Noise Abatement Trajectories for a UAV Delivery Fleet

Geoffrey Scozzaro, Daniel Delahaye, Adan Ernesto Vela

Noise Abatement Trajectories for a UAV Delivery Fleet

Geoffrey Scozzaro
Université de Toulouse ENAC
7 Avenue Edouard Belin
31400 Toulouse, France
geoffreyscozzaro27@gmail.com

Daniel Delahaye
Université de Toulouse ENAC
7 Avenue Edouard Belin
31400 Toulouse, France
delahaye@recherche.enac.fr

Adan Ernesto Vela
University of Central Florida
4000 Central Florida Blvd
Orlando FL 32816, United States
Adan.Vela@ucf.edu

***Abstract*—The use of unmanned aerial vehicles (UAV) for package delivery is one potential solution for resolving the *last-mile* delivery problem. And despite their potential benefits, UAV delivery fleets are concerning to communities due to their potential noise impact. In response to noise concerns this paper seeks to reduce impact of a UAV delivery-fleet on communities by generating an optimized set of noise-conscience trajectories. Through simulation, we demonstrate that by optimizing flight trajectories it is possible to significantly limit their overall impact to communities with only a marginal increase in energy/fuel costs.**

Keywords: UAVs, Trajectories, Simulated annealing, Noise.

I. Introduction

Utilization of Unmanned Aerial Vehicles (UAV) is likely to be a key solution in resolving the last-mile delivery problem in the coming years [1]. As such, numerous technology companies are making significant investments into developing and deploying UAV delivery fleets, this includes GoogleX, which was recently approved to operate as an airline in the United States by the Federal Aviation Authority (FAA) [2]. Moreover, UAV demand forecasts [3] suggest significant UAV air traffic congestion in the urban areas. The future use of UAVs raises several significant community concerns like noise pollution (and certainly privacy). Indeed, over the past few years aviation-related noise annoyance has increased [4], despite advances in procedure design and the phasing out of louder engines. UAVs present a unique challenge as some consider their sound profile akin to the high-pitch noise of honeybees [5]. In fact a National Aeronautics and Space Administration (NASA) study indicates that the noise produced by UAVs is far more annoying to people than traditional aviation noise [6].

Considering the most common multi-rotor prosumer UAVs, each generally produces noise at around 80dB [7]. This noise level is equivalent to that of a busy intersection. Given a potential future with ubiquitous UAV delivery fleets, we are presented with the prospect of adding a loud and annoying noise source within many communities, and just outside homes near distribution points or common UAV flight routes. Therefore, the aviation community and relevant stakeholders should address how to best limit the noise impact of large-scale UAV fleets, while still providing an ecosystem engendering a burgeoning market-place. Accordingly, this work seeks to apply an operations research framework to minimize the noise pollution of a UAVs fleet. Using a point-source spherical propagation noise model for UAVs [8], we simulate fleets operating at low altitudes. Using simulated annealing the flight paths are optimized to limit noise, while taking into consideration energy expenditure.

The remainder of the paper is structured as follows. A description of the problem definition is provided in section II. Section III presents a simplified noise model. A path planning framework is defined in section IV, while the optimization modeling and the simulated annealing technique used to optimize routes are introduced in Section V and Section VI. The implementation and subsequent results are put forward in Section VII and Section VIII.

II. Problem description and Related Works

Consider a set of N origin-destination pairs located within a pre-defined boundary, as shown in Figure 1. Each origin-destination pair is defined by a corresponding set of points A_i and B_i (with $i \in \{1, \dots N\}$), and a demand-density D_i parameter indicating the nominal distance-in-trail between subsequent UAVs operating on the route [1]. For simplicity, aerial conflicts are not taken into account and the UAV fleet is assumed to have homogeneous performance. The bounded region, representing an urban or suburban environment, is divided into a regular grid of size n-by-m, whereby origin and destination points are set to be located at grid-points.

As UAVs traverse between their origins and destinations they will generate noise that is projected onto the urban landscape below. Through the accumulation of UAV operating on multiple routes within the region, specific areas (represented by grid cells) are expected to exceed allowed noise thresholds. The goal therefore is to distribute routes so as to minimize the noise impact of UAVs repeatedly traveling over the same areas, while remaining aware that distributing routes may extending route lengths or require UAV to climb to higher altitudes, thereby increasing the energy required to service a route.

The origin-destination demand, in combination with a gridded spatial representation finds direct correspondence with

[1]Consideration of the demand-density is consistent with the expectation that multiple UAVs or a single UAV provides service to a region by traversing back-and-forth between a distribution center and a neighborhood

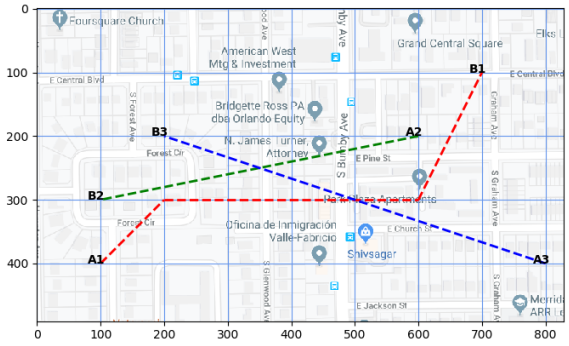


Figure 1: City map divided by a grid with origin-destination pairs.

the canonical transportation problem in operations research of Strategic Conflicts detection [9]. Such a node-based representation grows quickly according to the planer and vertical discretization, which even for standard aviation problems is difficult to construct and solve in real-time. Moreover, a gridded spatial representation artificially restricts UAVs to traverse predefined street-like segments as in [10], which in themselves are sub-optimal as they do not allow for direct point-to-point paths.

To overcome the challenges associated with the standard transportation problem, this work makes use of segment piece-wise linear routes like the dashed route illustrated in Figure 1. By defining UAV routes in such a manner, the routes can be arbitrarily adjusted by translating key inflection points in the paths. A similar approach for generating inflection points for three-segment piece-wise linear routes is proposed in [11] for strategic air conflict reduction problem. Here, we focus on a similar problem where routes contain UAV flows, however, our aim is to space traffic for the goal of noise mitigation. In this context, exploring a broader search space with a route generation model allowing an arbitrary number of inflection points and thus larger deviations seems more relevant. A generation model of such routes is presented in section IV.

Numerous optimization methods have already been implemented to solve problems related to Air Traffic Flow Management. [12] and [13], respectively, put forward an implementation of genetic algorithms and constraint programming to solve air conflict problems. Such methods can be limited by the large size of a search space. Indeed, genetic algorithms need to duplicate the simulation environment which may require an excessive amount of memory. Constraint programming may be limited by the number of constraints which often see exponential increases in problem-size with the dimension of the search space. One algorithm often able to overcome challenges related to computation feasibility within large search spaces is simulated annealing. Its efficiency has already been shown in Air Traffic Flow Management with a broad search space in [11] [14] [15] or [16]. This algorithm is introduced in section VI.

Furthermore, as noise modeling and concurrent nonlinear path generation and evaluation is computationally difficult within standard optimization formulation, we make sure of a repeating two-step process involving optimization and evaluation within a simulated environment. This process is illustrated in Figure 2. The benefit of such an optimization framework is that fleet-level routing decisions can be optimized without the need for complex noise models embedded within an optimization formulation. Moreover, this process is able to work with only one simulation environment, thereby limiting computing memory requirements.

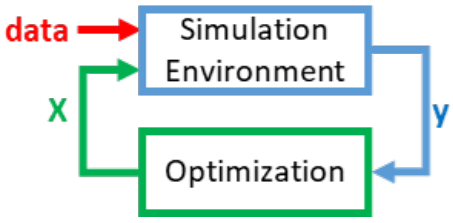


Figure 2: Objective function evaluation based on a simulation process

III. Noise Modeling

To date there exist numerous models of sound propagation, many taking into account complexities like wind and other weather related phenomena [17], and reflection or absorption off buildings, canopies, and geographic features [18]. To enable rapid calculation within a simulation-based optimization framework we make use of a point-source spherical propagation model for sound generated by each UAV (see the corresponding depiction of the model in Figure 3). Although such a point-source model makes numerous simplifications for computational efficiency, it remains reasonably representative of prior finding established in [8]. So despite multi-rotor UAVs having a highly directional component to the noise they produce in their immediate space, at the distances considered we assume the noise propagating from the rotors moves outward isotropically. Another consequence of the model is that as the UAVs traverse their routes, the noise pattern generated on the ground does not distort when turning. When modeling the UAVs we assume they produce noise at 80dB at 1m in line with existing studies [7].

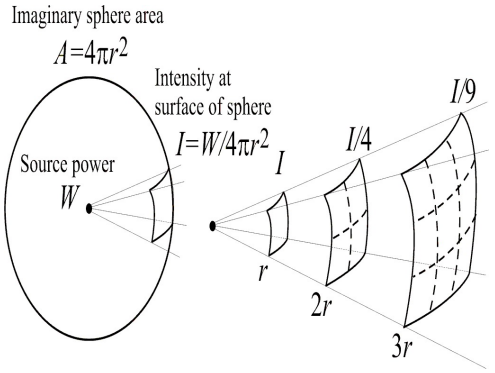


Figure 3: Source point model with a spherical propagation

IV. Route Generation Model

Describing and generating routes will occur within a grid as illustrated in Figure 1. The grid lines can correspond to roads, with nodes representing both intersections and points of interest. For this paper the dimension of the grid is linked to the dimension of a downtown area with commercial and residential neighborhoods; the grid is assumed to be regularly spaced with grid cells of 20x20 meters squared. Every grid cell is associated with a location, a maximum building height, and a noise sensitivity level. The noise sensitivity takes into account population and zoning (residential, commercial, industrial). For example, for a grid corresponding to a commercial downtown area with large buildings, the associated building heights will be high and its noise sensitivity will be low. On the contrary, if the grid cell is in a single-family home residential area, its building heights will be low and its noise sensitivity will be high.

Our model formulation will allow UAVs to traverse along or across grid lines given that they remain above buildings. To maintain compact descriptions of the trajectories, the UAV are constrained to a single altitude while cruising, that means they will vertically climb and descend while holding in place above their origin or destination. The cruise altitude must remain above the building heights over all grid cells it passes over. When flying over buildings the UAV will produce noise that radiates towards each grid cell, however, the simulated environment takes into account noise masking effects by neighboring buildings. Noise masking is determined by the height of buildings between a UAV and any nearby grid cells. For this paper noise masking is taken to be absolute; noise reflection is not accounted for in the simulation environment.

In order to generate trajectories between any origin and destination A and B, a supplemental coordinate frame and grid is stretched between the points as indicated by Figure 4. In the coordinate frame two new points C and D, contained within the rectangle defined by A and B, are randomly placed. The points C and D generate another rectangle inside the A-B rectangle. As illustrated by Figure 4, one can connect all points moving from A towards B to generate numerous trajectories between A and B. Yen's algorithm [19] makes possible to enumerate the k-shortest paths between two points and, thus, to count all the paths between two points. In the event that A,B,C and D have all different coordinates, Yen's algorithm enumerates 120 unique paths from A to B. This set of paths will be registered on a file and used for pre-processing by the optimization algorithm.

V. Optimization Modeling

In this section we provide a more detailed description of the problem definition, along with the decision variables, constraints, and objective function of the optimization problem.

A. Problem definition

To reiterate, a problem instance is defined by: **(1)** a $n * m$ grid representing a region where each grid cell is defined by an altitude, a position, a noise sensitivity coefficient. Initially, UAVs noise exposure in a grid cell is initialized at 0, the noise exposure level is updated with each additional UAV entering the airspace; **(2)** a set of N missions defined by origin and destination points A and B; **(3)** a set of 120 paths generated by the 2-rectangles method described in Section IV; **(4)** a list of allowed altitudes for each route; and **(5)** the noise produced by a UAV along a route as denoted by "UAVNoise".

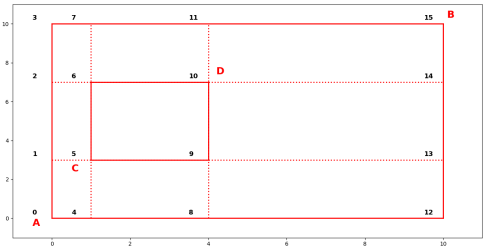


Figure 4: Partial graph generated by A(0,0) B(10,10), C(1,3) and D(4,7)

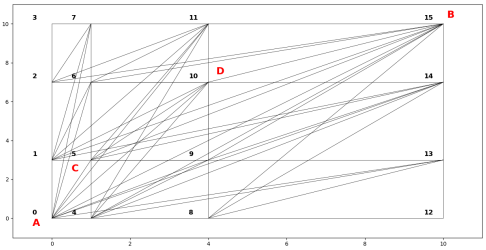


Figure 5: Complete graph generated by A(0,0) B(10,10), C(1,3) and D(4,7)

B. Decision Variables

The decision variables represent the variables on which the optimization algorithm acts on, in order to optimize the objective function. The decision variables for the N routes are gathered into a vector $\vec{X} = [\vec{d_1}, \vec{d_2}, ..., \vec{d_N}]^T$. In our problem the decisions $\vec{d}$ for a route is represented by a set of variables: **(1)** the position of points C and D, (x_C, y_C) and (x_D, y_D); **(2)** a variable indicting which of the 120 path options connecting the origin and destination is selected for the route; and **(3)** a cruise altitude, h.

C. Constraints

For our problem, the constraints only relate to the allowed positions of the points C and D $((x_C, y_C), (x_D, y_D))$ and the allowed cruise altitude h. The points C and D must be located inside the rectangle constructed by the origin point A (x_A, y_A) and the destination point B (x_B, y_B). Supposing that A is located on the lower-left corner of the rectangle and B is located on the upper-right corner of the rectangle (so that $x_A < x_B$ and $y_A < y_B$, like in Figure 4), and that the point C is located on the lower-left corner of the rectangle built with

the points C and D, then the following constraints exists:

$$x_A < x_C < \frac{x_B - x_A}{2}$$
$$\frac{x_B - x_A}{2} < x_D < x_B$$
$$y_A < y_C < \frac{y_B - y_A}{2}$$
$$\frac{y_B - y_A}{2} < y_D < y_B$$

With regards to the cruise altitude, h, once the path is set, the UAV must safely clear all buildings while remaining below maximum altitude restriction. This constraint can be expressed as $h_{min} \leq h \leq h_{max}$, where h_{min} is path dependent, while h_{max} is tied to airspace altitude restrictions.

D. Objective Function

Each decision d_k associated with the k^{th} route manifests a trajectory of length M_k, denoted γ_k, given by a set of points $\gamma_k = \{\gamma_{k,1}, \gamma_{k,2}, ..., \gamma_{k,M_k}\}$ so that $\gamma_{k,1} = A$ and $\gamma_{k,M_k} = B$. For each point $p = \gamma_{k,i}$, we define its 3D distance to a grid box c according to the euclidean distance $d(p,c) = \sqrt{(x_p - x_c)^2 + (y_p - y_c)^2 + (z_p - z_c)^2}$. Accordingly, the noise produced on the grid point c by the UAV when it is located at point $p = \gamma_{k,i}$ of its trajectory is:

$$No_{p,c} = \frac{UAVNoise}{4 \times \pi \times d(p,c)^2} \tag{1}$$

Figure 6 shows an example of noise footprint of a UAV trajectory on the grid. The noise produced by a fleet of UAVs on the same grid cell c is:

$$No_c = \sum_{i=1}^{K} \sum_{p \in traj_{X_i}} No_{p,c} \tag{2}$$

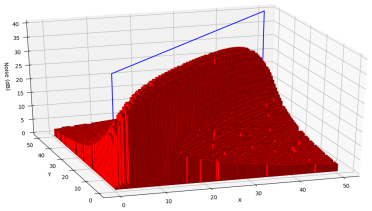


Figure 6: Noise map of a trajectory between A(0,0) and B(50,50) at the altitude h=40m

Consideration of energy expenditure is calculated through a weighted path length taking into account both vertical and horizontal segments. Specifically, for trajectory γ_k with points $\{\gamma_{k,1}, \gamma_{k,2}, ..., \gamma_{k,M_k}\}$, the weighted path-length is given by

$$L(\gamma_k) = K_1 \times d(\gamma_{k,1}, \gamma_{k,2}) + \sum_{i=0}^{M_k - 2} d(\gamma_{k,i}, \gamma_{k,i+1}) + K_2 \times d(\gamma_{k,M_k-1}, \gamma_{k,M_k})$$

where the coefficients K_1 and K_2 are cost weightings corresponding to the climb and descent of a UAV.

Balancing noise impact and energy expenditure occurs through the weighting coefficient Q, the associated costs for any route (as defined by decision variable d_k) is then

$$C(d_k) = \sum_{p \in traj_{X_k}} \sum_{c \in range(p)} \mathbb{1}(No_c) \times No_{p,c} + Q \times L(\gamma_k)$$

where

$$\mathbb{1}(noise) = \begin{cases} 1 & \textit{if noise is above 55dB} \\ 0 & \textit{else} \end{cases}$$

is an indicator function for when the noise in a grid cell exceeds allowed thresholds. Over a fleet of UAV it follows that the total costs are:

$$f = \sum_{i=1}^{N} C(d_i).$$

VI. Simulated Annealing

One can demonstrate that the optimization problem as described in Section V is NP Hard. Accordingly, we then propose to use meta-heuristics to solve it. Simulated annealing (SA) is one such metaheuristic methods technique to solve *black box* global optimization problems, whose objective function is not explicitly given, yet can be calculated via simulation. This optimization method is often quite powerful in real-life applications, especially when presented with a large search space. The strength of Simulated Annealing is illustrated in [20] with a large-scale aircraft trajectory planning problem. The similarity of the problem to the problem considered in this paper invites us to use simulated annealing in order to reduce the noise impact of the UAVs fleet.

Simulated annealing is defined according to two parameters: c_k, commonly referred to as the temperature parameter, and L_k, the number of transitions generated at iteration k. The SA algorithm can be summarized as follows:

Simulated annealing

1) **Initialization** $i := i_{start}$**,** $k := 0$**,** $c_k = c_0$**,** $L_k := L_0$**);**
2) **Repeat**
3) **For** $l = 0$ **to** L_k **do**
   - **Generate a solution** j **from the neighborhood** S_i **of the current solution** i**;**
   - **If** $f(j) < f(i)$ **then** $i := j$ (j **becomes the current solution);**
   - **Else,** j **becomes the current solution with probability** $e^{\left(\frac{f(i)-f(j)}{c_k}\right)}$**;**
4) $k := k + 1$**;**
5) **Compute**(L_k, c_k)**;**
6) **Until** $c_k \simeq 0$

One of the primary features of simulated annealing is its ability to accept transitions that degrade the objective function. This makes it possible to avoid becoming trapped in a local optimum.

At the beginning of the process, the value of the temperature c_k is high, which makes it possible to accept transitions with high objective degradation, and thereby to explore the state space thoroughly. As c_k decreases, only the transitions improving the objective, or with a low objective deterioration, are accepted. Finally, when c_k tends to zero, no deterioration of the objective is accepted, and the SA algorithm behaves like a Monte Carlo algorithm.

In many optimization applications, the objective function is evaluated thanks to a computer simulation process which requires a simulation environment. In such a case, the optimization algorithm controls the vector of decision variables, X, which are used by the simulation process in order to compute the performance (quality), y, of such decisions, as shown in Figure 2.

In the standard simulated annealing algorithm, a copy of a state space point is requested for each proposed transition. In fact, a point $\vec{X}_j$ is generated from the current point $\vec{X}_i$ through a copy in the memory of the computer. In the case of state spaces of large dimension, the simple process of implementing such a copy may be inefficient and may reduce drastically the performance of simulated annealing. In such a case, it is much more efficient to consider a *come back* operator, which cancels the effect of a generation. Let G be the generation operator which transforms a point from $\vec{X}_i$ to $\vec{X}_j$:

$$\vec{X}_i \xrightarrow{G} \vec{X}_j$$

the comeback operator is the inverse G^{-1} of the generation operator.

Usually, such a generation modifies only one component of the current solution. In this case, the vector $\vec{X}_i$ can be modified without being duplicated. Depending on the value obtained when evaluating this new point, two options may be considered:

1) the new solution is accepted and, in this case, only the current objective function value is updated.
2) else, the come back operator G^{-1} is applied to the new position in order to come back to the previous solution, again without any duplication in the memory.

This process is summarized in Figure 7.

The next section presents how such simulated annealing algorithm has been implemented to solve our problem.

VII. Implementation

Having computing all the possible paths, one must only consider the number of the path and the position of points C and D in order to be able to generate the UAV trajectory.

As for any optimization algorithm one must be able to initialize the starting point ($\vec{X}_0$). To do that, decision is initialized by the shortest path between the departure point A and the arrival point B which is the diagonal between A and B. The initial altitude is the same for each drone and is an input of the algorithm. Points C and D are randomly chosen into the rectangle A, B.

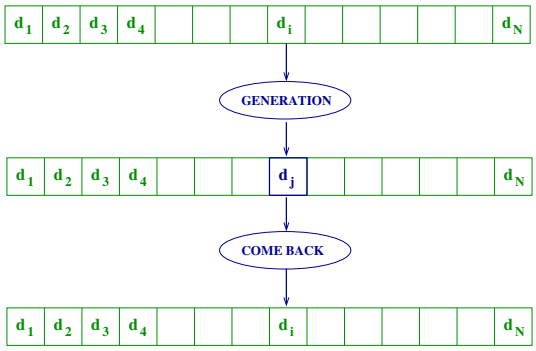


Figure 7: Optimization of the generation process. In this figure, the state space is built with a decision vector for which the generation process consists of changing only one decision (d_i) in the current solution. If this modification is not accepted, this component of the solution recovers its former value. The only information to be stored is the integer i and the decision $\vec{d}_i$.

When the vector of decisions is initialized, we then compute the associated trajectories which are inserted in the airspace. The noise footprint is then computed and one can then determine the individual performance for each decision. Based on this performance indicator, decisions with the worse performance are more often selected by the neighborhood operator of the simulated annealing. This operator will then change this decision in order to create a new trajectory for the associated UAV. This process works in the following way:

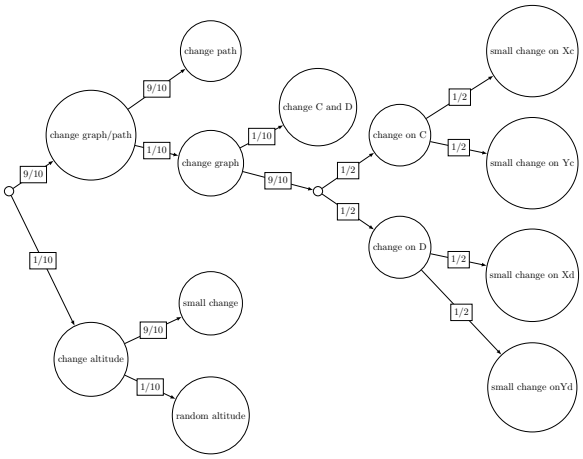
- 1/10 times altitude is changed. This modification consists in operating a small variation on the current altitude most of the time and sometimes picking a random altitude.
- 9/10 times path is modified. Two options are available to change it:
  - keep the same graph and choose another path.
  - change the graph. 1/10 times random new points C and D are selected. Otherwise C or D is randomly picked and a small change is operated on it 9/10 times. Else a random new point is selected.

Probabilities can be adjusted depending on initial conditions. This generation of a neighbor is summarized in Figure 8.

When the new decision is computed through the neighborhood operator, one can compute the associated trajectory. Then, we first remove the former trajectory associated to this decision from the airspace. We update the noise footprint and we then put the new trajectory in the airspace and update again the noise footprint. Finally, the performance indicators are updated for all the decisions. This principle is summarized in Figure 9.

VIII. Results

The optimization framework was implemented in JAVA and executed in Ubuntu 18.04 on an Intel 4th Generation Core i7. The performance of the algorithm was tested on two major scenarios with an execution time of 300s for both scenarios. Each scenario is described below.



1

Figure 8: Probability tree summarizing the neighbor generation process

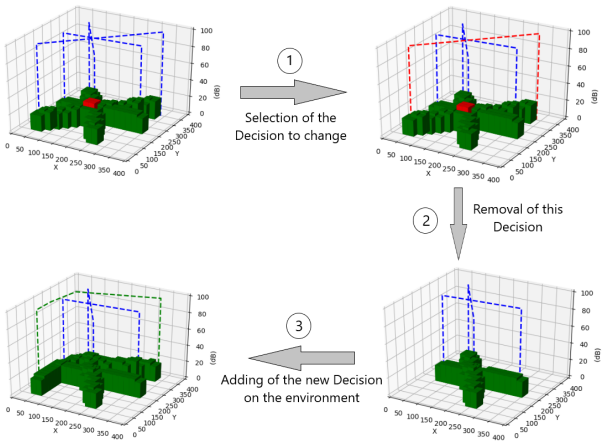


Figure 9: Illustration of the Implementation process. Red and green cubes represent respectively areas where the noise is above and under the threshold.

Scenario I

-The dimensions of the map are 50*50
-The map is composed of 50% housing estate areas and 50% commercial/industrial building areas.
-The possible flight altitudes for each drone are {10m,15m,20m,25m,30m,35m}.
-The initial altitude of each drone is 35m.
-The number of missions is K=500.
-The coefficient Q of the objective function is calibrated so that the noise impact proportion and the distance proportion are respectively 90% and 10%.

Scenario II a)

-The dimensions of the map are 100*100
-The map is composed of 20% housing estate areas and 80% commercial/industrial building areas.
-The possible flight altitudes for each drone are {10m,15m,20m,25m,30m}.
-The initial altitude of each drone is 30m.
-The number of missions is K=1000.
-The coefficient Q of the objective function is calibrated so that the noise impact proportion and the distance proportion are respectively 95% and 5%.

Scenario II b)

This scenario is similar to the prior one except that the possible flight altitudes are expanded to: {10m,15m,20m,25m,30m,35m,40m,45m,50m}. The initial altitude remains equal to 30m.

Simulation

Figure 10 shows the noise map before and after optimization and Table I summarizes both solutions.

One can see Scenario I as an illustration of a city neighborhood. The high number of residential areas suggests a low density zone where noise annoyance would be a matter of deep concern. Thus, the noise weighting of the objective function is selected to be nine times higher than the distance weighting. The altitude of residential areas and commercial/industrial areas are respectively assumed equal to 5 and 20 meters. Finally, the noise coefficient tolerance is 10 times higher for housing estate areas than for commercial/industrial areas in order to give priority to the deactivation of noise thresholds in residential areas. After optimization, one can notice in the Scenario I respectively a reduction by a third and by a half for the number of thresholds activated in commercial/industrial areas and in housing estate areas. This involves an increase of 16.5% for the average distance. Optimization is here in 2D because drones are already flying at the maximum altitude and decreasing drones altitude is likely to increase the UAVs noise footprint. That's why the algorithm reduces the noise on the ground by spreading the drones on the whole map.

The Scenario II is illustrative of the case when considering a city center with taller commercial buildings and less residential areas. In this scenario, more UAVs are added to the simulation so that the airspace gets congested by drones. Their altitude can only be increased in the case b). Thus, put the altitude as a decision parameter in the case a) becomes inefficient in practice as in Scenario I. Due to the large number of UAVs, noise proportion of the objective function is even more higher than in the previous Scenario in order to focus on the reduction of the noise. After optimization in the Scenario II a), 84 thresholds in residential areas are suppressed but 112 additional thresholds in building areas are activated. The

TABLE I: RESULTS FOR THE TWO SCENARIOS

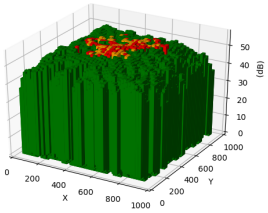
Features / Status	AA^0	AD^1	$nbTABA^2$	$nbTAHEA^3$
Before Optimization	35m	0.61km	185	101
After Optimization	35m	0.71km	119	46
Total	+0%	+16.5%	-35.6%	- 54.5%

(a) Results Scenario I

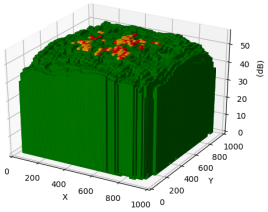
Features / Status	AA^0	AD^1	$nbTABA^2$	$nbTAHEA^3$
Before Optimization	30m	1.10km	1158	231
After Optimization II a)	30m	1.27km	1270	167
Total II a)	+0%	+15.4%	+7.7%	-27.7%
After Optimization II b)	39.55m	1.26km	30	2
Total II b)	+31.8%	+14.5%	-97.4%	-99.1%

(b) Results Scenario II

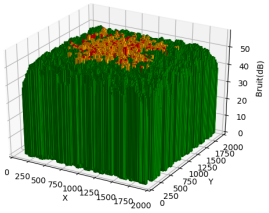
0) AA:Average Altitude
1) AD:Average distance travelled by each drone
2) nbTABA:number of Thresholds Activated in Building Areas
3) nbTAHEA: number of Thresholds Activated in Housing Estate Areas



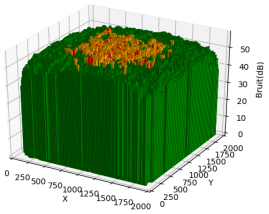
(a) Noise Map Scenario I before optimization



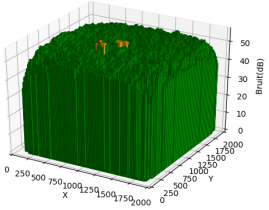
(b) Noise Map Scenario I after optimization



(c) Noise Map Scenario II before optimization



(d) Noise Map Scenario II a) after optimization



(e) Noise Map Scenario II b) after optimization

Figure 10: Noise map before and after optimization for both scenario. Red and orange cubes represent respectively areas where noise is above 55dB in housing estate areas and in commercial/industrial areas. Green cubes represent areas where noise is under 55dB.

algorithm reduces the noise produced in housing estate areas by increasing it in building areas. Because of the number of drones, changing trajectories at the same altitude only involves a transfer of the noise in one area to another. In the Scenario II) b) drones are allowed to climb in order to reduce their noise footprint. Therefore, the average altitude raises by 31.8%. This was expected due to the fact that noise propagation follows an inverse-square law. Thus, flying at the highest altitude allows to significantly decrease the noise. Indeed, nearly all the thresholds activated have been suppressed, especially 99.1% for those in residential areas. The average altitude doesn't reach the maximum allowed altitude because the algorithm changes only the altitude of trajectories which are above noisy areas. One can notice an increase in the average distance traveled. This increase is the consequence of the climbing of drones and its diversion following an extended trajectory to avoid zones already overwhelmed by others UAVs.

In both scenarios the first aim is reached because we notice a significant reduction of noise in housing estate areas. If drones are not allowed to climb , the algorithm is less efficient but still reduces the noise on the ground by spreading the UAVs across the region.

Finally, we can wonder about the operational cost to implement these new solutions. The increase of the average distance in both scenarios suggests it would be higher than the initial solution. However this increase seems small in comparison with the decrease of the noise in residential areas.

IX. Conclusion

This paper has presented a way to evaluate the noise produced by a UAVs fleet in an urban environment. Then we introduced a new method to generate quickly a set of trajectories and to efficiently optimize them through the use of simulated annealing. Results are encouraging as the technique effectively distributes noise with minimal impact to operating costs. We must note however, the noise propagation model considered within the optimization frame is simple and doesn't take into the complexities of noise annoyance. As such we suggest additional human subject studies on UAV annoyance with regards to noise. Such studies will allow us to improve our embedded noise models contained within the simulations. Moreover time is not taken into account in our simulation. In order to add value to this study, we suggest to include the departure time of each UAVs as a decision variable. Aerial conflicts should also be taken into consideration. Finally, future work will apply this approach in a simulated environment based on a real city.

References

[1] Simran Brar, Ralph Rabbat, Vishal Raithatha, George Runcie, and Andrew Yu. Drones for deliveries, November 2015. https://scet.berkeley.edu/wp-content/uploads/ConnCarProjectReport-1.pdf.

[2] Google Wing drones approved for US home deliveries. Bbc, april 2019. https://www.bbc.com/.

[3] D. Jenkins, B. Vasigh, C. Oster, and T. Larsen. Forecast of the commercial uas package delivery market, May 2017. https://news.erau.edu/-/media/files/news/forecast-commercial-uas-package-delivery-market.pdf.

[4] Wolfgang Babisch, Danny Houthuijs, Göran Pershagen, Ennio Cadum, Klea Katsouyanni, Manolis Velonakis, Marie-Louise Dudley, Heinz-Dieter Marohn, Wim Swart, Oscar Breugelmans, et al. Annoyance due to aircraft noise has increased over the years—results of the hyena study. *Environment international*, 35(8):1169–1176, 2009.

[5] Raya Islam and Dr. Alexander Stimpson. Small uav noise analysis. 2017.

[6] Andrew W Christian and Randolph Cabell. Initial investigation into the psychoacoustic properties of small unmanned aerial system noise. In *23rd AIAA/CEAS Aeroacoustics Conference*, page 4051, 2017.

[7] Fredrik Christiansen, Laia Rojano-Doñate, Peter T Madsen, and Lars Bejder. Noise levels of multi-rotor unmanned aerial vehicles with implications for potential underwater impacts on marine mammals. *Frontiers in Marine Science*, 3:277, 2016.

[8] Vishwanath Bulusu, Leonid Sedov, and Valentin Polishchuk. Noise estimation for future large-scale small uas operations. *Institute of Noise Control Engineering*, nov 2017.

[9] Karl Dörner, Ivana Ljubic, Georg Pflug, and Gernot Tragler. *Operations Research Proceedings 2015: Selected Papers of the International Conference of the German, Austrian and Swiss Operations Research Societies (GOR, ÖGOR, SVOR/ASRO), University of Vienna, Austria, September 1-4, 2015*. 01 2017.

[10] Harald Bayerlein, Paul De Kerret, and David Gesbert. Trajectory optimization for autonomous flying base station via reinforcement learning. In *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5. IEEE, 2018.

[11] Supatcha Chaimatanan, Daniel Delahaye, and Marcel Mongeau. A methodology for strategic planning of aircraft trajectories using simulated annealing. In *ISIATM 2012, 1st International Conference on Interdisciplinary Science for Air traffic Management*, pages pp–xxxx, 2012.

[12] Sofiane Oussedik and Daniel Delahaye. Reduction of air traffic congestion by genetic algorithms. In *International Conference on Parallel Problem Solving from Nature*, pages 855–864. Springer, 1998.

[13] Nicolas Barnier and Cyril Allignol. 4d-trajectory deconfliction through departure time adjustment. 2009.

[14] Arianit Islami, Supatcha Chaimatanan, Moaxi Sun, and Daniel Delahaye. Optimization of military missions impact on civilian 4d trajectories. 2017.

[15] Olga Rodionova, Mohammed Sbihi, Daniel Delahaye, and Marcel Mongeau. Optimization of aicraft trajectories in north atlantic oceanic airspace. 2012.

[16] Pierre Dieumegard, Supatcha Chaimatanan, and Daniel Delahaye. Large scale adaptive 4d trajectory planning. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pages 1–9. IEEE, 2018.

[17] L Nijs and CPA Wapenaar. The influence of wind and temperature gradients on sound propagation, calculated with the two-way wave equation. *The Journal of the Acoustical Society of America*, 87(5):1987–1998, 1990.

[18] Richard H Lyon. Role of multiple reflections and reverberation in urban noise propagation. *The Journal of the Acoustical Society of America*, 55(3):493–503, 1974.

[19] E.Q.V Martins and M.M.B Pascoal. A new implementation of yen's ranking loopless paths algorithm. In *4OR*, 2003.

[20] Daniel Delahaye, Supatcha Chaimatanan, and Marcel Mongeau. Simulated annealing: From basics to applications. In Gendreau M. and Potvin JY., editors, *Handbook of Metaheuristics. International Series in Operations Research and Management Science*. Springer, Cham, 2018.