estimated flight time while satisfying the battery-range constraint.
- **Visualization and export.** The optimized route is displayed on an interactive map, together with total route length, estimated flight time, altitude, and delivery cost. The route can additionally be exported in MAVLink/QGroundControl-compatible format for mission execution.

10.Computational Complexity Analysis

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 25, MARCH 2026

The proposed routing problem belongs to the class of Traveling Salesman Problem (TSP) and Vehicle Routing Problem (VRP), which are NP-hard optimization problems. Therefore, the exact computational complexity grows combinatorially with the number of delivery points and, in the worst case, can be expressed as (14)

$$O(n!), \qquad (14)$$

for exhaustive search over all possible tours.

In practice, exact enumeration is computationally infeasible even for medium-sized urban delivery instances. For this reason, the present study uses Google OR-Tools, which applies efficient heuristic and metaheuristic search strategies such as local search, guided local search, and first-solution strategies. These methods do not guarantee a globally optimal solution in all cases, but they provide near-optimal routes within practical computation time.

The computational burden of the proposed system consists of two main components: (1) preprocessing time for geocoding, weather requests, and building-data extraction; and (2) optimization time for solving the routing model. As the number of delivery points increases, both the size of the distance/time matrices and the routing search space grow rapidly. However, for practical urban last-mile scenarios involving a moderate number of customers, the method remains computationally tractable and suitable for interactive use.

The proposed system operates in an interactive mode oriented toward operators or end users who have no prior UAV piloting skills. The input of initial data can be performed through two complementary methods (Figure 2):

1. *Text-based address entry with autocomplete*. When the user begins typing an address, the front-end initiates a request to the geocoding service and displays relevant suggestions. Once the user selects an option, it is converted into geographic coordinates (latitude and longitude) in the WGS-84 system.

2. *Spatial input via the map interface (point selection)*. The user can click on the map or drag a marker to the desired location (for instance, a courtyard or building entrance). This functionality allows the user to refine the exact delivery point and recalculate the route if necessary.

The marker's position serves as the *ground truth* for the planner. After the marker is moved, reverse geocoding is automatically performed — the address field is synchronized and filled with the nearest postal address or description.

Both input methods are available for the warehouse and for an unlimited number of client addresses; sequential addition of delivery points is supported. When the **"Generate Route"** button is pressed, the system constructs an optimal route and displays aggregated performance indicators, including total distance, estimated flight time, and approximate delivery cost.

The technical implementation in the source code is organized as follows: a server-side Flask endpoint /api/plan receives input data in JSON format of the following structure:

```
{
  "warehouse_lat": 51.15, "warehouse_lon": 71.47,
  "customers": [{"lat": 51.14, "lon": 71.47}, ...],
  "battery_km": 30, "departure": "2025-07-30T12:00",
  "flight_alt": 60
}
```

and returns the optimal route, node coordinates, and aggregated indicators.

The generated delivery missions are stored in an SQLite database together with route metrics and timestamps, which allows experiment reproducibility and later analysis of routing scenarios.

The system uses OpenStreetMap (OSM) as the address reference source via the public Nominatim geocoder [19, 20]:

• Forward geocoding (string → coordinates) — for autocomplete and address confirmation;

DOI: 10.37943/25ONQV4873

• Reverse geocoding (coordinates → string) — for automatic text field updates when a marker is moved.

Building data: Building geometries and available height-related attributes were retrieved from OpenStreetMap (OSM) via Overpass API / OSMnx. For each building, the tags height and building levels were analyzed. When explicit height values were unavailable, building height was approximated from the number of levels using $h_b = 3L$, where $L$is the number of floors and 3 m is the assumed average floor height.

Weather data: a wind coefficient is obtained from the *OpenWeatherMap* service; if unavailable, a neutral multiplier of 1.0 is applied.

This choice ensures reproducibility and sufficient spatial coverage across Kazakhstan. For industrial deployment, the public Nominatim instance can be replaced with a self-hosted or corporate geocoder, and additional data sources (e.g., national address registries) can be integrated through a caching layer to improve reliability and reduce API request limits.

System robustness and error handling:

• If no address is found or the geocoder returns an ambiguous result, the user can manually specify the position by moving the marker — these coordinates are accepted unconditionally.

• If the geocoding service becomes unavailable or API limits are exceeded, the system continues operating with the already provided coordinates, caches responses, and suggests retrying later.

• If height tags are missing in OSM, a user-defined minimum safe altitude with a fixed safety margin is applied.

• The public Nominatim service is rate-limited; for production environments, self-hosted deployment or a corporate geocoding API is recommended.

• OSM address coverage may be incomplete in some urban areas; therefore, a manual refinement mode (dragging the marker) is provided as the primary accuracy improvement method.

• The prototype can be extended to incorporate restricted flight zones such as airports and protected government areas through additional spatial filtering layers.

Results

The experimental results for different routing scenarios are summarized in Table 1. It should be noted that the obtained route distance, flight time, and delivery cost depend not only on the number of delivery points, but also on their spatial distribution within the urban area.

The experimental scenarios were constructed using real delivery locations in Astana and the surrounding urban area, including Kosshy and Koyandy.

Table 1. Quantitative performance results for different UAV routing scenarios

Scenario	Number of delivery points	Route distance (km)	Estimated flight time (min)	Solver runtime (s)	Delivery cost (KZT)
S1	3	24.36	45.7	0.005	21361
S2	5	24.70	47.6	0.008	21 688
S3	7	66.38	122.2	0.053	57 270
S4	10	103.33	189.1	0.689	88 836

The results presented in Table 1 indicate that the proposed model remains computationally efficient across all tested scenarios. Although route distance, estimated flight time, and delivery cost generally increase with the number of delivery points, their exact values are also determined by the spatial arrangement of the selected destinations.

To assess the practical benefit of the proposed method, the optimized routes were compared with a baseline sequential routing strategy, in which delivery points were visited in the same order as entered by the user. The comparison results are summarized in Table 2.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 25, MARCH 2026

Table 2. Comparison of the proposed OR-Tools optimization method with baseline sequential routing

Method	Number of delivery points	Route distance (km)	Estimated flight time (min)	Delivery cost (KZT)
Baseline sequential order	3	29.50	54.7	25 740
Proposed OR-Tools optimization	3	24.36	45.7	21361
Baseline sequential order	5	30.45	57.7	26565
Proposed OR-Tools optimization	5	24.70	47.6	21 688

As shown in Table 2, the proposed OR-Tools-based optimization method reduced the total route distance, estimated flight time, and delivery cost compared with the baseline sequential strategy.

An example of the optimized route generated by the system is illustrated in Figure 2.

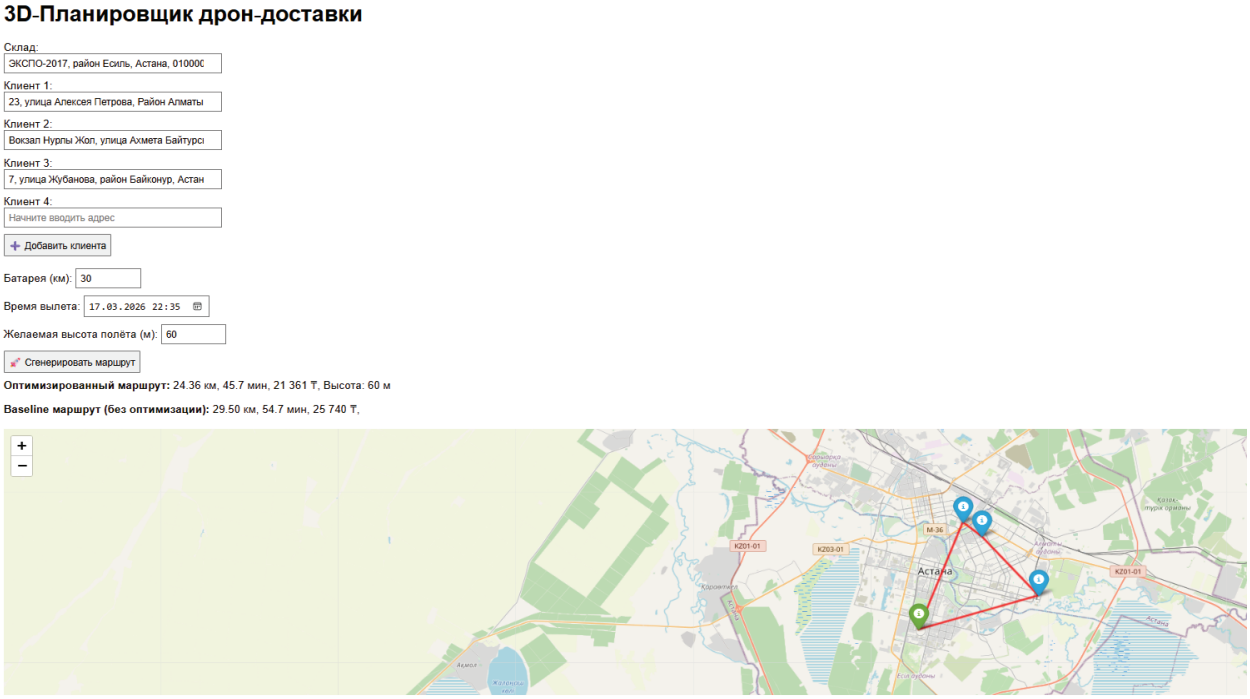

Figure 2. Example of optimized UAV delivery route generated by the proposed system

As shown in Table 1, the total route distance, flight time, and delivery cost generally increase with the number of delivery points; however, these values also depend on the geographic configuration of the selected delivery locations.

The system performs the following operations:

- Geocodes all input points into WGS-84 coordinates;
- Retrieves building information from OpenStreetMap (OSM) along the route to estimate the safe flight altitude;
- Applies a weather adjustment coefficient (wind factor) obtained from *OpenWeatherMap* when calculating flight time;
- Solves the routing problem (a variant of TSP/VRP with a “battery range” constraint) using *Google OR-Tools*;

- Visualizes the resulting optimal route on a web map and outputs a metrics table (distance, time, cost);
- Accounts for the vertical clearance between the cruising altitude and the maximum building height.

Discussion

The results obtained show that the proposed route-planning model is computationally efficient for small and medium-sized urban delivery scenarios. Even when the number of delivery points increased from 3 to 10, the optimization time remained below one second, which indicates the practical suitability of the approach for interactive route generation.

At the same time, the results demonstrate that route distance, estimated flight time, and delivery cost depend not only on the number of delivery points, but also on their geographic configuration. Therefore, these indicators do not increase strictly linearly and are influenced by the spatial distribution of delivery destinations within the urban environment.

The comparison with the baseline sequential routing strategy confirms the practical value of the proposed optimization approach. For the tested scenarios, the OR-Tools-based solution produced shorter routes and lower estimated flight times and delivery costs than the non-optimized baseline. This indicates that algorithmic route optimization is preferable to simple input-order delivery planning.

The results also confirm the feasibility of combining open geospatial and meteorological data in an urban UAV routing workflow. The use of building data from OpenStreetMap and weather-based correction factors improves the realism of route estimation compared with purely geometric routing models.

However, the current prototype has several limitations. Public geocoding services may be rate-limited, building-height attributes in OpenStreetMap may be incomplete, and the experiments were limited to a relatively small set of delivery scenarios in the Astana metropolitan area. In addition, restricted flight zones are not yet fully integrated into the optimization model.

Future research should focus on integrating no-fly-zone layers directly into route feasibility constraints, extending the model to dynamic weather conditions, and evaluating the method on larger routing scenarios and multi-UAV delivery tasks.

Conclusion

The study formally defined the UAV route-planning task as a constrained TSP/VRP optimization problem and experimentally evaluated the proposed model across multiple urban delivery scenarios in Astana and the surrounding metropolitan area, including Kosshy and Koyandy. The proposed approach integrates open geospatial data from OpenStreetMap, meteorological information, and OR-Tools-based optimization to generate safe and cost-efficient drone delivery routes. The obtained results confirmed both the practical feasibility of the approach and its computational efficiency for small and medium-sized urban delivery tasks.

Advantages:

- Automatic consideration of building heights reduces the risk of collision.
- The wind correction factor improves the accuracy of flight-time and cost estimation.
- The proposed implementation is reproducible, flexible, and easily extensible for further research and practical applications.

Limitations:

- The Overpass API has request rate and latency limitations.
- OSM data may be incomplete (missing height/levels tags for some buildings).
- Static planning does not account for unexpected obstacles in real time.

Future improvements:

- Implement caching or a local OSM layer to reduce latency.
- Utilize DSM LiDAR data for precise elevation modeling.
- Integrate onboard computer vision (CV) modules for obstacle avoidance and route updating in real time.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 25, MARCH 2026

References

[1] Důbravová, H., Bureš, V., & Velfl, L. (2024). Review of the application of drones for smart cities. IET Smart Cities, 6(4), 312–332. https://doi.org/10.1049/smc2.12093

[2] Garg, V., Niranjan, S., Prybutok, V., Pohlen, T., & Gligor, D. (2023). Drones in last-mile delivery: A systematic review on efficiency, accessibility and sustainability. Transportation Research Part D: Transport and Environment, 123, 103831. https://doi.org/10.1016/j.trd.2023.103831

[3] Rajabi, M. S., Beigi, P., & Aghakhani, S. (2023). Drone delivery systems and energy management: A review and future trends. In Handbook of Smart Energy Systems (pp. 1273–1291). https://doi.org/10.1007/978-3-030-97940-9_196

[4] Magdum, A., Nikam-Patil, V., Mokashi, V., & Waykule, J. (2020). Smart drone delivery system. International Journal of Scientific Research in Engineering and Management (IJSREM), 4(3). ISSN 2582-3930.

[5] Adeoye, Y., Onotole, E. F., Ogunyankinnu, T., Aipoh, G., Osunkanmi, A. A., & Egbemhenghe, J. (2025). Artificial intelligence in logistics and distribution: The function of AI in dynamic route planning for transportation, including self-driving trucks and drone delivery systems. World Journal of Advanced Research and Reviews, 25(2), 155–167. https://doi.org/10.30574/wjarr.2025.25.2.0214

[6] Kim, B.-H. (2018). Implementation of the centralized control system for drone training. International Journal of Engineering and Technology, 7(3.24), 595–599. https://doi.org/10.14419/ijet.v7i2.33.14190

[7] Raivi, A. M., Huda, S. M. A., Alam, M. M., & Moh, S. (2023). Drone routing for drone-based delivery systems: A review of trajectory planning, charging, and security. Sensors, 23(3), 1463. https://doi.org/10.3390/s23031463

[8] Li, J. (2024). The use of AI technology in drone delivery [Online]. AIFT – Laboratory for AI-Powered Financial Technologies, May 14, 2024. Available: https://hkaift.com/the-use-of-ai-technology-in-drone-delivery/ (accessed: 12.09.2025).

[9] Daud, S. M. S. M., Yusof, M. Y. P. M., Heo, C. C., Khoo, L. S., Singh, M. K. C., Mahmood, M. S., & Nawawi, H. (2022). Applications of drone in disaster management: A scoping review. *Science & Justice, 62*(1), 30–42. https://doi.org/10.1016/j.scijus.2021.11.002

[10] Rinaldi, M., Primatesta, S., & Guglieri, G. (2025). Low-noise path planning for urban drone missions: Acoustic ray tracing and DDPG algorithm. The Aeronautical Journal, 129(1320), 1–23. https://doi.org/10.1017/aer.2025.10054

[11] Sarhan, S., Rinaldi, M., Primatesta, S., & Guglieri, G. (2025). Noise-aware UAV path planning in urban environment with reinforcement learning. Engineering Proceedings, 90(1), 3. https://doi.org/10.3390/engproc2025090003

[12] Rienecker, H., Hildebrand, V., & Pfifer, H. (2023). Energy optimal 3D flight path planning for unmanned aerial vehicle in urban environments. CEAS Aeronautical Journal, 14, 621–636. https://doi.org/10.1007/s13272-023-00666-x

[13] Choudhury, S., Solovey, K., Kochenderfer, M. J., & Pavone, M. (2020, June 11). *Efficient large-scale multi-drone delivery using transit networks* (arXiv:1909.11840v4) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1909.11840

[14] Fu, Y., An, W., Su, X., & Song, B. (2025). Real-Time Wind Estimation for Fixed-Wing UAVs. *Drones, 9*(8), 563. https://doi.org/10.3390/drones9080563

[15] Sziroczak, D., Rohacs, D., & Rohacs, J. (2022). Review of using small UAV-based meteorological measurements for road weather management. *Progress in Aerospace Sciences, 134*, 100859. https://doi.org/10.1016/j.paerosci.2022.100859

[16] Adkins, K. A., Becker, W., Ayyalasomayajula, S., Lavenstein, S., Vlachou, K., Miller, D., Compere, M., Krishnan, A. M., & Macchiarella, N. (2023). Hyper-local weather predictions with the enhanced general urban area microclimate predictions tool. *Drones, 7*, 428. https://doi.org/10.3390/drones7070428

[17] Lee, S., Kim, E., & Baik, H. (2024). Path planning for urban air mobility considering weather conditions. In Proceedings of the 2024 AIAA DATC / IEEE 43rd Digital Avionics Systems Conference (DASC), September 29, 2024, pp. 1–6. https://doi.org/10.1109/DASC62030.2024.10749551

DOI: 10.37943/25ONQV4873

[18] Benoit, A., & Asef, P. (2025, March 5). *Navigating intelligence: A survey of Google OR-Tools and machine learning for global path planning in autonomous vehicles* (arXiv Preprint arXiv:2503.03338). https://doi.org/10.48550/arXiv.2503.03338

[19] Tenkanen, H., Heikinheimo, V., & Whipp, D. (2020–2025). *Retrieving OpenStreetMap data.* In *Python for Geographic Data Analysis* (Part II, Chapter 9). Retrieved from https://pythongis.org/part2/chapter-09/nb/00-retrieving-osm-data.html

[20] Kang, L., Wang, Q., & Yan, H. W. (2018). *Building extraction based on OpenStreetMap tags and very high spatial resolution image in urban area.* The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-3, 715-718. https://doi.org/10.5194/isprs-archives-XLII-3-715-2018