N. Hohmann, M. Bujny, J. Adamy and M. Olhofer, "Multi-objective 3D Path Planning for UAVs in Large-Scale Urban Scenarios," 2022 IEEE Congress on Evolutionary Computation (CEC), 2022, pp. 1-8, doi: 10.1109/CEC55065.2022.9870265.
https://ieeexplore.ieee.org/document/9870265

Multi-objective 3D Path Planning for UAVs in Large-Scale Urban Scenarios

Nikolas Hohmann*, Mariusz Bujny†, Jürgen Adamy* and Markus Olhofer†
*Control Methods & Robotics Lab, Technical University of Darmstadt, Darmstadt, Germany
Email: {nikolas.hohmann, adamy}@rmr.tu-darmstadt.de
†Honda Research Institute Europe GmbH, Offenbach, Germany
Email: {mariusz.bujny, markus.olhofer}@honda-ri.de

Abstract—**In the context of real-world path planning applications for Unmanned Aerial Vehicles (UAVs), aspects such as handling of multiple objectives (e.g., minimizing risk, path length, travel time, energy consumption, or noise pollution), generation of smooth trajectories in 3D space, and the ability to deal with urban environments have to be taken into account jointly by an optimization algorithm to provide practically feasible solutions. Since the currently available methods do not allow for that, in this paper, we propose a holistic approach for solving a Multi-Objective Path Planning (MOPP) problem for UAVs in a three-dimensional, large-scale urban environment. For the tackled optimization problem, we propose an energy model and a noise model for a UAV, following a smooth 3D path. We utilize a path representation based on 3D Non-Uniform Rational B-Splines (NURBS). As optimizers, we use a conventional version of an Evolution Strategy (ES), two standard Multi-Objective Evolutionary Algorithms (MOEAs) – NSGA2 and MO-CMA-ES, and a gradient-based L-BFGS-B approach. To guide the optimization, we propose hybrid versions of the mentioned algorithms by applying an advanced initialization scheme that is based on the exact bidirectional Dijkstra algorithm. We compare the different algorithms with and without hybrid initialization in a statistical analysis, which considers the number of function evaluations and quality features of the obtained Pareto fronts indicating convergence and diversity of the solutions. We evaluate the methods on a realistic 3D urban path planning scenario in New York City, based on real-world data exported from OpenStreetMap. The examination's results indicate that hybrid initialization is the main factor for the efficient identification of near-optimal solutions.**

Index Terms—**multi-objective optimization, three-dimensional, path planning, hybrid algorithms, evolutionary algorithms, UAV, unmanned aerial vehicle**

I. Introduction

Applications of Unmanned Aerial Vehicles (UAVs) are becoming more and more widespread in our cities. Whereas early use cases of UAVs mainly considered (nearly) unpopulated environments, e.g., in agricultural crop monitoring [1] or power line inspection [2] tasks, urban applications, where UAVs operate in close proximity to humans, are not far from becoming very common. Quite recently, the practical purpose of UAVs has been extended by tasks such as package delivery [3] or sensor data acquisition in smart cities [4]. In cities, autonomous flying agents encounter a more complex environment than in open-terrain applications. Cities are populated by thousands of people that could be seen as individual autonomous agents themselves. Therefore, we can see cities with UAV infrastructure as a huge multi-agent, human-machine system. As always in such a system, humans have to be safeguarded against potential threats emanated by machines. More precisely, the negative impacts that UAVs have on city residents must be minimized, such as the hazard of air crashes that could harm people underneath or the noise pollution generated by a propeller-driven aircraft. For example, according to Torija et al. [5], noise is the main harm to avoid, in order to gain social acceptance for UAVs operating in cities.

However, safety and noise protection can not be the single objective. If so, an optimizer would probably find a global optimum by ”flying around the city”, which would be a highly inefficient solution considering travel time and energy consumption. Therefore, minimizing energy costs, is another legitimate objective when planning paths through the city.

In this work, we formulate a two-objective, constrained optimization problem considering smooth, three-dimensional paths. A path is optimized regarding 1) the noise immission on the city residents and 2) the energy consumption needed to follow the path. Additionally, by formulating several safety constraints, we ensure that the UAVs 1) do not collide with static obstacles and 2) keep a minimal flight height.

One way of solving an optimization problem with several (often contradictory) objectives, is to weight and aggregate the objectives and solve a single-objective problem. This would require the user to choose appropriate weights for the objectives without knowing how a different weighting influences the solution. Another way to solve a Multi-Objective Optimization (MOO) problem is by evaluating solutions independently with respect to all E objectives. A solution belongs to the so-called Pareto front in the E-dimensional objective space if no other solution exists that is 1) better regarding one objective and 2) not worse regarding the other objectives. The Pareto front is the output of an MOO approach. The users can then select an adequate solution by either choosing it manually according to their needs or by applying a solution selection scheme [6].

It has been shown that Evolutionary Algorithms (EAs) are well-suited to identify diversified Pareto fronts in MOO problems, also for multimodal objective functions. On the downside, EAs need high computational resources and can not guarantee optimality. In a recent study [7], it was shown that the performance of EAs in *two-dimensional* MOPP problems can be increased significantly by an advanced initialization step. However, in order to obtain practically feasible paths for aerial vehicles while making use of the available physical degrees of freedom, it is necessary to search paths in the 3D space. In this work, we solve the MOPP problem by planning smooth paths in the *three-dimensional* space. Thereby, on the one hand, multiple new problems, like an increased search space dimension, or a higher number of local extrema, occurs. On the other hand, using three dimensions gives us the opportunity to find more flexible solutions that would not exist in 2D. Thus, we propose new formulations of objective functions and constraints adapted to the 3D environment. Moreover, we develop a preprocessing step that is able to integrate those constraints during the advanced initialization in three dimensions. This initialization step, which utilizes a bidirectional Dijkstra [8] approach, constitutes the first part of our hybrid approach. The second part contains the main optimization step. As main optimizers, we use a basic Evolution Strategy (ES) [7], the state-of-the-art multi-objective Non-dominated Sorting Genetic Algorithm (NSGA2) [9] as well as the state-of-the-art Multi-Objective Covariance Matrix Adaptation Evolution Strategy (MO-CMA-ES) [10]. For the sake of completeness, we also include the gradient-based Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm with Bounds (L-BFGS-B) [11] in our comparison.

We evaluate the performances of the different approaches on the formulated MOPP problem on a large-scale, real-world, three-dimensional UAV path planning scenario in the city of New York, USA. Moreover, we show that the approaches with hybridization work more efficiently than those without advanced initialization. To the best of our knowledge, we are the first to conduct examinations on the 3D, Pareto-based MOPP problem regarding large-scale urban scenarios.

The remainder of this paper is structured as follows. In Section II, we give an overview of the related work in MOPP problems, categorize different approaches and show how they contrast with our approach. A short introduction to MOO and Non-Uniform Rational B-Splines (NURBS) follows in Section III. After that, we define the problem, objectives, and constraints as well as the advanced initialization scheme in Section IV. The validation of the initialization scheme for different optimization approaches follows in Section V. Finally, we conclude the paper and give an outlook to interesting future investigations in Section VI.

II. Related Work

Several different path planning methods have been established over the last years [12] that can, looking at utilized *optimization techniques*, be roughly distinguished into 1) classical optimization formulations like Mixed Integer Linear Programming (MILP) approaches [13], 2) sampling-based exploration like Rapidly-exploring Random Tree (RRT) approaches [14], 3) graph-based search like A* or Dijkstra approaches [15], or 4) meta-heuristic optimization, like Particle Swarm Optimization (PSO) [16], Ant-Colony Optimization (ACO) [4], or evolutionary optimization [17] approaches. From other viewpoints, path planning problems can be distinguished regarding the *UAV's environment* (rough terrain vs. urban), *formulated objectives and constraints*, the *strategy to handle multiple objectives*, the *path representation*, or the *spatial dimension* (2D vs. 3D). In Table I, a classification into the different categories is shown. For the purpose of UAV flights over rough terrain, Nikolos et al. [18] optimize a 3D B-spline path with respect to four optimization goals that are: solid boundary collision avoidance, path length minimization, guaranteeing safety distance from solid boundaries, and satisfying a minimum curvature radius. However, they aggregate the objectives and thus only perform a single-objective optimization, loosing valuable information about the particular, objective-related characteristics of a path.

Sadallah et al. [19] propose an algorithm for optimizing paths regarding travel time and obstacle avoidance. They generate a cost distribution map by applying the Fast Marching Method (FMM) on an obstacle map two times. A path is then determined on the calculated map by a gradient descent. Different Pareto solutions are achieved by varying the saturation weight for the input map. However, changing weights might in general not always lead to solely non-dominated solutions in the objective space, as shown in a recent study [7]. Moreover, the path is only two-dimensional. Especially in urban applications, adding a z-component to the paths gives the planner more flexibility in a highly complex and occluded environment.

Ghambari et al. [20] do Pareto-based multi-objective 3D path planning in an urban setting minimizing energy and maximizing the distance to obstacles. In their advanced energy model, they consider the drone to consume more energy in higher flight altitudes due to the decreasing atmospheric density. However, the applied path representation is cell-based, which can lead to sharp turns and a zigzag-shaped appearance of the path, again increasing energy inefficiency.

In our work, we want to resolve the described drawbacks of the state-of-the-art approaches by utilizing three-dimensional, generalized B-Spline curves as representation in a Pareto-based, multi-objective, evolutionary optimization that is guided by an exact preprocessing step. As far as we know, we are the first to approach this smooth 3D MOPP problem in large-scale urban scenarios.

III. Fundamentals

1) Multi-objective Optimization: MOO generally aims to find D-dimensional solutions $\mathbf{z} = \begin{bmatrix} z_1 & \dots & z_D \end{bmatrix}^T$ that minimize or maximize E objective functions

$$f_e(\mathbf{z}), \quad e = 1, \dots, E, \tag{1}$$

that are subject to F inequality constraints

$$g_f(\mathbf{z}) \geq 0, \quad f = 1, \dots, F, \tag{2}$$

as well as to G equality constraints

$$h_g(\mathbf{z}) = 0, \quad g = 1, \dots, G. \tag{3}$$

TABLE I: Classification of related work in the field of MOPP problems for UAVs.

Optimization algorithm	EA	[4], [17], [18], [20]–[25]
	ACO	[4]
	PSO	[16], [22]
	MILP	[13]
	FFM	[19]
	Gradient descent	[19]
	Dijkstra	[15]
	CFO	[26]
	RRT	[14]
UAV environment	Rough terrain	[15]–[18], [21], [22]
	Urban	[4], [19], [23], [25], [26]
Objectives & constraints	Path length	[16]–[18], [20]–[26]
	Safety, Risk	[4], [14], [16]–[21], [23]–[25]
	Energy cost	[4], [13], [20], [21], [24]
	Travel time	[4], [13], [14], [19]
	Turning angle	[16], [18], [22], [26]
	Flight height	[15]–[17], [21], [22], [26]
Multi-objective handling	Aggregate & weight	[4], [14], [16]–[18], [21], [23], [26]
	Pareto-based	[13], [19], [20], [22], [24], [25]
Path representation	Grid-based	[4], [14], [19], [20], [24]
	B-Spline	[15], [17], [18]
	Line segments	[13], [16], [21]–[23], [25], [26]
Spatial dimension	2D	[4], [14], [19], [23]
	3D	[15]–[18], [26] [20]–[22], [24]

Additionally, each element of $\mathbf{z}$ can be limited by a lower and an upper bound

$$z_d^{(L)} \leq z_d \leq z_d^{(U)}, \quad d = 1, \ldots, D. \tag{4}$$

All solutions of an optimization problem constitute the search space $\mathcal{S}$. A solution is called feasible if it satisfies every constraint and variable bound. The feasible region comprises all feasible solutions. Objective functions map a point in the search space $\mathcal{S}$ into the objective space $\mathcal{O}$. A solution that is optimal regarding all E objectives is called utopia point $\mathbf{z}_{\text{Utopia}}$. In practice, it does not exist. This is the reason to introduce the concept of Pareto-dominance. A solution $\mathbf{z}_1$ Pareto-dominates another solution $\mathbf{z}_2$ ($\mathbf{z}_1 \preccurlyeq \mathbf{z}_2$) if $\mathbf{z}_1$ is not worse than $\mathbf{z}_2$ for all objectives and $\mathbf{z}_1$ is strictly better than $\mathbf{z}_2$ in at least one objective. The set of non-dominated solutions contains every solution $\mathbf{z}$ that is not Pareto-dominated by another solution. The set that results from mapping the non-dominated set into the objective space is called the Pareto set.

2) Non-Uniform Rational B-Splines (NURBS): A NURBS curve of order $p+1$ is defined by Piegl et al. [27] as

$$\mathbf{C}(u) = \left(\sum_{i=0}^{n_p - 1} N_{i,p}(u) w_i \mathbf{P}_i \right) \left(\sum_{i=0}^{n_p - 1} N_{i,p}(u) w_i \right)^{-1}, \tag{5}$$

where

- n_p is the number of control points,
- p is the degree of the basis function $N_{i,p}$,
- $\mathbf{P}_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T$ is the i^{th} control point (assuming a 3D curve) and
- w_i is its weight.

The basis functions $N_{i,p}$ are defined with respect to the parameter u and a fixed knot vector

$$\mathbf{U} = \begin{bmatrix} u_0 & \ldots & u_m \end{bmatrix}^T,$$

containing $m+1$ knots, whereas $m = n_p + p$. The De-Boor-Cox formulas [28], [29], [30] allow to calculate the basis functions in a recursion.

IV. 3D Multi-objective Path Planning

We begin by defining the tackled optimization problem in Section IV-A. The definition of the optimization vector $\mathbf{z}$, thus the representation of the path, is introduced in Section IV-B. We formulate the utilized energy consumption and noise pollution model, which serve as objective functions, in Section IV-C, before introducing the developed safety constraints in Section IV-D. Finally, we emphasize characteristics of the developed advanced initialization in Section IV-E.

A. Problem Definition

We tackle a path planning problem in the cubic, three-dimensional space, which is given by $\mathbb{D} = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}]$. We assume a start position vector $\mathbf{x}_{\text{s}} = \mathbf{C}(a) \in \mathbb{D}$ and a goal position vector $\mathbf{x}_{\text{g}} = \mathbf{C}(b) \in \mathbb{D}$. The aim of the considered MOPP problem is to find parametric curves $\mathcal{C} = \{\mathbf{C}(u) : u \in [a, b]\}$, so that E objectives $\{f_1, \ldots, f_E\}$ are minimized. The defined vector of optimization variables $\mathbf{z}$ and the objectives are introduced below.

B. Representation

Like in a recent study [7], we use NURBS curves as a path representation. Here, we adapt them to represent 3D paths by adding a third component to the control point vectors.

Thus, the vector of optimization variables is given by

$$\mathbf{z} = \begin{bmatrix} x_1 \; y_1 \; z_1 \; \ldots \; x_{n_p-1} \; y_{n_p-1} \; z_{n_p-1} \end{bmatrix}^T. \tag{6}$$

In all experiments, we choose the hyperparameter $n_p = 20$ (number of control points) empirically to fit to the scenario size.

With the basis function degree $p = 2$, the knot vector

$$\mathbf{U} = \begin{bmatrix} 0 & 0 & \ldots & \frac{1}{n_p - p} k & \ldots & 1 & 1 \end{bmatrix}^T,$$

where $k = 0, \ldots, n_p - p$, is defined such that the curve is clamped to the first and last control point. Note that the first control point $\mathbf{x}_{\text{s}}$ and the last control point $\mathbf{x}_{\text{g}}$ are given in the problem definition and are therefore not optimized.

Moreover, during the optimization, the control point positions are bounded by the size of the design domain $x_{\min} \leq x_d \leq x_{\max}$, $y_{\min} \leq y_d \leq y_{\max}$ and $z_{\min} \leq z_d \leq z_{\max}$.

C. Objectives

In the following, we introduce the modelling of two objectives for our 3D MOPP formulation.

1) Noise: Similar to a recent study [7], this objective is designed to minimize noise pollution in the city. Torija et al. [5] show that people perceive the same level of drone noise less loud and annoying in environments with higher road traffic noise. Therefore, paths that go over urban streets are assumed to produce less additional noise pollution to the residents than paths that go over residential or recreation areas. In a recent study [7], a two-dimensional noise grid map $F_\mathrm{n} : \mathbb{N}^2 \rightarrow \mathbb{R}$ was discretized into $x_\mathrm{res} \times y_\mathrm{res}$ sized cells. Each cell contained a noise value that represented the noise pollution cost for a drone passing this specific cell. We expand the 2D grid map by a z-component with discretization z_res, yielding a three-dimensional grid array $\hat{F}_\mathrm{n} : \mathbb{N}^3 \rightarrow \mathbb{R}$. An exemplary section through the noise map at height z_min can be seen in Fig. 3d. In the following, we describe the way to calculate $\hat{F}_\mathrm{n}$ from F_n. To account for residents perceiving more noise by low flying drones than by high flying drones, we scale noise values dependent on the drone's flight height, following

$$\begin{aligned}\hat{F}_\mathrm{n}(\Delta h) &= \min\left(F_\mathrm{max}, \max\left(0, f(\Delta h)\right)\right), \quad \text{with} \qquad (7)\\ f(\Delta h) &= \frac{-F_\mathrm{max}}{z_\mathrm{max}^2}(\Delta h)^2 \mathrm{sgn}(\Delta h) + F_\mathrm{max}\end{aligned}$$

with z_max being the maximum flight height, $\Delta h = C_z(u) - z_\mathrm{min}$ being the difference between current flight height and minimum flight height and $F_\mathrm{max} = F_\mathrm{n}(C_x(u), C_y(u))$ being the 2D noise map's value at the current xy-position of the path. Thus, the noise values have their maximum at minimum flight height and decrease quadratically with increasing flight height to zero at the maximum flight height.

Finally, the noise objective is determined by calculating the line integral

$$f_\mathrm{n}(\mathbf{z}) = \int_\mathcal{C} \hat{F}_\mathrm{n}(\mathbf{C}(u)) ds = \int_a^b \hat{F}_\mathrm{n}(\mathbf{C}(u)) |\mathbf{C}'(u)| du \qquad (8)$$

along the path $\mathcal{C}$ over the three-dimensional grid array $\hat{F}_\mathrm{n}$.

2) Energy: The quadcopter flies along a three-dimensional path that can be partitioned into horizontal and vertical components. For our energy consumption model, we assume the drone's speed for vertical components to be much smaller than the constant cruise speed v_c for the horizontal components, which we set to $v_\mathrm{c} = 14\mathrm{m/s}$. Hence, time and energy for flying along a vertical segment of length L_z are considered higher than for flying along a horizontal segment of length L_{xy} of the same length. In vertical descending maneuvers, the downwash effect has to be considered. Consequently, we assume the vertical descending speed $v_\mathrm{d} = 1\mathrm{m/s}$ to be smaller than the vertical climbing speed $v_\mathrm{c} = 2\mathrm{m/s}$, thus consuming more time and energy. To conclude, in our energy model, we consider the length of the flown path and its partitioning into horizontal parts L_{xy}, as well as upward segments $L_{z,\uparrow}$ and downward segments $L_{z,\downarrow}$, leading to the energy cost function

$$f_\mathrm{e}(\mathbf{z}) = \frac{1}{2} m v_\mathrm{cruise}^2 + c_\mathrm{e}\left(L_{xy} + 10 L_{z,\uparrow} + 15 L_{z,\downarrow}\right), \qquad (9)$$

containing the mass m of a drone, which we set to $m = 1.2\mathrm{kg}$. Based on the energy consumption models by Reid [31], we determined the parameter c_e to be $c_\mathrm{e} = 9.12\mathrm{J/m}$, depending on the constant of gravitation $g = 9.81\mathrm{m/s}^2$, the density of air $\rho_\mathrm{air} = 1.225\mathrm{kg/m}^3$, the drone mass m, and some other drone parameters that we exemplary set to $c_\mathrm{d} = 0.6$ (drag coefficient), $A_\mathrm{d} = 0.1\mathrm{m}^2$ (drag area), $n_\mathrm{r} = 4$ (number of rotors) and $r_\mathrm{r} = 0.1\mathrm{m}$ (radius of a rotor).

D. Constraints

To be able to consider constraints $g_f(\mathbf{z})$ during the optimization, we add them as so-called soft constraints to one of the objective functions (in our case to the energy cost function $f_\mathrm{e}(\mathbf{z})$), yielding the constrained energy cost function

$$f_\mathrm{e,C}(\mathbf{z}) = f_\mathrm{e}(\mathbf{z}) + \xi\sqrt{H(-g_f(\mathbf{z}))g_f(\mathbf{z})^2}, \qquad (10)$$

with H being the Heaviside function and ξ being an arbitrary large number, ensuring that $f_\mathrm{e,C}(\mathbf{z}) \gg f_\mathrm{e}(\mathbf{z})$ when the constraint is violated.

1) Minimum Flight Height: Except for the start and landing sequence, the quadcopter is supposed to fly above a minimum flight height z_min. We ensure this by the inequality constraint

$$g_1(\mathbf{z}) = \sum_{i=1}^{n_p - 1} (z_i - z_\mathrm{min}) \geq 0, \qquad (11)$$

with z_i being the z-coordinate of the i^th control point.

2) Static Obstacle Collision Avoidance: For safe maneuvers in cities, the drone's path must either overfly or circuit static obstacles like buildings. We introduce a twofold collision avoidance constraint $g_2(\mathbf{z}) = g_\mathrm{overfly}(\mathbf{z}) + g_\mathrm{circuit}(\mathbf{z})$. It allows the optimizer to make use of both actions (overflying and circuiting) to adapt paths that lead through obstacles.

- To allow overflying, we utilize the city's height map $F_\mathrm{h} : \mathbb{N}^2 \rightarrow \mathbb{R}$. Each cell of this grid map is assigned the height of the tallest building located in the respective cell. The overfly inequality constraint then appears to

  $$g_\mathrm{overfly}(\mathbf{z}) = C_z(u) - F_\mathrm{h}(C_x(u), C_y(u)) \geq 0, \qquad (12)$$

  punishing all dangerous waypoints $\mathbf{C}_{danger}(u)$ of the path $\mathbf{C}(u)$ that lie beneath the height of the respective height map entry $F_\mathrm{h}(C_x(u), C_y(u))$. Additionally, we save all dangerous waypoints for the next calculation.
- A better choice to avoid tall buildings during path planning might be to fly around instead of over the obstacle. So far, we have only created a drift for the optimizer to fly over buildings. For the drift to fly around buildings, we introduce an obstacle grid map $F_\mathrm{o} : \mathbb{N}^2 \rightarrow \mathbb{R}$. The obstacle grid map is a discrete potential field with zero potential at free cells and a non-zero potential at occupied areas. The potential depends on the distance to the nearest free space and is expressed by the number of separating cells. With this, we have

  $$g_\mathrm{circuit}(\mathbf{z}) = -F_\mathrm{o}(C_{\mathrm{danger},x}, C_{\mathrm{danger},y}) \geq 0, \qquad (13)$$

  as an inequality constraint that creates a drift of dangerous waypoints along the horizontal plane in the direction towards free space areas.

E. Advanced Initialization

1) Extension to 3D: The contribution of this paper is to show that the hybrid approach for 2D Multi-Objective (MO) path planning [7] can also be applied to the 3D case. The main idea of the hybrid approach is to find good initial solutions for each objective in a preprocessing step by the use of the exact, graph-based (but not MO-suitable) bidirectional Dijkstra solver. The good initial solutions are then used as start vectors in a MO-suitable algorithm. In this way, a more diversified Pareto front with higher Hypervolume can be obtained faster. However, in the 2D hybrid approach [7] the calculation of all objectives was based on grid maps. By this means, the preprocessing could be done by the following steps. First the *grid was converted to a graph* (S1), then the initial solutions were calculated for every *objective individually* (S2A) and finally *for different weighted sums of the objectives* (S2B) by weighting, aggregating and normalizing the grid maps.

The energy objective (Eqn. (9)) in this paper is not grid-based and therefore not suited for the described steps (S1) and (S2B). Thus, for the graph calculation of the energy objective, instead of step (S1), we generate an initial 3D graph that is consistent with the energy model (9) used in this paper. An exemplary graph section is visualized in Fig. 1. Furthermore, step (S2B) is omitted in this paper.

Note that we can integrate static obstacles and no-fly zones in the preprocessing step by removing nodes from the grid graph that belong to cells with $F_\mathrm{o} > 0$ in the obstacle grid map.

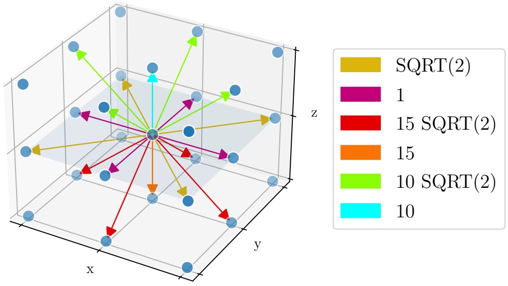


Fig. 1: According to the energy model (9), the costs for travelling along an edge in the energy graph depends on the movement direction of the agent.

2) Challenges: When working with graph-based shortest path algorithms, their performance depends on the number of nodes and edges in the graph. As it can be seen from Fig. 2, the transition from 2D to 3D graphs comes at the cost of a massively increasing number of edges. Thus, it is not practical to use the same resolution x_res, y_res and z_res for the grid graph in the preprocessing step as it is used in the main optimization. Instead, we rescale the grid graph to a resolution $\tilde{x}_\mathrm{res}$, $\tilde{y}_\mathrm{res}$ and $\tilde{z}_\mathrm{res}$, respectively. Rescaling is crucial as 1) the preprocessing is still too slow or consumes too much computing power if the resolution is too fine, and 2) the found solutions are too poor to serve as adequate initial solutions for the main optimization if the resolution is too coarse. We empirically found $\begin{bmatrix}\tilde{x}_\mathrm{res} & \tilde{y}_\mathrm{res} & \tilde{z}_\mathrm{res}\end{bmatrix} = \begin{bmatrix}15 & 15 & 10\end{bmatrix}$ to achieve a good compromise.

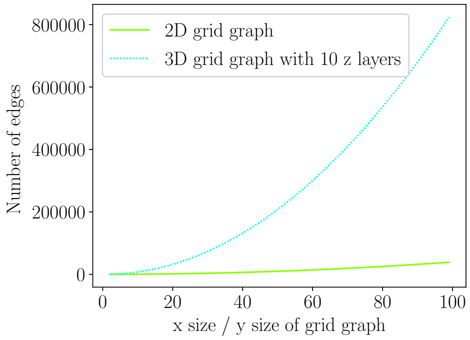


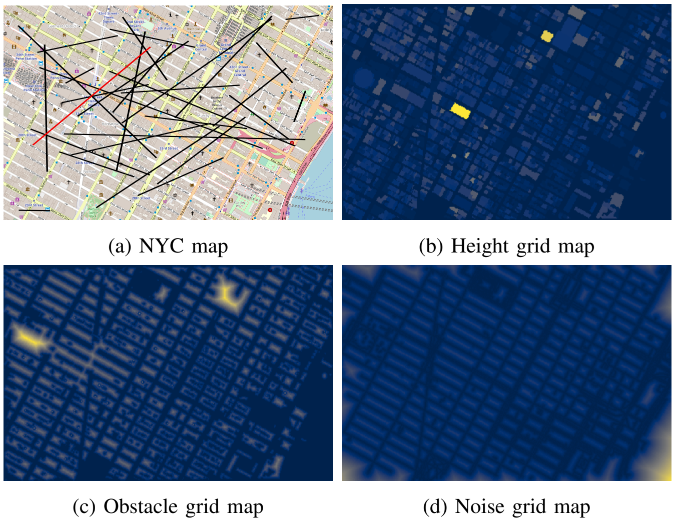
Fig. 2: Comparison of the number of edges in a 2D (8-neighborhood) and a 3D (18-neighborhood) square grid graph with $z = 10$ layers.

V. Evaluation

The urban scenario and the solvers that are used for the evaluation are introduced in Section V-A and V-B, respectively. After showing the parameters in Section V-C and the utilized metrics for evaluation in Section V-D, we present the results of the experiments in Section V-E.

A. Scenario

The evaluation is based on OpenStreetMap (OSM) [32] data of a map section from New York City (NYC) that is visualized in Fig. 3a. Visualizations of the derived height grid map F_h, the obstacle grid map F_o, and a cross section through the noise grid array $\hat{F}_\mathrm{n}$ at height $z = z_\mathrm{min}$ can be seen in Fig. 3b, Fig. 3c, and Fig. 3d, respectively.



(a) NYC map

(b) Height grid map

(c) Obstacle grid map

(d) Noise grid map

Fig. 3: Used (grid) maps of the NYC scenario. In (a), the lines indicate the start and end points of routes for the evaluation. For the scenario indicated by the red line, 3D plots of the paths are given further below. In (b-d), grid cells with lighter colors indicate a higher height/obstacle collision severity/noise annoyance value.

B. Solvers

To solve the 3D MOPP, we evaluate different optimizers. Next to a straightforward implementation of an Evolution Strategy (ES) [7], we choose two state-of-the-art multi-objective, evolutionary algorithms, 1) the NSGA2 (Non-dominated Sorting Genetic Algorithm) [9] as well as 2) the MO-CMA-ES (Multi-Objective Covariance Matrix Adaptation Evolution Strategy) [10].

Furthermore, we also solve the problem with the gradient-based optimizer L-BFGS-B [11], a variant of the widely-used L-BFGS algorithm [33], which is capable of incorporating variable bounds. Please note that the L-BFGS-B algorithm is not capable of handling more than one objective at once. Therefore, to maintain comparability, we use a weighted aggregation approach in order to calculate a Pareto front. We calculate the necessary weights according to an Adaptive Weight Determination Scheme (AWDS) [34].

To show the advantage of the hybrid approach, each solver is tested both with and without the advanced initialization step. In the following, the hybrid algorithms are denoted with the H. prefix.

C. Parameters

In Table II, we give an overview of the most important parameters for each approach. Parameters that were not introduced by us are set to standard values from literature [9], [10]. We compare the solvers for 30 different routes in the described urban scenario. We choose the start and end positions $\mathbf{x}_\mathrm{s}$ and $\mathbf{x}_\mathrm{g}$ of the routes with the same approach like in a recent study [7], to obtain a representative set that is evenly distributed in the scenario space as well as in route distances. The derived start and end positions are visualized as straight lines in Fig. 3a.

TABLE II: Setup of the optimization methods used in the paper.

	Parameter	**Symbol**	**Value**
General	map dimensions	$[x_\mathrm{min}\ x_\mathrm{max}]$	$[0 \quad 2280]$
		$[y_\mathrm{min}\ y_\mathrm{max}]$	$[0 \quad 1500]$
	air space limits	$[z_\mathrm{min}\ z_\mathrm{max}]$	$[50 \quad 300]$
	discretization step	$[x_\mathrm{res}\ y_\mathrm{res}\ z_\mathrm{res}]$	$[4\ 4\ 10]$
	# of control points	n_p	20
	deg. (basis function)	p	2
Method			
CMA-ES	init. step size	σ	1
	# of parents	μ	100
	# of offspring	λ	100
NSGA2	crowding degree	$\eta_\mathrm{crossover}$	20
	crossover prob.	$p_\mathrm{crossover}$	0.9
	crowding degree	η_mutation	20
	mutation prob.	p_mutation	$1/D$
	population size	N	100
ES	init. step size	$[\sigma_{x,0}\ \sigma_{y,0}\ \sigma_{w,0}]$	$[3\ 3\ 5]$
	# of parents	μ	10
	# of offspring	λ	100

D. Metrics

To compare the optimizers, we use different metrics, which are shortly described here. The interested reader is referred to the detailed explanations and formulas in another study [7].

- The Hypervolume (HV) is the area in the objective space enclosed by all non-dominated solutions and a defined reference point, which we set to $\mathbf{p}_\mathrm{HV,ref} = \begin{bmatrix}10 & 10\end{bmatrix}^T$. To be able to compare solvers across different scenarios, we normalize the achieved HV of all solvers in one scenario to the best and the worst HV in this scenario.
- The lower the generational distance (GD) metric of a Pareto front, the closer are the solutions of the examined front to a reference front. Thus, GD is a metric for the convergence of a Pareto front.
- The lower the inverted Generational Distance (IGD) metric of a Pareto front, the closer are the solutions of a reference front to the examined front. Thus, IGD is a metric for both, convergence and diversity of a Pareto front.

Due to the lack of optimal Pareto fronts, in every scenario we choose the front with the highest HV as the reference front to calculate the GD and IGD metrics.

E. Results

During the optimization using evolutionary and gradient-based solvers, the cost function evaluations need the most computational resources in each iteration. Accordingly, for a fair basis of comparison, we limit the number of objective function evaluations to $n_\mathrm{fun,eval} = 40000$ for all optimizers, which corresponds to an evolution of 400 generations for the evolutionary approaches.

The obtained results can be seen in Table III. We explicitly show the optimizer's normalized HV, GD, and IGD metric after 1000 and 40000 function evaluations, respectively. Additionally, Fig. 4 gives a more detailed look on the development of the normalized HV, which is averaged over all runs, in dependence on the increasing number of function evaluations. The results clearly indicate the weakness of the standard BFGS approach, resulting in a final normalized HV of 31%. The reason for that could be the multimodal character of the objective functions. The L-BFGS-B solver pushes the paths into local optima and then terminates. By the use of the advanced initialization step, the performance of the BFGS algorithm can be increased significantly[1] by 129%.

The same effect can be observed for the other optimizers: by utilizing the hybrid approach, the solvers' achieved final normalized HVs can be increased by 41% (CMA-ES), 26% (NSGA2), and 17% (ES), respectively. The overall best results can be achieved by the H. NSGA2 algorithm. Having a look at the development of its normalized HV in Fig. 4 (orange curve) in comparison to NSGA2 without advanced initialization (red curve), we can observe that the hybridization has the biggest impact in the beginning of the optimization, leading to a huge difference in the initial normalized HV of both solvers. In order to further examine characteristics of the solvers after only a few function evaluations, we show the boxplot for

[1] Wilcoxon two-tailed rank-sum test: $n = 30$, $P < 0.05$, $M_\mathrm{Norm.HV,BFGS} = 0\%$, $M_\mathrm{Norm.HV,H.BFGS} = 75\%$, $U_\mathrm{BFGS,H.BFGS} = 822$.

the normalized HV after 1000 function evaluations in Fig. 5. Regarding the normalized HV, the NSGA2 and H. NSGA2 algorithms differ significantly[2].

All hybrid evolutionary approaches show the same beneficial characteristic regarding the GD metric. By looking in Table III, we can observe that after 1000 evaluations the hybrid evolutionary optimizers achieve a smaller and therefore better generational distance measure than their standard counterparts. Again, the best performing algorithm is H. NSGA2 that performs significantly[3] better than the standard NSGA2 algorithm. Finally, we want to have a look at the performances regarding the IGD characteristic. From Table III, we can observe that after 1000 evaluations the H. ES approach achieves the best mean IGD value, differing significantly[4] from the standard ES.

We look at the paths that were calculated after 1000 function evaluations by each optimizer for the scenario that is depicted by the red line in Fig. 3a. In Fig. 6, we can see the paths with the minimal noise values. Whereas the non-hybrid optimizers (blue, red, brown and gray paths) only find solutions at lower altitude, the hybrid optimizers found paths (green, orange, violet, and yellow) with lower noise values at higher flight altitudes. Similarly, the hybrid approaches also benefit from the preprocessing step regarding the calculated paths that achieved the lowest energy consumption values. They are visualized in Fig. 7. It can be seen that the paths that were generated by the standard optimizers still lead through static obstacles, whereas the paths that were generated by the hybrid algorithms are already nearly collision-free.

TABLE III: Mean results for the evaluation on 30 scenarios.

	Norm.HV		**GD** $(\times 10^2)$		**IGD** $(\times 10^2)$	
Evaluations	1000	40000	1000	40000	1000	40000
H. CMA-ES	75%	96%	1.4	1.6	0.7	**1.0**
CMA-ES	23%	68%	4.2	2.5	4.3	2.7
H. NSGA2	75%	**98%**	**0.3**	**0.3**	2.5	1.9
NSGA2	37%	78%	1.0	0.6	5.6	3.9
H. BFGS	58%	71%	1.3	3.0	3.5	3.8
BFGS	13%	31%	0.6	2.0	6.5	5.9
H. ES	**80%**	90%	0.9	1.3	**0.2**	1.1
ES	36%	77%	4.1	1.3	4.3	2.3

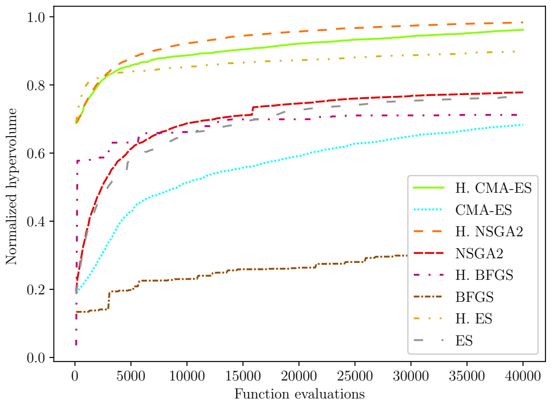


Fig. 4: For each scenario, the optimizers' achieved hypervolumes are normalized to the best and worst achieved hypervolumes. The normalized hypervolumes are averaged over all 30 scenarios and shown dependent on the number of cost function evaluations.

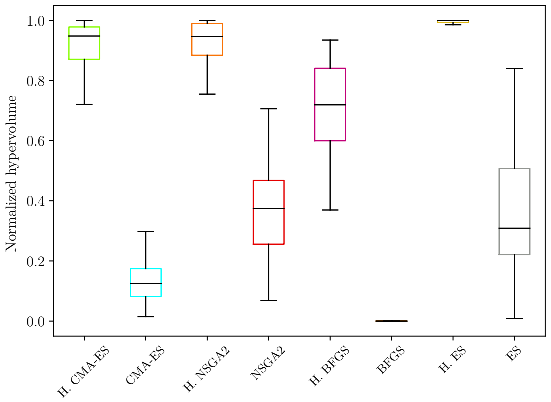


Fig. 5: Boxplots for the normalized HV after 1000 iterations for 8 optimizers on 30 scenarios. One box extends from the lower to the upper quartile of the data. The black line indicates the median. The whiskers show the range of the data.

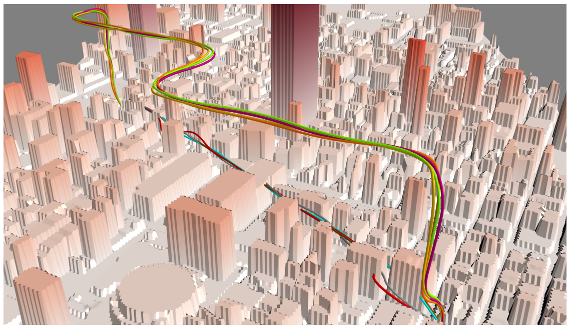


Fig. 6: Examples of the best paths calculated by the optimizers for the minimum noise value in one scenario.

VI. Conclusion & Outlook

In this work, we formulated a three-dimensional Multi-Objective Path Planning (MOPP) problem in an urban scenario setting. Paths were optimized regarding two objectives, 1) a height-dependent noise cost function, and 2) an energy cost model that includes safety constraints for adherence to a minimal flight height and static obstacle collision avoidance. We introduced an advanced preprocessing scheme that is able to calculate good initial paths in the three-dimensional space. The initialization scheme was adapted to several state-of-the-art evolutionary and gradient-based optimizers, called hybrid solvers. Based on several instances of a real-world scenario for Unmanned Aerial Vehicles (UAVs) in New York City, the hybrid solvers were benchmarked against their non-hybrid versions. The statistical results of the comparison revealed the advantages of the hybrid approaches over the standard approaches for all examined metrics. The hybrid approaches allow the user to find better non-dominated solutions within

[2] Wilcoxon two-tailed rank-sum test: $n = 30$, $P < 0.05$, $M_{\text{Norm.HV,NSGA2}} = 37\%$, $M_{\text{Norm.HV,H.NSGA2}} = 95\%$, $U_{\text{NSGA2,H.NSGA2}} = 843$.

[3] Wilcoxon two-tailed rank-sum test: $n = 30$, $P < 0.05$, $M_{\text{GD,NSGA2}} = 0.009$, $M_{\text{GD,H.NSGA2}} = 0.003$, $U_{\text{NSGA2,H.NSGA2}} = 155$.

[4] Wilcoxon two-tailed rank-sum test: $n = 30$, $P < 0.05$, $M_{\text{IGD,ES}} = 0.047$, $M_{\text{IGD,H.ES}} = 0.0$, $U_{\text{ES,H.ES}} = 7.5$.

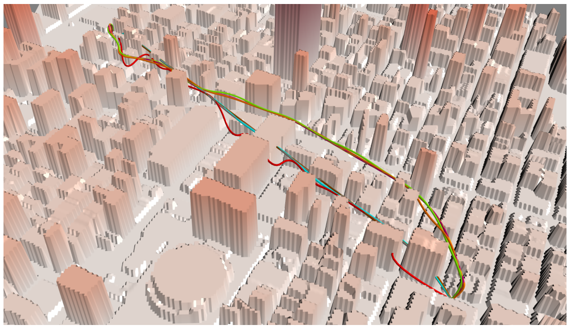


Fig. 7: Examples of the best paths calculated by the optimizers for the minimum energy value in one scenario.

less function evaluations. This is a significant advantage in real-world applications, where usually the computational resources are limited.

In future studies, we would like to extend the formulated 3D MOPP to a Multi-objective Multiple Path Planning problem (MOMPP) that is able to plan the paths for multiple interacting agents concurrently, avoiding collisions between them.

References

[1] P. Lottes, R. Khanna, J. Pfeifer, R. Siegwart, and C. Stachniss, "UAV-based crop and weed classification for smart farming," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3024–3031.

[2] C. Deng, S. Wang, Z. Huang, Z. Tan, and J. Liu, "Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications." *J. Commun.*, vol. 9, no. 9, pp. 687–692, 2014.

[3] M. Roca-Riu and M. Menendez, "Logistic deliveries with drones: State of the art of practice and research," in *19th Swiss Transport Research Conference (STRC 2019)*. STRC, 2019.

[4] Q. Yang and S.-J. Yoo, "Optimal UAV path planning: Sensing data acquisition over IoT sensor networks using multi-objective bio-inspired algorithms," *IEEE Access*, vol. 6, pp. 13 671–13 684, 2018.

[5] A. J. Torija, Z. Li, and R. H. Self, "Effects of a hovering unmanned aerial vehicle on urban soundscapes perception," *Transportation Research Part D: Transport and Environment*, vol. 78, p. 102195, 2020.

[6] H. K. Singh, T. Ray, T. Rodemann, and M. Olhofer, "Identifying solutions of interest for practical many-objective problems using recursive expected marginal utility," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2019, pp. 1734–1741.

[7] N. Hohmann, M. Bujny, J. Adamy, and M. Olhofer, "Hybrid evolutionary approach to multi-objective path planning for UAVs," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2021, pp. 1–8.

[8] E. W. Dijkstra *et al.*, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[10] T. Voß, N. Hansen, and C. Igel, "Improved step size adaptation for the MO-CMA-ES," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, 2010, pp. 487–494.

[11] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on scientific computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[12] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, and Y. Xia, "Survey of robot 3D path planning algorithms," *Journal of Control Science and Engineering*, vol. 2016, 2016.

[13] Y. Hao, B. Li, L. Shao, Y. Zhang, and J. Cui, "Multi-objective path planning for unmanned aerial vehicle based on mixed integer programming," in *2017 Chinese Automation Congress (CAC)*. IEEE, 2017, pp. 7035–7039.

[14] S. Primatesta, L. S. Cuomo, G. Guglieri, and A. Rizzo, "An innovative algorithm to estimate risk optimum path for unmanned aerial vehicles in urban environments," *Transportation research procedia*, vol. 35, pp. 44–53, 2018.

[15] L. Babel, "Three-dimensional route planning for unmanned aerial vehicles in a risk environment," *Journal of Intelligent & Robotic Systems*, vol. 71, no. 2, pp. 255–269, 2013.

[16] Y. Fu, M. Ding, and C. Zhou, "Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 2, pp. 511–526, 2011.

[17] I. Hasircioglu, H. R. Topcuoglu, and M. Ermis, "3-D path planning for the navigation of unmanned aerial vehicles by using evolutionary algorithms," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, 2008, pp. 1499–1506.

[18] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, "Evolutionary algorithm based offline/online path planner for UAV navigation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 6, pp. 898–912, 2003.

[19] N. Sadallah, S. Yahiaoui, A. Bendjoudi, and N. Nouali-Taboudjemat, "Multi-objective offline and online path planning for UAVs under dynamic urban environment," *International Journal of Intelligent Robotics and Applications*, pp. 1–20, 2021.

[20] S. Ghambari, M. Golabi, J. Lepagnot, M. Brévilliers, L. Jourdan, and L. Idoumghar, "An enhanced NSGA-II for multiobjective UAV path planning in urban environments," in *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2020, pp. 106–111.

[21] V. Roberge, M. Tarbouchi, and G. Labonté, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Transactions on industrial informatics*, vol. 9, no. 1, pp. 132–141, 2012.

[22] X. Zhen, Z. Enze, and C. Qingwei, "Rotary unmanned aerial vehicles path planning in rough terrain based on multi-objective particle swarm optimization," *Journal of Systems Engineering and Electronics*, vol. 31, no. 1, pp. 130–141, 2020.

[23] Q. Ren, Y. Yao, G. Yang, and X. Zhou, "Multi-objective path planning for UAV in the urban environment based on CDNSGA-II," in *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE, 2019, pp. 350–3505.

[24] M. Golabi, S. Ghambari, J. Lepagnot, L. Jourdan, M. Brévilliers, and L. Idoumghar, "Bypassing or flying above the obstacles? A novel multi-objective UAV path planning problem," in *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2020, pp. 1–8.

[25] J. Rubio-Hervas, A. Gupta, and Y.-S. Ong, "Data-driven risk assessment and multicriteria optimization of UAV operations," *Aerospace Science and Technology*, vol. 77, pp. 510–523, 2018.

[26] Y. Chen, J. Yu, Y. Mei, Y. Wang, and X. Su, "Modified central force optimization (MCFO) algorithm for 3D UAV path planning," *Neurocomputing*, vol. 171, pp. 878–888, 2016.

[27] L. Piegl and W. Tiller, *The NURBS book*. Springer Science & Business Media, 1996.

[28] M. G. Cox, "The numerical evaluation of B-splines," *IMA Journal of Applied Mathematics*, vol. 10, no. 2, pp. 134–149, 1972.

[29] C. De Boor, "On calculating with B-splines," *Journal of Approximation theory*, vol. 6, no. 1, pp. 50–62, 1972.

[30] C. De Boor and C. De Boor, *A practical guide to splines*. springer-verlag New York, 1978, vol. 27.

[31] J. S. Reid, "Drone flight — what does basic physics say?" https://homepages.abdn.ac.uk/nph120/meteo/DroneFlight.pdf, 2020, [Online; accessed February 11, 2022].

[32] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org, 2017.

[33] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.

[34] N. Ryu and S. Min, "Multiobjective optimization with an adaptive weight determination scheme using the concept of hyperplane," *International Journal for Numerical Methods in Engineering*, vol. 118, no. 6, pp. 303–319, 2019.