

Optimization and Representation of a Network of Food Delivery Drones in Simulation

Dávid Kelemen

John von Neumann Faculty of Informatics, Óbuda University, Hungary
davidkelemen74@stud.uni-obuda.hu

Sándor Szénási

John von Neumann Faculty of Informatics, Óbuda University, Hungary
Faculty of Economics and Informatics, J. Selye University, Slovakia
szenasi.sandor@nik.uni-obuda.hu, szenasis@uj.sk

Abstract—Couriers of food delivery services are playing a significant role in urban areas by efficiently delivering packages. However, the problem of coordinating many couriers cannot be easily managed. Self-driving vehicles are likely to easily reduce courier companies' logistical costs soon. This paper explores how a multi-node drone network can be centrally coordinated to deliver orders quickly, aiming to achieve customer satisfaction. Since the number of entities and calculations outlined in the problem is significantly large, the calculations require the help of one of the optimization solutions, which is guided by a genetic algorithm for route planning. This paper aims to create a simulation that uses the heuristic optimization algorithm to solve this issue. The goal of the simulation is to consider as many factors as possible to generate, calculate, and visualize the environment. Simulated courier drones must work stably under deterministic and stochastic conditions. The data provided by the simulation can serve as an estimate of how a realistic or hypothetical resource system would operate.

Keywords—drone, delivery, optimization, GA, simulation, VRP

I. INTRODUCTION

The rapid growth of food and package delivery services of specific companies has significantly influenced urban environments. These companies have been striving to efficiently coordinate the delivery of packages and meals to residents, utilizing various modes of transportation, including cars, motorcycles, and bicycles. However, the coordination of many couriers presents a complex problem due to the random nature of their behavior and the dynamic nature of traffic conditions.

Researchers have explored alternative solutions to this challenge, such as using autonomous vehicles and drones. Autonomous vehicles can reduce logistical costs for delivery companies in the near future. Drones, in particular, offer unique and alternative solutions for task execution. Although drones may not be capable of delivering large quantities, their ability to bypass existing urban infrastructure limitations allows for swift delivery, making them a key player in food delivery.

However, the implementation of drone-based food delivery faces several obstacles. Many urban areas have restricted or no-fly zones that limit their access. Additionally, the absence of gardens or open spaces in some households poses challenges regarding delivery drop-off. Weather conditions also impact drone operations, as they cannot operate reliably and safely under strong winds or heavy precipitation. This research assumes the optimal conditions for drone operations are met and focuses on addressing the central question of centrally coordinating a multi-node drone network for efficient delivery while prioritizing customer satisfaction and time efficiency.

This scientific study adopts a heuristic search approach to provide quantitative data based on realistic conditions and input parameters. The research methodology involves dividing the problem into smaller components and creating a

mathematical model representing the behavior of the system. By running simulations, the study can offer insights into potential outcomes and estimates that would be difficult, expensive, or time-consuming to obtain in real-world scenarios.

The simulation aims to create a model incorporating various factors to ensure stable operations of centrally controlled drones under deterministic and stochastic conditions. The developed algorithm operates based on system-specific requirements and constraints. The simulation data can provide valuable insights into implementing such a drone delivery system's feasibility, environmental impact, and cost-effectiveness. It can assist in determining the appropriate types of drones, identifying suitable environments for the system, and optimizing the placement of service stations for drone maintenance, charging, and collection to minimize costs.

In conclusion, this research focuses on a dynamic variant of the Vehicle Routing Problem (VRP) called Dynamic Vehicle Routing Problem (DVRP) within the context of coordinating drone networks. While existing VRP solutions address static delivery routes with known locations, this study explores the dynamic nature of delivery orders. The simulation-based approach, coupled with the heuristic optimization algorithm, provides a valuable tool for assessing the performance and potential of a realistic drone delivery system.

II. RELATED WORK

Simple scheduling strategies like *First Come First Served*, *Shortest Job First*, or *Nearest Neighbor* may be useable when a delivery system needs management [1,2]. Although, to achieve the best possible, the least expensive, or the fastest solution at the most decision point, it is worth trying other complex, more sophisticated algorithms.

In addition to common strategies, it is possible to refer to solutions as "classical insertion procedures" that re-optimize the route to be traversed and serviced after the arrival of new data or new demands. Such solutions utilize the "rolling horizon" principle, which periodically examines and considers new demands and requests at certain intervals. [2]

Metaheuristics can also provide a solution, as they are able to provide a large best or approximate value for a combinatorial problem with high computational requirements approximate. Similar to the optimization solution in this paper, the *Decision Support for Vehicle Dispatching Using Genetic Programming* research is also based on evolutionary algorithms. The difference is that genetic programming was used by the authors, and the order assignment is sequential. [3]

MIP, i.e., 'mixed integer programming' based heuristic approach is presented in *Dynamic Dispatch* algorithm. MIP is a category of 'integer programming,' where variables are not only discrete variables. Its use may occur in transportation scheduling; the binary decision variables can indicate the

assignments to vehicle routes. The author of *Dynamic Dispatch* works with a time window that optimizes the system state based on demand. [4]

Another work utilizes the backpropagation algorithm of neural networks to train a simulated dispatcher to make good decisions. The input vectors of the neural network represent the description of each courier's assigned location. [5]

A study like this paper uses a genetic algorithm for calculations between couriers and delivery/pick-up points. This study distributes both static and dynamic order demands among the traveling couriers in this way. However, the solution outlined in the study, differing from the implementation described in this paper, sequentially assigns the orders between moving points in the dynamic phase. [6]

A similar simulation to the one outlined here was developed in the work titled *Profit Driven Drone Scheduling in Last Mile Food Delivery Networks*. This study demonstrated the operation of multiple scheduling procedures (FCFS, SJF, PS, LLV) in a network of autonomous drones across multiple waypoints, considering different scenarios. The thesis simulated no-fly zone areas, so one of the tasks for the drones was to avoid them. The research has a time function that assigns a value to the order according to how long it has (not) been delivered. [1]

III. METHODOLOGY

This paper focuses on addressing the scheduling and order assignment problem dynamically and continuously. It introduces mobile drone entities, also known as couriers, that deliver orders from restaurants to designated locations of customers in a simulated urban environment. The orders are randomly generated and treated as movable objects. The restaurants are static entities with fixed locations. A key aspect of the study is simulating realistic drone behavior, including the sensitivity of the scheduling algorithm to changes in drone battery levels. Charging stations are created as static objects where battery replacements can take place. The paper avoids unnecessary details unrelated to the problem and focuses on the relevant models and system components.

A. Environment and model

The simulation environment, with a predefined number of objects and their properties, is generated by the executing environment, allowing for the study of various scenarios and behaviors in a 2D representation while considering the vertical movements of drones based on their maximum ascending and descending speeds, with the simulation speed adjustable before program execution.

The paper utilizes the properties of an existing commercially available drone model from a reliable manufacturer instead of a specifically designed drone by a large company, aiming to create a realistic and general representation of current drone capabilities, with access to all necessary attributes publicly disclosed by the manufacturer. This model is DJI's Matrice 300 RTK. [7]

The energy consumption must be calculated continuously, and it is calculated by this formula:

$$E_{pm} = \frac{g \sum_{k=1}^3 m_k}{r\eta} + \frac{P_a}{v_{air}} \quad (1)$$

where P_a is the power required for all avionics on the drone [Watt = J/s], v_{air} is airspeed [m/s], g is the acceleration of gravity [m/s^2], m_k is the mass of drone component k [kg], r is lift-to-drag ratio [unitless], and η is battery and motor power transfer efficiency (from battery to propeller) [unitless]. Therefore, E_{pm} returns the energy required for steady drone flight per unit distance [J/m]. [8]

The simulated drones fly at 100 meters and travel at their maximum speed. The primary safety rule for a drone is that it must not fall below a certain energy level. If it does fall below the established level, it will switch to emergency mode and head to the nearest charging station. According to the specifications, the drone can carry a maximum weight of 2.7kg, rejecting heavier orders, and the simulation limits the maximum number of deliveries to 2, with no option for transferring a delivery to another drone once assigned. The states that the drones can pick up are *airborne*, *charging*, *transfer*, and *at-rest* states, with a total of 7 such states distinguished in the simulation.

The restaurants in the model have a limited role compared to the drone model. They "prepare" orders at their coordinates. There is an infinite number of available landing slots for drones to execute order pickups. While the drones handle loading the food into their delivery units, they are unaffected by other drones' movements or states.

The restaurant will handle multiple orders, each with its own order ID and weight, which are necessary data for the program's operation. The weight of the orders will indicate the combined weight of the ordered items, and the program will assign random food, drink, and dessert to the orders. The generation of orders will be based on Poisson random values, and it considers the population density and size of the areas of the used map of Budapest. No orders are generated for unpopulated areas. Similar to the drones, the orders also go through different states with one-way transitions: they start in the *preparation* phase, then become *ready*, followed by being *in delivery*, and finally, their state changes to *delivered*.

A charging station serves as a landing zone for drones to start their workday in the simulation and replace their batteries if needed, with the capacity to accommodate and charge all drones simultaneously. It would be worth considering problems and solutions for optimizing the finite number of batteries. When drones enter a charging station, they automatically receive a fully charged "new" battery instead of starting to charge themselves. The battery swap takes 40 seconds, and multiple charging stations can be created and tested to identify the most efficient operation. In the presented general scenario, the location of the charging stations were the fairly densely populated, hot spots of Budapest.

B. Precalculations

As mentioned in the introduction, a genetic algorithm will perform system optimization. To provide appropriate data for the GA to optimize, preliminary calculations are necessary to assess the state and position of various simulation elements. The program performs mathematical calculations/formulas in the following order:

- To determine the possible route allocation between drones and active, new orders, it's important to identify the objects suitable for

route allocation. This survey is performed by the *Dispatcher* object, which has knowledge of the state of drones and elements of the order queue. The essence of this step is to establish an order among the *completed-by-restaurant* or *in-progress* orders based on their given importance over time.

- Next, it is necessary to examine which drone-order pairings are infeasible and then initiate the examination of the fulfillment conditions for the feasible ones.
- The third step is calculating the energy loss for a drone and the time required for delivering an order based on the given information. Here the program first checks how many tasks are assigned to the drone, and if there are already two orders assigned, it does not start the calculation because it's redundant. The formula used here distinguishes three cases that determine the type of route for the drone. The analysis of the algorithm does not consider the duration and energy loss of take-off and landing, as they are represented by constant time and energy values, which are calculated within the program. The three cases of route types are the following:
 1. *Solo*: The courier does not have any active tasks. If it receives a task, it can start fulfilling it immediately.
 2. *Comeback*: In this case, the examined drone has an active task that needs to be prioritized for delivery. Assuming the feasibility, the algorithm used here determines the estimated time required to deliver the active order; and then returns and delivers the calculated new order also, which is the input value of the function.
 3. *Together*: The courier's active task is to deliver an order prepared in the same restaurant as the input order, which also needs a courier to deliver. In this case, the program evaluates the time and energy cost associated with the drone picking up multiple orders simultaneously. Certainly, if a route is feasible according to both *Comeback* and *Together* types, the program will determine which one is more advantageous in terms of time and provide that as the output.
- After gathering the data on possible drone-order routes, it is forwarded to the optimization module.

C. Optimization

The problem described in the paper is stochastic in nature and requires a dynamic approach. Unlike hill-climbing algorithms, the genetic algorithm can effectively handle combinatorial problems and always converge towards the global optimum, even in the presence of local optima. The implementation discussed in the paper focuses on single-

criterion optimization, considering only the shortest delivery time.

The variables of the algorithm are called individuals. These individuals form the set of the algorithm, which is referred to as the population. In this set, the individuals compete for survival. This survival, which represents the speed and strength of a cell, is ensured in the implementation by the effectiveness of the individual's solution to the given task. The variation of the problem is represented by the individual's own collection or sequence of chromosomes in the solution. It is through these chromosomes that the state and "goodness", the fitness of the individuals can be described.

In the presented solution, the role of individuals is fulfilled by arrays that contain the drone-order pairings. The elements of these arrays indicate how long it takes for the drone to reach the restaurant for each examined route during the pre-calculation phase and the time it takes to deliver the respective order afterward. These arrays serve as a representation of the individuals in the optimization process.

Table I. Example for an individual

Number of pairing	Drone ID	Order ID	Delivery time (s)	Pickup time (s)	Fitness sum
1	4	1	216	400	616
2	6	2	270	178	448
3	2	3	542	255	797

The fitness function is defined as follows:

$$f(x) = \sum_{i=1}^n t_{delivery}^{p_i} + t_{pickup}^{p_i} \quad (2)$$

where i is the index of the pairing, n is the number of pairings, p is the pairing itself, $t_{delivery}$ is the delivery time from the restaurant to the address of the customer, t_{pickup} is the time for the drone to arrive to the restaurant for pickup and x is the array or individual.

The initial step in the program is to generate random pairings for the distributions. Then, fitness evaluation takes place, followed by elitism, where a portion of the best-value distributions is saved in an array. During crossover and mutation, where the program swaps drone IDs between individuals, ensuring that no repetitions occur was necessary because a courier can only be assigned one task at a time in a single computational round. Therefore, precautions were taken to avoid assigning the same task to a courier more than once.

IV. EVALUATION



Fig. 1. Snapshot from the simulation

Quoting from a research discussing VRP solutions: "Strikingly, no standard problem definitions or formulations are available for dynamic VRPs. Additionally, to the authors' knowledge, no benchmark instances are available to test and compare the proposed solution methods objectively." [9]

Regarding simulations, the processes created with the program will only partially depict reality. This is influenced by the presence of random events, which can occur in various ways in real life. The model presented here can be examined in operation to answer whether it functions as expected.

Table II. Settings of the genetic algorithm

Number of individuals	300
Number of generations	500
Crossover rate	40%
Mutation rate	15%
Elitism rate	5%

Table III. Runtimes of the genetic algorithm with different settings (s)

		Number of orders						
		5	7	10	15	30	50	100
Number of drones	10	0.112	0.138	0.300	0.300	0.300	0.310	0.300
	20	0.105	0.127	0.152	0.247	0.847	0.830	0.840
	40	0.106	0.120	0.145	0.188	0.437	2.950	2.950
	80	0.109	0.118	0.146	0.176	0.306	0.590	12.866
	120	0.106	0.122	0.140	0.179	0.299	0.515	2.688
	150	0.110	0.122	0.140	0.179	0.292	0.490	1.582
	200	0.107	0.122	0.146	0.176	0.295	0.497	1.282

The system only considers maximum as many orders in the optimization phase as the number of drones used in that particular run, which can lead to stagnation in the data. Irregularity can be seen in the data marked in yellow, where there is a jump compared to the previous regular pattern. This is because as the program tries to optimize the number of orders closer to the number of drones, the system needs to search for more exchanges during the special crossover and mutation processes.

The simulation was run with different settings to examine the program's functionality thoroughly. One run took 6 hours, simulating the operations of a 6-hour urban restaurant network. The data was saved by the system in a textual format

during or immediately after the program's execution, depending on the data type.

Table IV. Values involved in the simulation

Simulated handing time at customer and at restaurant	30 seconds
Battery swap time	40 seconds
Preparation time of orders	10-20 minutes (randomized)
Number of restaurants	4
Number of charging stations	4
Generated orders	~1600
Weight of an order	830-1585 gram (randomized)

Table V. Workload of couriers, measured at the end of simulation run

Drone ID	Number of chargings	Average measured battery level at charging station (V)
1	11	3.495
2	10	3.530
3	12	3.631
4	12	3.619
5	11	3.528
6	12	3.577
7	10	3.429

The couriers were occupied in a similar manner, their average remaining energy level before battery replacement ranged from 3.4V - 3.6V.

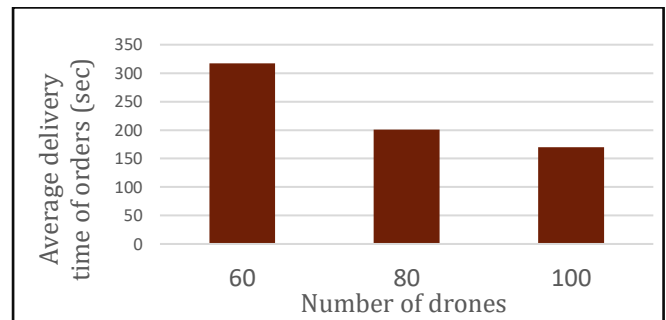


Fig. 2. Average delivery times with different drone fleets

The expected results align with the system's ability to achieve shorter average delivery times with multiple couriers and find the most suitable courier for each order more easily. With such a level of order frequency, the couriers were able to maintain delivery times of approximately 5 minutes.

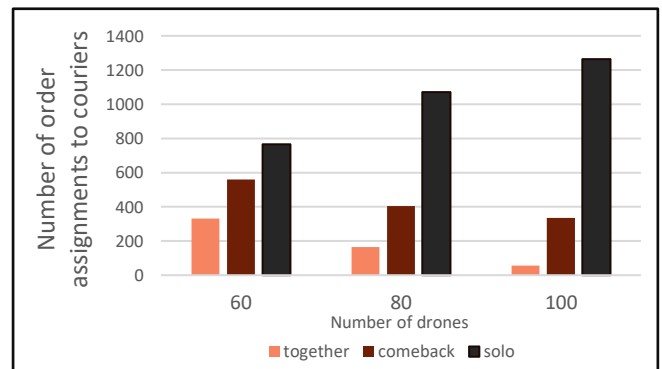


Fig. 3. Distribution of route types with different drone fleets

In changing the route types, it was clear that the more drones were available, the more the system favored the allocation of single-order routes, which were costly in energy but well worth the time, over more complex routes involving the delivery of multiple orders.

V. CONCLUSIONS

The research had a dual purpose: to create a simulation that can be used to model food delivery using aerial vehicles and to develop a suitable algorithm to optimize the network's operation. Both objectives were successfully achieved, and the model behaved exactly as expected during the tests. With the simulation, it was possible to perform a run where orders were continuously generated and drones allocated to destinations, thus realizing the collaboration of up to 100 drones and the simultaneous delivery of over 1500 orders.

Orders were generated locally weighted to represent the behavior of an urban environment. The operation of the simulation models was also designed to depict reality as accurately as possible. The research paid great attention to incorporating the energy consumption of drones as a significant factor in the model, thus establishing the physical aspect of the model. Order allocations were prepared using a mathematical model based on the presented route types. The new orders are processed by the system in sets, and the optimization is performed using a genetic algorithm [10]. Based on the tests, the optimized assignments of orders to drones effectively minimized the delivery time as expected, although the achieved speed may seem idealistic. To accomplish this, a simulation has been developed to generate results and visually represent the real-time appearance of orders and the functioning of the models [11,12,13].

During the setup of the system, a set of rules was created, including constraints that do not provide the full range of situational possibilities required by a food delivery network. In the simulation, delivery drones can handle a maximum of two orders simultaneously, and the system assigns only one order to a drone during a distribution phase. This demonstrates that the presented optimization solution works, but it may be necessary to expand the rule set and broaden the operational scope in the future.

ACKNOWLEDGMENTS

The authors thank the High Performance Computing Research Group of Óbuda University for its valuable support. The authors also thank NVIDIA Corporation for providing graphics hardware for the experiments.

REFERENCES

- [1] A. Percy, *Profit Driven Scheduling in Last Mile Food Delivery Networks*, Imperial College London, 2019
- [2] G. Ghiani, F. Guerriero, G. Laporte, R. Musmanno, Real-time vehicle routing: solution concepts, algorithms and parallel computing strategies, *European Journal of Operational Research*, Vol. 151, 2003, pp. 1-11
- [3] I. Benyahia, J.-Y. Potvin, Decision Support for Vehicle Dispatching Using Genetic Programming, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 28., 1998, pp. 306-314
- [4] L. Yanchao, An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones, *Computers and Operations Research*, Vol 111., 2019, pp. 1-20
- [5] Y. Shen, J.-Y. Potvin, J.-M. Rousseau, S. Roy, A computer assistant for vehicle dispatching with learning capacities, *Annals of Operations Research*, Vol. 61, 1995, pp. 189-211
- [6] Bernard K.-S. Cheung, K. L. Choy, C.-L. Li, W. Shi, J. Tang, Dynamic routing model and solution methods for fleet management with mobile technologies, *International Journal of Production Economics*, Vol 113, 2008, pp. 694-705
- [7] DJI Matrice 300 RTK Specification, 2021, URL: <https://www.dji.com/hu/matrice300/specs>, last visit: 2023. 05. 10.
- [8] J. Zhang, J.F. Campbell, D.C. Sweeney, A.C. Hupman, Energy Consumption Models for Delivery Drones: A Comparison and Assessment, *Transportation Research Part D: Transport and Environment*, Vol. 90., 2020
- [9] K. Braekers, K. Ramaekers, I. V. Nieuwenhuyse, The vehicle routing problem: state of the art classification and review, *Computers & Industrial Engineering*, vol 99., 2016, pp. 300-313
- [10] I. Lovas, Fixed point iteration-based adaptive controller tuning using a genetic algorithm, *Acta Polytechnica Hungarica*, vol. 19, no. 2, 2022, 59-77.
- [11] K. Czakoóová, O. Takáč (2020), The application of modern technologies for image processing and creating real model in teaching computer science at secondary school, *ICERI2020 Proceedings*, pp. 6180-6187.
- [12] Namestovski, Ž.; Kovari, A. Framework for Preparation of Engaging Online Educational Materials—A Cognitive Approach. *Appl. Sci.* 2022, 12, 1745.
- [13] István Á. Harmati, Robert Fullér, Imre Felde, On stability of maximal entropy OWA operator weights, *Fuzzy Sets and Systems*, vol. 448, 2022, pp. 145-156,

