

Three-Dimensional Urban Path Planning for Aerial Vehicles Regarding Many Objectives

NIKOLAS HOHMANN¹, SEBASTIAN BRULIN², JÜRGEN ADAMY¹, AND MARKUS OLHOFFER²

¹Department of Control Methods and Intelligent Systems, Technical University of Darmstadt, 64283 Darmstadt, Germany

²Complex Systems Optimisation and Analysis Group, Honda Research Institute Europe, 63073 Offenbach, Germany

CORRESPONDING AUTHOR: N. HOHMANN (e-mail: nikolas.hohmann@tu-darmstadt.de)

This work was supported in part by the Deutsche Forschungsgemeinschaft (DFG)-German Research Foundation and in part by the Open Access Publishing Fund of Technical University of Darmstadt.

ABSTRACT Planning flight paths for unmanned aerial vehicles in urban areas requires consideration of safety, legal, and economic aspects as well as attention to social factors for gaining public acceptance. To solve this many-objective path planning problem in the three-dimensional space, we propose a hybrid framework combining an exact Dijkstra search and a metaheuristic evolutionary optimization. Given a start and an endpoint, we optimize a path regarding the risk in case of a system failure, the radio signal disturbance between the aerial vehicle and a ground station, the energy consumption, and the noise immission on city residents. The optimization includes constraints for static obstacle collision avoidance and compliance with the minimum flight altitude. The result is a set of smooth and three-dimensional paths that realize different trade-offs between the defined objectives. As an example, we consider an urban transportation application for aerial vehicles in San Francisco. For all tests, we use real-world data from OpenStreetMap. In a statistical evaluation, we test the efficiency of our framework against different state-of-the-art optimizers. Moreover, we extend the framework with two features that allow the user to integrate arbitrary objectives and unknown scenarios into the path planning framework.

INDEX TERMS Dijkstra, evolutionary algorithm, hybrid algorithm, many-objective, optimization, path planning, three-dimensional, transportation, UAM, unmanned aerial vehicle (UAV), urban, urban air mobility.

I. INTRODUCTION

SCIENTISTS have identified last-mile delivery as a bottleneck in modern transportation systems [1]. Especially in congested urban areas, delivering packages or passengers from an intermediate stop to their final destination (i.e., last-mile problem) is resource intensive, and costly. Therefore, the problem has spawned a variety of different optimization approaches [2] and ideas for new modes of transportation like cargo bikes [3], and driverless shuttles [4]. Unmanned Aerial Vehicles (UAVs) have sparked the interest of companies in logistics [5] and mobility [6] as so-called Unmanned Aircraft Systems (UAS) [7], [8] open a whole new dimension of local transportation possibilities. Start-ups like Wingcopter [9] are already utilizing UAVs

to carry out important or urgent deliveries. The market for UAV deliveries is expected to increase with a growth rate of approximately 20% in the next 12 years to a market value of 4 billion U.S.\$ and the number of delivery UAVs is rising rapidly from 7 thousand in 2020 to assumed 125 thousand in 2035 [6]. Today, the prevalent modes of operation for aircraft are *airspace-based operations* in controlled airspace and *free flight operations* within the visual line of sight in uncontrolled airspace [10]. However, with the anticipated densities of UAV flights in urban airspaces, both concepts are likely to become inefficient, unsafe, or even infeasible [11]. Alternative approaches propose some form of structural traffic regulation for aerial near-ground operations in urban areas, e.g., flight corridors [12], [13]. This motivates the question about optimal flight corridor placement throughout the city. Especially in cities, where many different interest groups meet, the design of a new transportation

The review of this article was arranged by Associate Editor Francesco Corman.

system has to satisfy many demands, e.g., the legal and safety requirements of the aviation authority, the economic interests of logistics companies, and the social factors among the city residents. Bauranov and Rakas [11] point out that the consideration of social factors like noise pollution has been neglected in many recent approaches to urban air mobility, even though it is one of the most crucial factors for scaling an urban air mobility application. Quite often, the different demands on a transportation system conflict, which calls for some kind of trade-off solutions.

We propose a path planning framework that is capable to optimize three-dimensional smooth paths regarding all given objectives. Exemplarily, we optimize the paths regarding radio signal connection, the energy consumption of the UAVs, as well as the risk and noise that residents are exposed to. Though, arbitrary objectives can be included. We improve the optimization in our framework with a hybrid approach that uses the Dijkstra algorithm [14] and an evolutionary optimization algorithm [15].

The Dijkstra algorithm efficiently finds the guaranteed-optimal path regarding one single objective but is designed to search in graphs (e.g., grids), which limits the size of the search space and possible path representations. Evolutionary algorithms allow any kind of path representation and are particularly suitable to identify good trade-off solutions in many-objective optimization problems, but need high computational resources.

In this work, we show the advantages of the combination of both approaches. The Dijkstra algorithm computes solutions for all objectives that can be expressed as a grid or a graph respectively. Those solutions are smoothed and approximated and serve as good initial solutions in the many-objective evolutionary optimization. Moreover, we introduce two features that improve the search for good trade-off paths in a city. We test the framework's functionality using San Francisco as an example, but we can apply the approach to any area.

The distinctiveness of our framework compared to other approaches, presented in Section II, lies in the efficient planning of three-dimensional smooth paths for UAVs. For an arbitrary region, the user can define criteria, which could come from different stakeholders, and obtain a set of paths with different weighting for the criteria.

The remainder of this article is structured as follows. In Section II, we give an overview of current research in the field of many-objective path planning for UAVs and how our work differs from it. We describe the system model in Section III before introducing our proposed framework in Section IV. Then, we evaluate and discuss the framework in Section V before concluding in Section VI.

II. RELATED WORK

We have identified five criteria (C1 - C5) by which many-objective path planning approaches for UAVs can be classified. First, there is the *UAV's environment* (C1), which can be rough terrain with few to no people being

present, or an urban environment. Second, the *spatial dimension* (C2) for the UAV path planning is either simplified to a two-dimensional space of representation or set to a complete three-dimensional path representation. Third, the chosen *path representations* (C3) can be defined differently. Adjacent line segments are commonly used, as are grid-based representations. Alternatively, researchers use spline or polynomial functions. Fourth, each approach presents different *formulated objectives and constraints* (C4) ranging from path length, energy consumption, and travel time to safety and risk-related objectives and turning angle or flight height optimization. Fifth, the *strategy to handle multiple objectives* (C5) can differ. Many approaches either propose a weighted aggregation of multiple objectives to obtain a single-objective optimization problem, or they choose a Pareto-based approach. In the latter, a set of solutions is evaluated regarding all objectives. Solutions that are not dominated by other solutions build the so-called Pareto set, which approximates the Pareto front. We refer the interested reader to an introduction on multi-objective optimization [16].

One early approach for many-objective path planning was proposed by Nikolos et al. [17], who optimize UAV paths for flights over rough terrain (C1). They use a three-dimensional (C2) B-spline path representation (C3) and optimize the paths concerning four optimization goals (C4), which are collision avoidance, path length minimization, safety distance assurance, and compliance with a minimum curvature radius. However, during the evolutionary optimization, the function values for all objectives are aggregated to simplify the problem to a single-objective optimization (C5). This means that the approach is dependent on the weighting of the objectives and therefore limited to finding only one solution within the Pareto set of possible solutions. In our approach, we want to avoid this problem by using a meta-heuristic optimization approach known for its effective sampling of a Pareto set.

In a later approach, Rubio-Hervas et al. [18] adopt an urban setting in Singapore (C1) to plan 3D (C2) paths for UAVs. They propose a special path representation (C3) that is composed of the distance and the angle between waypoints and a straight line connecting the start and the goal point of the path. To optimize a path regarding its length and risk (C4), they make use of the NSGA-II algorithm [19], which is a state-of-the-art evolutionary algorithm for multi-objective optimization capable of calculating a Pareto set (C5). We adopt the approach of using a meta-heuristic optimizer for path planning. However, at the same time, we are aware of its need for large computational resources and want to address this problem with the hybrid approach presented below.

For an urban setting (C1), Ghambari et al. [20] propose a 3D path planning (C2) approach that adopts a cell-based path representation (C3) on a grid. The objectives are to minimize the UAV's energy consumption and to maximize the distance to obstacles (C4). They propose a detailed energy model that assumes a larger energy consumption for higher flight

altitudes due to the decreasing atmospheric density. To solve the many-objective path planning problem, Ghambari et al. utilize different evolutionary algorithms (C5). The use of a grid-based path representation can result in sharp turns and inefficient jagged paths, which is one reason why we decided on a smooth spline curve representation for our approach.

In a recent study, Sadallah et al. [21] present an urban environment (C1) to do path planning of two-dimensional (C2) paths concerning travel time and obstacle avoidance (C4). In their approach, a cost distribution map is computed by applying a fast marching method on an obstacle map two times. Then, a gradient descent operation calculates the path (C3). The variation of the saturation weight for the input map assures the calculation of different Pareto solutions (C5). Nevertheless, changing the weights of a weighted grid map does not necessarily result in non-dominated solutions in the objective space, as we showed in another study [22]. Moreover, the utilized path representation is two-dimensional, which may lead to unsolvable problems in complex and highly occluded urban environments, so we decided to use a three-dimensional path representation.

The conclusions that we have drawn from the presented studies are all incorporated into the design of our framework for solving a many-objective path planning problem in a city environment (C1). The utilized three-dimensional (C2) spline representation (C3) provides suitable smooth paths for UAVs, which therefore do not require post-smoothing. The urban setting not only requires attention to energy minimization and radio link optimization but also to risk and noise minimization (C4). A many-objective evolutionary algorithm (C5) generates a Pareto set of differently balanced paths. The evolutionary search is initialized with solutions from a pre-processing step (C5) that is supposed to increase the chances of finding a global optimum in the multimodal optimization landscape. To our knowledge, we are the first to present a direct-search-accelerated, evolutionary, many-objective 3D path planning framework for arbitrary objectives to address the needs of different stakeholders.

III. SYSTEM MODEL

In this section, we define the tackled optimization problem in Section III-A, and present the composition of the optimization vector in Section III-B. After that, we introduce the objectives of the optimization problem in Section III-C and the established constraints in Section III-D.

A. PROBLEM DEFINITION

Let us assume a continuous and cubic space $\mathbb{D} = [x_{\min} x_{\max}] \times [y_{\min} y_{\max}] \times [z_{\min} z_{\max}]$. Given a start point $\mathbf{x}_s \in \mathbb{D}$ and a goal point $\mathbf{x}_g \in \mathbb{D}$, we define a path to be a sequence of three-dimensional points $\Pi = [\boldsymbol{\pi}_0 = \mathbf{x}_s, \boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_g = \mathbf{x}_g]$. Our goal is to find a set of paths that approximate the Pareto Front of a many-objective optimization problem, which consists of $E = 4$ objectives and $F = 3$ inequality constraints.

B. REPRESENTATION

To represent the introduced path Π in our optimization problem, we utilize parametrized curves called Non-Uniform Rational B-Splines (NURBS). Adopting the notion of Piegl and Tiller [23], we denote the curve parameter u and the curve $\mathbf{C}(u)$. We set the curve's degree to a fixed value of $p = 2$ and also use a fixed knot vector. The curve's shape is then solely affected by so-called control points $\mathbf{P}_i = [x_i \ y_i \ z_i]^T$ and their weights w_i with $i \in \{0, \dots, n_p - 1\}$. The number of control points n_p is a hyperparameter that is either set by the user or automatically determined by an algorithm, which we present in Section IV-B. The weights are fixed to $w_i = 1$ with an exception that we describe together with the constraint formulation in Section III-D. We use a clamped uniform knot vector [24] ensuring that the curve is clamped to the first and last control point, which equal the start $\mathbf{P}_0 = \mathbf{x}_s$ and the endpoint $\mathbf{P}_{n_p-1} = \mathbf{x}_g$ of the path. The remaining $n_p - 2$ control points $\mathbf{P}_i = [x_i \ y_i \ z_i]$ are the variables

$$\mathbf{z} = [x_1 \ y_1 \ z_1 \ \dots \ x_{n_p-2} \ y_{n_p-2} \ z_{n_p-2}]^T \quad (1)$$

of our optimization problem. Descriptively, the optimizer changes the positions of control points and thus the curve's shape. Finally, we obtain the path Π by evaluating the curve $\mathbf{C}(u)$

$$\Pi(\mathbf{z}) = \mathbf{C}(u) \Big|_{[\mathbf{P}_1 \ \dots \ \mathbf{P}_{n_p-2}] = \mathbf{z}}, \quad (2)$$

in such a way that the resulting path points $\boldsymbol{\pi} \in \Pi \subset \mathbb{D}$ are spatially equidistant. The used NURBS curve representation allows paths of different shapes and lengths without changing the size of the optimization vector and thus the complexity of the optimization problem.

C. OBJECTIVES

The objectives for evaluating flight paths may change over time or depend on a city's specific requirements. Therefore, we have designed our framework to incorporate additional objectives in the future. Below, we specify exemplary evaluation procedures for four objectives to demonstrate and evaluate the presented framework. These procedures are realistic but simplified. Sophisticated commercial tools are available for all objectives, which can be integrated into the overall framework as block-box simulations if necessary. We distinguish between grid-based and non-grid-based objectives. Grid-based objectives are calculated on a three-dimensional discrete version \mathbb{D}_d of \mathbb{D} having a resolution of $[x_{\text{res}} \ y_{\text{res}} \ z_{\text{res}}]$. As a grid map, we define a scalar function $F_d : \mathbb{D}_d \rightarrow \mathbb{R}^+$ that maps each cell of the discretized operation space \mathbb{D}_d to a positive real value. Note that for optimizing paths on grids, only positive grid values are useful, otherwise, the optimization would converge to an infinite number of undesired path cycles over non-positive cells. Therefore, the objective grid maps F_d (risk, noise, and radio disturbance) presented in the following are always

mapped to positive values without the restriction of generality $\tilde{F}_d = F_d + |\min(0, \min(F_d) - 1)|$. The grid-based objective function $f(\mathbf{z})$ sums up grid values along the path $\Pi(\mathbf{z})$. This is formalized as trapezoidal integral

$$f(\Pi(\mathbf{z})) = \sum_{i=1}^{|\Pi|-1} \frac{F(\boldsymbol{\pi}_{i-1}) + F(\boldsymbol{\pi}_i)}{2} |\boldsymbol{\pi}_i - \boldsymbol{\pi}_{i-1}|, \quad (3)$$

along the path Π over the linear interpolation F of the grid map \tilde{F}_d , with $|\boldsymbol{\pi}|$ being the Euclidean norm of $\boldsymbol{\pi}$.

Every objective that can not be described by (3) belongs to the category of non-grid-based objectives.

Before we continue with the exemplary definition of different objectives, we introduce a distance map operator δ on a two-dimensional binary grid map $F_d^{2D} : \mathbb{N}^{m \times n} \rightarrow \{0, 1\}$ by

$$\delta : F_d^{2D} \rightarrow (a_{ij}) \in \mathbb{R}^{m \times n} \quad \text{with} \quad (4)$$

$$a_{ij} = \begin{cases} 0, & \text{if } F_d^{2D}(i, j) = 1 \\ \min_{k,l} \sqrt{(i-k)^2 + (j-l)^2} \Big|_{F_d^{2D}(k,l)=1}, & \text{else.} \end{cases}$$

The distance map operator δ calculates the distance from every cell in F_d^{2D} to its nearest cell containing the value ‘1’. In Section V-A1, we provide examples for all grid maps that we introduce formally in the following sections.

1) STATIC RISK

The three-dimensional space \mathbb{D}_d where we want to plan paths consists of cells with different risk values assigned. Following the static risk objective, the UAV is supposed to fly through low-risk spaces. We define spaces with little risk as those over buildings (excluding buildings of education, hospitals, military and railway buildings, and places of worship) and water surfaces, but other definitions can equally be used. For deriving the risk grid map, we start with a two-dimensional, binary building grid map $F_{d,B}^{2D}$ that consists of ‘1’-valued cells, where a building is located in the cell’s spatial position, and of ‘0’-valued cells, where no building is located. Then, we calculate the two-dimensional risk grid map $F_{d,R}^{2D}$ by applying the distance operator (4) $F_{d,R}^{2D} = \delta(F_{d,B}^{2D})$. Thus, the risk map obtains a gradient towards low-risk cells. Furthermore, we set cells of the risk grid map that are located in areas of water to a low-risk value. When calculating risk values for the z -dimension of the grid map, we assume that the potential risk increases with the flight altitude of the UAV. Therefore, we derive the three-dimensional risk grid map $F_{d,R}$ by applying a two-dimensional maximum filter with a 3×3 -window to $F_{d,R}^{2D}$ layer by layer. By calculating the line integral (3) over $F_{d,R}$ along the path $\Pi(\mathbf{z})$, we obtain the risk objective function $f_R(\mathbf{z})$.

2) ENERGY

We now want to approximate the energy consumption of a UAV following a path Π . We assume that the UAV flies at a reduced speed during climb compared to level

flight. Furthermore, the aerial vehicle flies even slower during descent to avoid air turbulence underneath the rotors, which leads to an unstable flight. Thus, for our energy consumption model, we set the horizontal cruise velocity to $v_{xy} = 14$ m/s, the ascending velocity to $v_{z,\uparrow} = 2$ m/s, and the descending velocity to $v_{z,\downarrow} = 1$ m/s. The UAV’s flight path is described by smooth NURBS curves in the three-dimensional space, but for the calculation of the UAV’s energy consumption, we project the path Π in the xy -plane and obtain the two-dimensional path Π_{xy} . Furthermore, we distinguish between the path’s z -components that point upwards $\Pi_{z,\uparrow}$ and those that point downwards $\Pi_{z,\downarrow}$. Following the energy models for hovering, climbing, and forward motion by Reid [25], we derive our energy consumption model for the given parameters

$$f_E(\mathbf{z}) = \frac{1}{2} m v_{xy}^2 + c_E (|\Pi_{xy}| + 10|\Pi_{z,\uparrow}| + 15|\Pi_{z,\downarrow}|), \quad (5)$$

where $|\cdot|$ is the Euclidean length of the path projections, m is the UAV mass, and c_E denotes a vehicle-specific energy parameter.

3) STATIC NOISE VIOLATION

Torija et al. [26] show that city residents perceive noise generated by aerial vehicles as less annoying if the flight paths lead over the streets as the flight noise vanishes in the traffic noise. Therefore, we model our noise violation objective to favor paths that lead over streets. We begin with a street grid map $F_{d,S}^{2D}$ that consists of ‘1’-valued cells, where a street is located at the cell’s spatial position, and of ‘0’-valued cells, where no street is located. Then, we apply the distance map operator (4) to generate a gradient towards streets. Additionally, we set the cells of the noise grid map that are located in areas of water or industrial regions to a minimum noise value. In the same way, we set the cells of the noise grid map that are located in parks or residential areas to a maximum noise value. We obtain a two-dimensional noise grid map $F_{d,N}^{2D}$.

We assume $F_{d,N}^{2D}$ to be a layer of the three-dimensional noise grid $F_{d,N} : \mathbb{D}_d \rightarrow \mathbb{R}$ at the UAV’s minimal flight height $z_{f,\min}$. Perceived noise decreases quadratically with distance. To derive the remaining layers of $F_{d,N}$, we apply the inverse square law

$$F_{d,N}(z) = \begin{cases} F_{d,N}^{2D}, & \text{if } \Delta z \leq 0 \\ \frac{F_{d,N}^{2D}}{(\Delta z + 1)^2}, & \text{else} \end{cases} \quad (6)$$

with $\Delta z = z - z_{f,\min}$. Finally, we use (3) to integrate over the noise grid map $F_{d,N}$ along the path $\Pi(\mathbf{z})$ and obtain the noise objective function $f_N(\mathbf{z})$.

4) RADIO DISTURBANCE

To ensure safe UAV operations, we assume the need for a permanent radio connection between the aerial vehicle and a ground station. Therefore, a stable connection to a cell tower is essential. We assume the radio signal disturbance

at the position of a cell tower to attain an arbitrarily chosen best value of $D_0 = -100$. The signal disturbance increases following the inverse square law. Dependent on the position of a single radio cell tower $\mathbf{x}_R \in \mathbb{D}$, we calculate the radio signal disturbance at a point $\mathbf{x} \in \mathbb{D}$ qualitatively by

$$D(\mathbf{x}) = \frac{D_0}{(\mu r + 1)^2}, \quad (7)$$

with $r = |\mathbf{x}_R - \mathbf{x}|$ being the Euclidean distance between the point \mathbf{x} and the radio cell tower position, and $\mu \in \mathbb{R}^+$ being a scaling factor. We now have a three-dimensional grid map for a single cell tower. Next, we calculate a superposition (minimum operator) for all cell towers and obtain a three-dimensional radio disturbance map $F_{d,D} : \mathbb{D}_d \rightarrow \mathbb{R}$. To summarize, when planning a path through the radio disturbance map, we want to preferably fly through cells with lower signal disturbance. Thus, we compute the radio disturbance objective function $f_D(\mathbf{z})$ by calculating the trapezoidal integral (3) over $F_{d,D}$ along the path $\Pi(\mathbf{z})$.

D. CONSTRAINTS

We include constraints $g_f(\mathbf{z})$ as soft constraints into our optimization problem by adding them to every objective function $f(\mathbf{z})$ yielding the constrained objective functions

$$f_{\{\cdot\},C}(\mathbf{z}) = f_{\{\cdot\}}(\mathbf{z}) + \xi \sum_i \sqrt{H(-g_{f,i}(\mathbf{z}))g_{f,i}(\mathbf{z})^2}, \quad (8)$$

with $\{\cdot\}$ being a wildcard for $\{R, E, N, D\}$, the Heaviside function H and a parameter ξ that we choose arbitrarily large to ensure that $f_{\{\cdot\},C}(\mathbf{z}) \gg f_{\{\cdot\}}(\mathbf{z})$ if the constraint is violated.

1) MINIMUM FLIGHT HEIGHT

Except for the take-off and landing phase, we want the UAV's path to stay above a minimum flight height $z_{f,\min}$. Introducing the inequality constraint

$$g_{1,i}(\mathbf{z}) = z_i - z_{f,\min} \geq 0, \quad \forall i \in \{1, \dots, n_p - 2\} \quad (9)$$

we make sure that the control points' z -components of the curve will not decrease below $z_{f,\min}$. Note that the convex hull property [23] of NURBS curves guarantees that no path point $\boldsymbol{\pi}$ will fall below $z_{f,\min}$ if there is no control point $\mathbf{P}_i = [x_i \ y_i \ z_i]^T$ below $z_{f,\min}$.

If the path's fixed start control point $\mathbf{P}_0 = [x_0 \ y_0 \ z_0]$ or endpoint $\mathbf{P}_{n_p-1} = [x_{n_p-1} \ y_{n_p-1} \ z_{n_p-1}]$ lie below $z_{f,\min}$, we also lock the second and second last control points in position and set their components to $\mathbf{P}_1 = [x_0 \ y_0 \ z_{f,\min}]$ with weight $w_1 = 100$, as well as $\mathbf{P}_{n_p-2} = [x_{n_p-1} \ y_{n_p-1} \ z_{f,\min}]$ with weight $w_{n_p-2} = 100$. This ensures a vertical flight beneath the minimal flight height during the takeoff and landing phases of the aerial vehicle.

2) STATIC OBSTACLE COLLISION AVOIDANCE

The UAV's path must not go through static obstacles like buildings. We use a twofold approach that allows us to compute the inequality constraint with two two-dimensional grid

maps instead of a three-dimensional grid map. In the first step, we make use of a two-dimensional height grid map $F_{d,H}^{2D}$. Each grid map's cell contains the height value of the tallest building that is located in the respective cell. If a waypoint $\boldsymbol{\pi}_i = [\pi_{i,x} \ \pi_{i,y} \ \pi_{i,z}]$ of the path Π lies below the height of the respective height map value, we save it in a list of unsafe waypoints Π^\times . The path Π is punished following the inequality constraint for all $i \in \{0, \dots, |\Pi| - 1\}$

$$g_{2,i}(\mathbf{z}) = \pi_{i,z} - F_{d,H}^{2D}(\pi_{i,x}, \pi_{i,y}) \geq 0. \quad (10)$$

As a consequence, the unsafe waypoints are pushed out of static obstacles in the direction of the z -axis.

In the second step, we also want to create a drift that moves unsafe waypoints out of static obstacles in the xy -plane. Therefore, we use the building grid map $F_{d,B}^{2D}$, which we already introduced in Section III-C1. By applying the distance map operator (4) to the inverted building grid map, we obtain an obstacle grid map

$$F_{d,O}^{2D} = \delta(\bar{F}_{d,B}^{2D}) \quad (11)$$

with gradients that point away from the centers of buildings in the direction of the nearest free-space cells. We use this obstacle grid map and obtain the second part of the constraint for all $i \in \{0, \dots, |\Pi^\times| - 1\}$

$$g_{3,i}(\mathbf{z}) = -F_{d,O}^{2D}(\pi_{i,x}^\times, \pi_{i,y}^\times) > 0, \quad (12)$$

which ensures that during optimization unsafe waypoints $\boldsymbol{\pi}_i^\times = [\pi_{i,x}^\times \ \pi_{i,y}^\times \ \pi_{i,z}^\times] \in \Pi^\times$ are pushed on the obstacle grid map towards cells that equal zero.

IV. PROPOSED FRAMEWORK

In the following Section IV-A, we introduce the different modules of our hybrid framework to solve many-objective path planning problems. Furthermore, we go into more detail about two features, which are the adaptive-number-of-control-points-feature in Section IV-B and the niching-feature in Section IV-C.

A. HYBRID APPROACH

We propose a hybrid many-objective path planning approach in which, in the first step, an efficient Dijkstra algorithm is applied, which provides optimal solutions for separate single-objective problems. In the second step, by utilizing many-objective evolutionary computation, we obtain solutions in a more suitable path representation and a set of trade-off solutions called the Pareto set. This set allows the user to select an optimal solution depending on preferences. For example, the user can adjust the priority of risk avoidance depending on the payload or the priority of noise avoidance depending on the daytime.

We show the structure of the framework in Fig. 1. The start point \mathbf{x}_s and goal point \mathbf{x}_g of the path, as well as the grid maps that we use to calculate the four objectives and three constraints are input into the framework. Note that before the optimization starts,

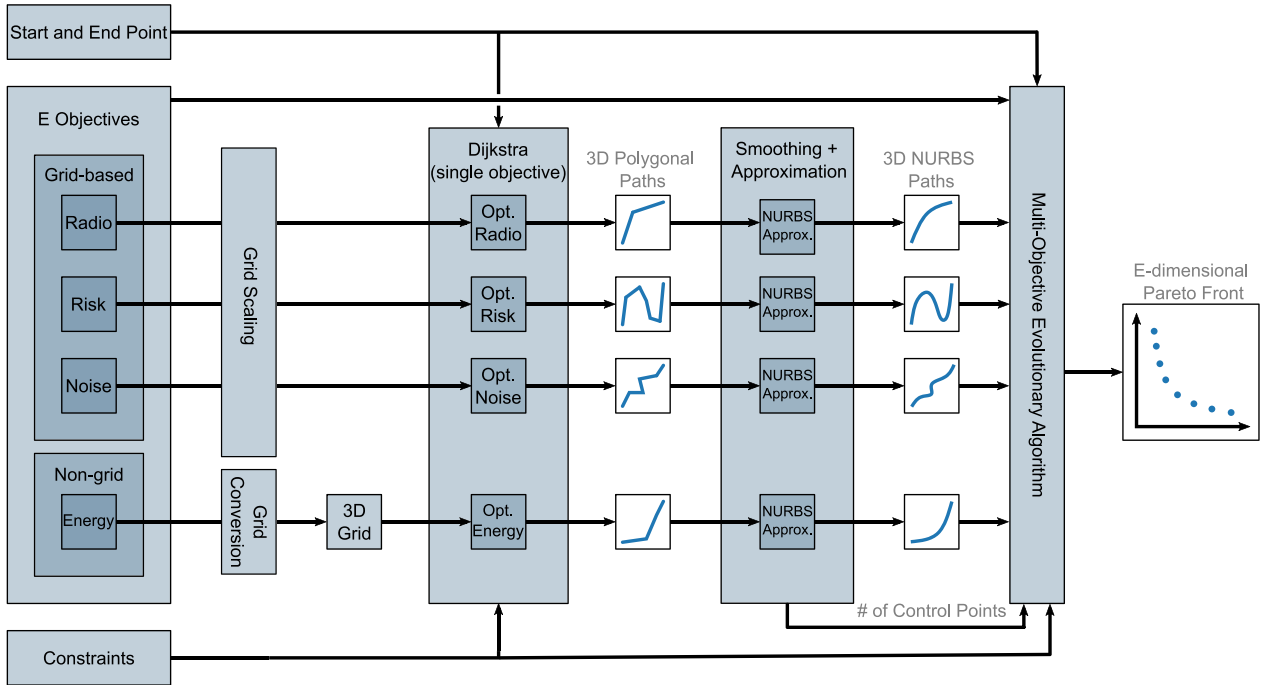


FIGURE 1. Diagram of the proposed hybrid optimization framework. Depending on the objective class (grid-based or non-grid-based), the corresponding grids are pre-processed before the Dijkstra algorithm finds an optimal solution for every objective separately. Then, the obtained solutions are smoothed and approximated by NURBS curves. Finally, the different smooth paths serve as initial solutions for the many-objective evolutionary algorithm, which generates a Pareto set of trade-off paths.

- 1) the grids of all grid-based objectives are scaled to a scenario size dependent resolution of $[\tilde{x}_{res} \tilde{y}_{res} \tilde{z}_{res}]$, which is a trade-off. A finer resolution would increase the solution quality of the following Dijkstra algorithm. A coarser resolution would make the Dijkstra algorithm need less computational time.
- 2) Following a previous work [27], all non-grid-based objectives are converted into a 3D grid representation to make the following Dijkstra algorithm applicable to them.
- 3) has its advantage over other optimization techniques in finding non-dominated solutions and building a Pareto set with highly diversified solutions.

In the first step of our hybrid approach, we apply the single-objective Dijkstra solver to every objective separately. For every objective, this results in a three-dimensional polygonal path, which is guaranteed to be optimal in the used grid structure. Next, we smooth and approximate the polygonal paths and obtain three-dimensional NURBS curves. Those splines are the initial solutions in a many-objective evolutionary optimization problem calculating an E -dimensional Pareto Front. Note that the evolutionary algorithm has versatile purposes as it

- 1) considers the grid-based objectives with their original resolution $[x_{res} y_{res} z_{res}]$ and the non-grid objectives in its original formulation (e.g., the energy consumption model (5)),
- 2) corrects constraint violations that may be induced by using the original grid resolution or the smoothing and approximation step, and

B. ADAPTIVE-NUMBER-OF-CONTROL-POINTS (ANCP) FEATURE

This feature is involved in the smoothing and approximation step during the transition from the Dijkstra algorithm to the evolutionary algorithm. Here, the path representation changes from a polygonal path definition to a NURBS definition. Therefore, we first smooth the polygonal path by applying a Gaussian filter with kernel $\mathcal{K} = [1 \ 2 \ 1]$ three times. Then, we approximate the smoothed path with a NURBS curve [28]. For this operation, the crucial parameter is the number of control points n_p of the resulting NURBS curve. If n_p is too small, the approximation error becomes too large. The resulting path has nothing in common with the original Dijkstra path, which makes the Dijkstra calculation obsolete. Whereas, if n_p is too large, this will increase the size of the optimization vector \mathbf{z} and therefore the search space in the many-objective optimization step. A suitable number of control points depends on the Dijkstra path's length and curviness, which is why it would be pointless to set n_p as a hyperparameter before the optimization. Instead, we determine the best number of control points for a respective scenario adaptively. For that, we increase the number of control points beginning with the minimum value of $n_p = 3$

in an iterative process. The process includes 1) the approximation of the NURBS curve with the current n_p and 2) the calculation of the error ε between the Dijkstra path Π_d and its NURBS approximation Π . If the error falls below a given threshold τ_{ANCP} , we have found a suitable number of control points for the approximation.

Note that this process is carried out for each objective and therefore for each Dijkstra path Π_d separately producing potentially different optimal numbers of control points. By picking the maximum value, we finally obtain our parameter n_p .

As error term ε , we use an adjusted Chamfer distance metric

$$\varepsilon(\Pi, \Pi_d) = \max \left(\sum_{\pi \in \Pi} \min_{\pi_d \in \Pi_d} |\pi - \pi_d|^2, \sum_{\pi_d \in \Pi_d} \min_{\pi \in \Pi} |\pi - \pi_d|^2 \right), \quad (13)$$

using the maximum operator to ensure that the approximation does not exceed a threshold at any point.

C. NICHING FEATURE

The second feature has a crucial role in the population building of the metaheuristic algorithm. Let us consider an example where one of $E = 4$ different pre-processed paths slightly violates a constraint due to the spline approximation step. This solution would then be dominated by the remaining three pre-processed solutions and thus not be considered anymore in the evolutionary algorithm. This means that the Dijkstra calculation of this one poor path would have no purpose although after only some minor adaptations to the path's shape, it would satisfy the constraints and become a good quality solution.

We want to solve this problem by introducing *niches*. Niching is the generic term for a class of techniques commonly used in evolutionary multimodal optimization [29]. By dividing the set of solutions into sub-populations, niching aims for the preservation of the solution diversity during optimization to find multiple (local) optima. The similarity of the different state-of-the-art approaches lies in the utilization of some kind of distance metric to assign solutions to different sub-populations [30]. Our niching approach differs from the niching strategy of other constrained optimization solvers [31], [32], [33] in that we neither need any additional computations nor extra parameters to build sub-populations. Instead, we can directly use the pre-processed candidate solutions as the origin of a sub-population. A separate niche exists for every objective that only holds the candidate solutions related to one of the objectives. The evolutionary algorithm applies the reproduction and selection process within the niches until at least one candidate solution in every sub-population satisfies all constraints. When this constraint satisfaction condition is met, the niches are dissolved into a complete population and the evolutionary process continues.

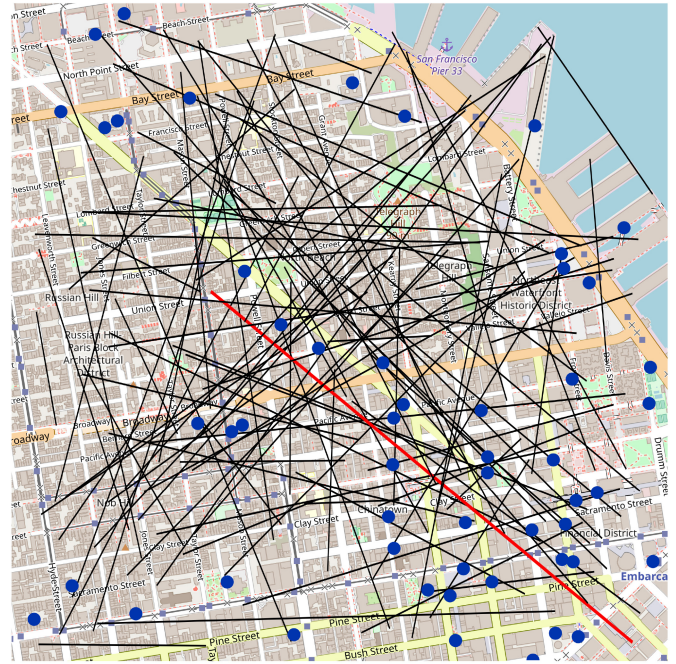


FIGURE 2. The map extract of San Francisco [36] that was used to derive the grid maps. The overlying lines denote the start and endpoints of the scenarios that we used for the evaluation. For the red path, we provide more detailed information on the calculated paths further below. The blue circles denote the positions of radio masts.

V. EVALUATION

In the following, we describe the evaluation of the proposed framework. Therefore, in Section V-A, we explain the experimental setup. In Section V-B, we then present and discuss the results of the experiments.

A. SETUP

In order to evaluate the proposed framework, we describe our experiment's setup consisting of an exemplary city where the paths are planned, different metaheuristic solvers, which were utilized in our framework, all hyperparameters, and parameters of the framework, and the metrics that we apply for the evaluation.

1) SCENARIO

As an example city for our evaluation, we choose a section of San Francisco that is visualized in Fig. 2. The rendered map shows lines, which indicate the 100 random start and end positions for the statistical path planning evaluation. Moreover, we extract the positions of 4G radio masts, indicated by blue circles in Fig. 2, from OCID [34]. We use them for the calculation of the radio disturbance map. With the help of OpenStreetMap [35], we can access data like the height of the buildings, the course of the streets, or semantic data containing information about, for example, the positions of industrial areas or water areas. With this information, we derive the different grid maps that were introduced in Section III-C that are 1) the risk grid map $F_{d,R}$, which we visualize as a cross-section at height $z = 0$ m in Fig. 3, 2) the noise grid map $F_{d,N}$, whose cross-section at height

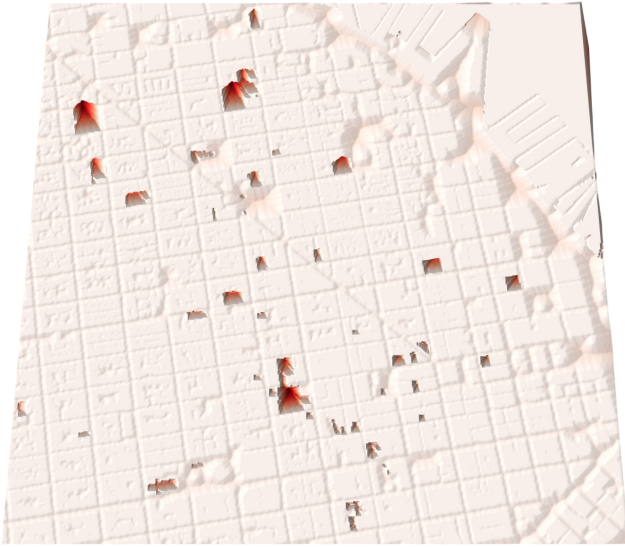


FIGURE 3. The risk information of the San Francisco scenario at height $z = 0$ m that we derived from OpenStreetMap data. Light-colored areas indicate low-risk values over normal buildings and water surfaces, gray areas indicate medium-risk areas over streets, and red areas indicate high-risk values over educational, medical, railway, military, and worship buildings.

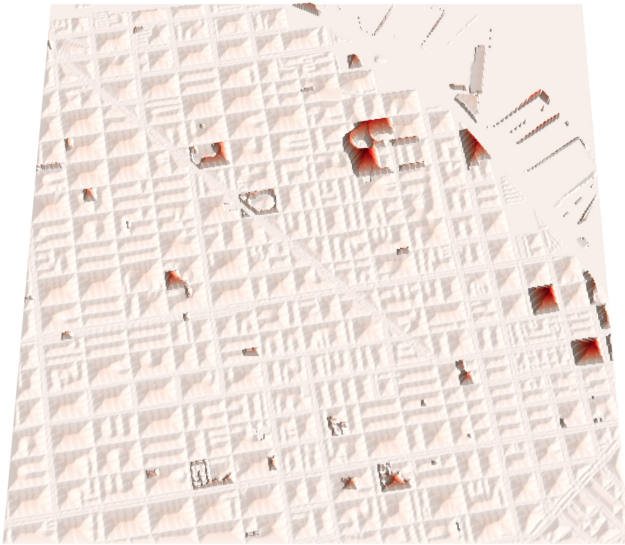


FIGURE 4. The noise sensitivity information of the San Francisco scenario at height $z = 0$ m that we derived from OpenStreetMap data. Light-colored areas indicate low noise values over streets, industrial areas and water surfaces, gray areas indicate medium-noise values over normal buildings, and red areas indicate high-noise values over residential areas, and parks.

$z = 0$ m we display in Fig. 4, and 3) the radio disturbance map $F_{d,D}$, whose cross-section at height $z = z_R$ we show in Fig. 5. In the visualizations, low values are shown in white and high values in red. In the radio disturbance map, for example, one can derive the positions of the radio masts from the white-colored spots.

2) SOLVERS

For the metaheuristic optimization step in our framework, we can utilize an arbitrary many-objective evolutionary algorithm. In our experiments, we choose three

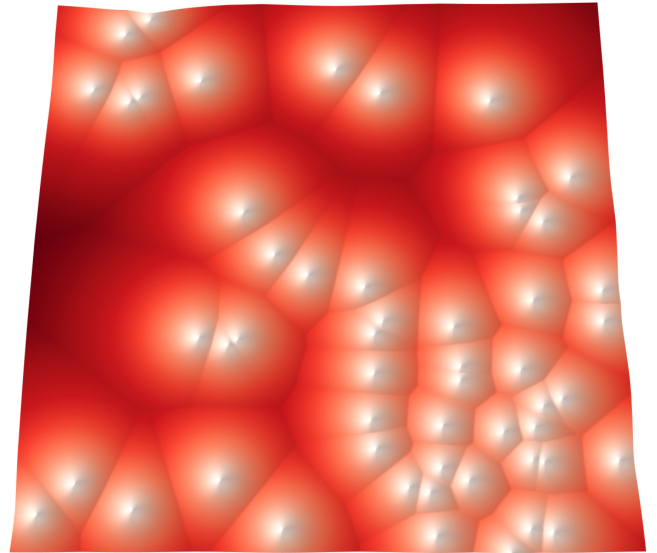


FIGURE 5. The radio disturbance information of the San Francisco scenario at height $z = z_R$ that we derived from OCID [34] data. White areas indicate low radio disturbance values around radio masts, and red areas indicate high radio disturbance values.

TABLE 1. Parameters and specifications.

	Parameter	Symbol	Value
Problem Definition	Map Dimensions	$[x_{\min} \ x_{\max}]$	[0 2000]
		$[y_{\min} \ y_{\max}]$	[0 2000]
		$[z_{\min} \ z_{\max}]$	[0 300]
	Min. Flight Height	$z_{f,\min}$	50
	UAV Mass	m	1.2 kg
	Energy Parameter	c_E	9.12 J/m
Obj.	Radio Cell Height	z_R	80
	Radio Scaling Fac.	μ	0.01
Dijkstra	Discretization	$\tilde{x}_{\text{res}} \ \tilde{y}_{\text{res}} \ \tilde{z}_{\text{res}}$	[10 10 10]
Approx.	Threshold	τ_{ANCP}	7
Meta-heuristic	Discretization	$x_{\text{res}} \ y_{\text{res}} \ z_{\text{res}}$	[4 4 10]
	Crowding Degree	η_X	20
	Crossover Prob.	p_X	0.9
	Crowding Degree	η_{mut}	20
	Mutation Prob.	p_{mut}	1/dim(\mathbf{z})
	Population Size	N_{pop}	100

state-of-the-art implementations that are the Non-dominated Sorting Genetic Algorithm (NSGA-3) [37], the Reference Vector guided Evolutionary Algorithm (RVEA) [38], and the S -Metric Selection Evolutionary Multiobjective Optimization Algorithm (SMS-EMOA) [39] with their default parameter settings. The three algorithms differ primarily in their selection operator, which affects the result's quality (i.e., the Pareto set's convergence and diversity) significantly.

3) PARAMETERS

For the sake of completeness, we summarize important parameters in Table 1. The gray highlighted parameters are default parameters of the used state-of-the-art

algorithms [37]. The rest of the parameters were introduced in Sections III and IV and were either determined empirically or arbitrarily but realistically.

4) METRICS

We utilize different metrics to evaluate our framework.

The *Hypervolume* (HV) metric measures the E -dimensional volume between the E -dimensional Pareto set and a user-defined reference point in the objective space. If one set of non-dominated solutions has a higher hypervolume than another for the same reference point, this is an indicator of a better convergence towards the optimal Pareto front.

The *Generational Distance* (GD) of a solution front measures the distance of an examined set of non-dominated solutions to a reference set of solutions. The lower the GD value of a front, the better its convergence to the reference front.

The *Inverted Generational Distance* (IGD) of a front measures the distance of a reference set of solutions to the examined set of non-dominated solutions. The lower the IGD value of a front, the better its convergence and its diversity compared to the reference front.

We discuss the determination of the reference front in Section V-B1. For the metrics' equations, we refer the interested reader to another study [22].

B. RESULTS & DISCUSSION

In the following Section V-B1, we examine the behavior of the proposed framework and compare it to different metaheuristic solvers. Then, we evaluate the effectiveness of the two introduced features in Sections V-B2 and V-B3 respectively. We show exemplary optimized paths in Section V-B4. For our statistical analysis, we conduct all tests on 100 different start and endpoint configurations (i.e., scenarios), which are visualized as lines in Fig. 2.

1) HYBRID VS. STANDARD

In the first experiment, we compare the performance of the standard metaheuristic solvers NSGA3, RVEA, and SMS-EMOA with their respective performance in our hybrid framework (H. NSGA3, H. RVEA, H. SMS-EMOA). The standard solvers are initialized with *straight-line* paths.

For every scenario, we normalize the obtained Pareto sets' HVs to the best and worst HV over all iterations. In Fig. 6, we plot the mean of the normalized HVs for all scenarios over the number of cost function evaluations. Note that a cost function evaluation is the evaluation of one candidate solution regarding all four objective functions. The optimization terminates after 5000 function evaluations. We can observe the large initial performance difference of 0.986 between hybrid and standard solvers. In the further course, the mean values of the three standard approaches increase and converge towards values in the range between 0.76 and 0.8, but do not reach the initial performance of the hybrid approaches.

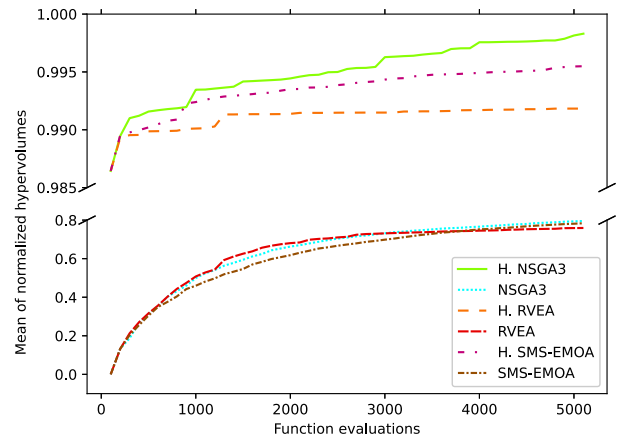


FIGURE 6. Performance of the hybrid and standard algorithms indicated by the obtained hypervolumes that are normalized to the best highest hypervolume per scenario and averaged over 100 scenarios.

TABLE 2. Performance comparison of the 6 optimizers.

Optimizer	Norm. HV		GD		IGD	
	μ	σ	μ	σ	μ	σ
H. NSGA3	99%	0.00	0.034	0.00	0.046	0.10
NSGA3	23%	0.17	2.641	25.10	2.77	25.13
H. RVEA	96%	0.10	0.075	0.00	0.109	0.00
RVEA	5%	0.14	0.132	0.00	0.244	0.14
H. SMS-EMOA	98%	0.10	0.031	0.00	0.027	0.00
SMS-EMOA	17%	0.14	2.628	25.09	2.773	25.12

The best hybrid approach is H. NSGA3, which achieves a mean normalized HV of 0.998.

Furthermore, over all scenarios, we compute the mean and the standard deviation of the normalized HV, GD, and IGD values of the final obtained sets of non-dominated solutions for all six solvers. The numeric values can be taken from Table 2. Note that the HV is normalized to the worst and best HV of the last iteration. We choose the set of non-dominated solutions with the highest HV as a reference set for the calculation of GD and IGD metrics. The best GD and IGD values are achieved by the H. SMS-EMOA method, which are 77% and 89% better than those of the best standard method RVEA. For a statistical comparison of each hybrid approach with its state-of-the-art counterpart respectively, we use the two-tailed Mann-Whitney U test with $n_1 = 100$, $n_2 = 100$, and $p < 0.05$. The results indicate that the samples of the normalized HV, the GD, and the IGD metric differ significantly between hybrid and standard approaches.

We show the influence of our hybrid initialization step compared to the standard *straight-line* initialization. Therefore, in Fig. 7, we visualize the 4-dimensional Pareto sets that the NSGA3 and the H. NSGA3 solvers obtain after optimizing the path that is indicated by the red line in Fig. 2. For both solvers, we plot the space that is spanned by the dominated solutions of all generations in dark yellow and the non-dominated solutions of the last iteration in gray. In particular, we highlight the extreme points of the Pareto sets

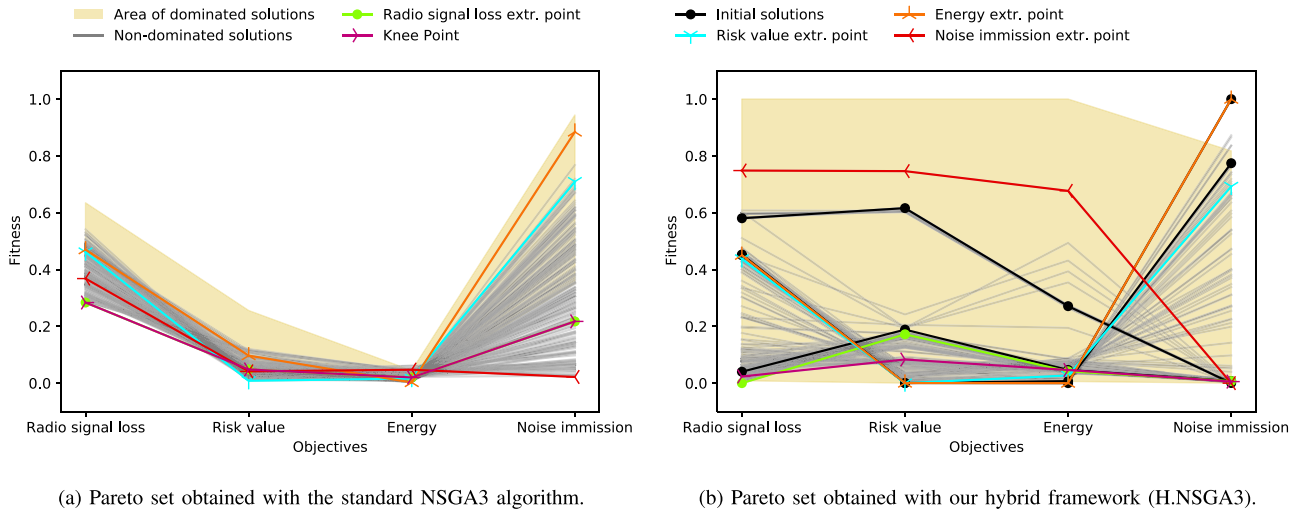


FIGURE 7. Pareto set comparison for a selected scenario (colored in red in Fig. 2).

for all objectives as well as the knee point. The knee point is the solution that has the smallest Euclidean distance to the origin of the objective space. It often realizes a suitable compromise between the objectives, which we examine in more detail later. In Fig. 7b, we also see the solutions derived from Dijkstra pre-processing in black, which serve as initial values in the H. NSGA3. It is noticeable that these are very close to the extreme points finally reached.

Comparing the Pareto sets of the standard and the hybrid approach, the differences in the extreme values in Fig. 7a and Fig. 7b are in order 0.284, 0.008, 0.002, and 0.022. We observe a comparatively larger difference in the noise immission and radio loss objectives, and we will take a closer look at these two objectives in the following.

Let us first consider the noise immission objective. The corresponding initial solution in Fig. 7b (black line with a dot in the lower right corner) differs only slightly from the noise immission extreme point of the standard method in Fig. 7a concerning the noise immission fitness. Regarding the other objectives, this solution even achieves a significantly worse fitness than the standard method. For the user, this solution may not be useful because of the high-risk value. But during the optimization, this solution is responsible for finding other non-dominated solutions that the standard algorithm does not find (gray lines with $f_E(\mathbf{z}) \approx 0.4$ in Fig. 7b) although they could be interesting for the user. Thus, the hybrid approach creates more diversity in the computed Pareto set, giving the user more flexibility in deciding on an appropriate path.

Second, let us consider the radio signal loss objective. In this case, the Dijkstra pre-processing provides a much better solution (black line in the lower left corner of Fig. 7b) than the standard approach. Around this solution, the evolutionary algorithm subsequently provides a high density of non-dominated solutions in the objective space, as can be seen from the many gray lines in the immediate neighborhood.

Among these is the knee point solution (violet line), which is a good trade-off for a user.

Discussion: We could observe that our hybrid framework achieves significantly better results than the standard methods. The reason for this lies in the pre-processing step, where the Dijkstra algorithm finds good approximations for the final Pareto set's extreme points. This can be seen from the fact that the initial solutions' objective values (black points) of the evolutionary algorithm in Fig. 7b, which come from the Dijkstra solutions, almost correspond to the final extreme points. The important insight is that pre-processing increases the quality of the final extreme points, although the Dijkstra algorithm in the first stage of our framework searches on a different grid scaling (i.e., another representation) than the evolutionary algorithm in the second stage. By knowing good initial extreme points, the second-stage evolutionary algorithm does not need to excessively search in unknown areas of the search space (exploration) but rather finds non-dominated solutions that compromise the already pre-computed extreme point solutions (exploitation).

We could observe from Fig. 6 that without those pre-computed extreme point solutions, the standard metaheuristic algorithms need considerably more iterations to explore the search space and find comparably good solutions. During this exploration, the evolutionary algorithm can most likely get stuck in local optima due to the highly multimodal character of the optimization problem, and therefore achieves a worse convergence to the true Pareto Front, as the higher (i.e., worse) GD values in Table 2 suggest. Moreover, the higher (i.e., worse) IGD values in the same table show that the standard evolutionary algorithms fail to achieve a comparably good diversity of the Pareto set.

To sum up, our hybrid approach for many-objective path planning problems shows several advantages in comparison to standard many-objective optimization approaches, which are the quality of the obtained Pareto sets and the number

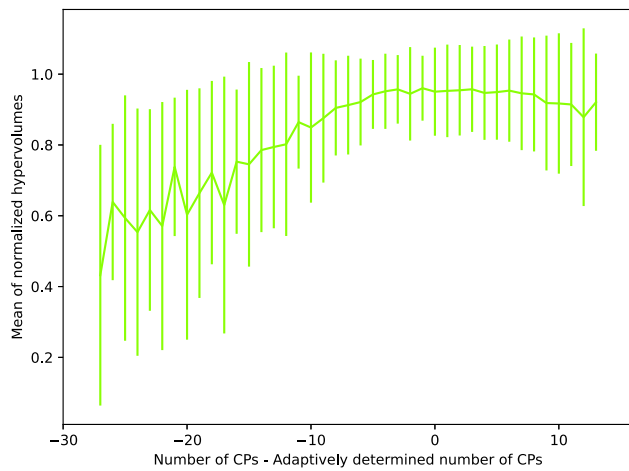


FIGURE 8. Examination results of the adaptive-number-of-control-points feature.

of iterations (i.e., the computation time) needed to calculate them. The user obtains more diversified paths that satisfy the needs of multiple stakeholders differently. Briefly, this means that the hybrid framework efficiently produces a set of UAV flight paths that are better regarding at least one objective and not worse regarding the other objectives in comparison to the standard approaches. On the downside, we can state that the efficient pre-computation of initial solutions with the Dijkstra algorithm depends on the size of the grid maps and thus on map dimensions and discretization resolution. Thus, for larger scenarios either the grid resolution has to be reduced or the scenario has to be decomposed into sub-problems. Furthermore, the framework requires the user to formulate the objectives in a grid-based or at least in a grid-transformable form, which means additional programming effort. However, if an objective is not grid-transformable in a meaningful way, it can still be integrated into the framework by bypassing the Dijkstra stage.

We note that assuming grid maps of the same size and resolution, it is also possible to sum the grid maps of all objective functions in a weighted fashion to generate more initial solutions. Questions about the design of the weights and the effectiveness of this approach are beyond the scope of this paper and will be investigated and answered in a future publication.

2) ADAPTIVE-NUMBER-OF-CONTROL-POINTS FEATURE

In the second test, we evaluate the ANCP feature. For the 100 scenarios separately, we run the optimization with an adaptively determined parameter n_p . Then, we run the optimization again for all 100 scenarios separately for a number of control points reaching from $n_p = 5$ to $n_p = 30$ control points. We plot the mean and the standard deviation (as symmetric errorbar) of the obtained normalized hypervolumes over the difference between the respective number of control points and the adaptively determined number of control points in Fig. 8. If n_p lies in the range around the adaptively determined ones (i.e., $x = 0$ on the x-axis), the

HV reaches its maximum value of 0.96. Apart from there, the mean HV values decrease visibly, while their standard deviations increase. If ten more control points are used for the approximation, the HV decreases by 4.2%. If n_p is reduced by ten control points, the HV even decreases by 10.4%.

Discussion: The test of the ANCP feature results in two insights. First, the number of control points parameter n_p has a strong influence on the HVs and therefore the quality of the obtained Pareto sets, as we can deduce from the curve's incline in Fig. 8. If n_p is too small, the NURBS curve no longer approximates the pre-processed Dijkstra path well enough and the resulting path is very likely to have a worse fitness, resulting in a smaller HV. Thus, the advantage of pre-processing is lost. If n_p is too large, the better approximation quality adds no benefit to the optimization, because it only increments the sampling rate for the very same path. Instead, the optimization vector becomes unnecessarily large, which increases the complexity of the optimization problem and thus the chances of a stagnating optimizer.

Second, using the adaptively computed parameter n_p , we obtain high hypervolumes persistently and independently of the chosen scenario. The reason for this is that the Dijkstra path solutions computed in the pre-processing step for all grid-based objectives of a scenario can already give a good estimate for the characteristic (i.e., curviness) of the final path solutions on the scenario. The curviness of the path correlates directly with the number of control points n_p in the NURBS representation of the path as the representation of a curved path requires more control points. From this correlation and the first insight described above, the better quality of the results with the ANCP feature can be explained. In summary, we find that the advantage of the ANCP feature lies in the applicability of the hybrid path planning framework to unknown scenarios without hyperparameter tuning as the crucial parameter n_p is automatically determined. However, we achieve this generalization by making the size of the optimization vector $\dim(\mathbf{z}) = 3(n_p - 2)$ (if the first and last control point is fixed) and thus the complexity of the optimization problem dependent on the curviness of the path.

3) NICHING FEATURE

In the third test, we evaluate the niching feature. For each of the 100 scenarios, we do two separate H. NSGA3 runs (enabled ANCP feature) with and without the niching feature. Each time, the obtained HVs of both Pareto sets are normalized to the best and worst HV of both iterations. In Fig. 9, we plot the mean of the normalized hypervolumes for all scenarios over the number of cost function evaluations. Initially, the mean values of the HVs of both strategies increase similarly. After about 100 iterations, the curves diverge strongly and reach a difference of 0.19 after 5000 iterations. Finally, the method with niching strategy is 29% better than the one without the niching strategy. In other words, the niching strategy requires 2300 fewer function

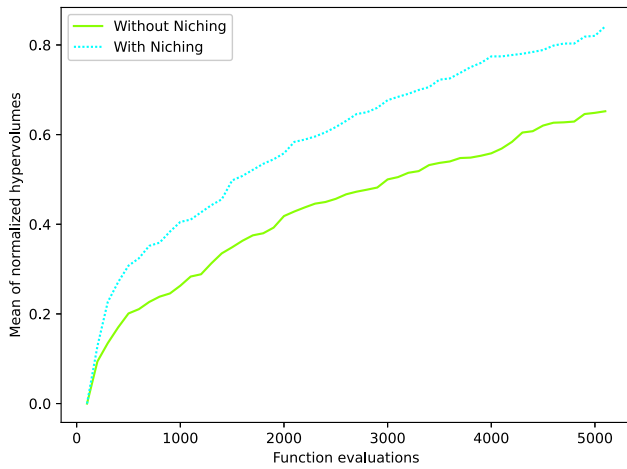


FIGURE 9. Performance test results for the niching feature.

evaluations to achieve the same HV as the approach without niching.

Discussion: The results of the described experiment demonstrate the positive effect that niching has on our hybrid framework. The problem solved with the niching feature lies in the different grid scaling and path representation in both stages of the hybrid framework. This causes good Dijkstra solutions to potentially violate the constraints in the second stage. The better performance of the framework with the niching feature is due to the initial preservation of all pre-processed solutions, constraint violating or not, knowing that within a few iterations, the constraints are potentially resolved. To sum up, niching allows us to integrate constraints into the hybrid optimization framework while still ensuring that no pre-processed solution will initially be rejected for constraint violation. However, the user of the framework must ensure that the defined constraints are satisfiable, otherwise, the niches will never migrate into the complete population and the intended many-objective optimization will fall back into E single-objective optimizations.

4) ANALYSIS OF VARIOUS PATHS DISCOVERED ACROSS THE PARETO SET

Finally, we examine some optimization results of the H. NSGA3 algorithm for the exemplary scenario that is indicated by the red line in Fig. 2. We have already seen the Pareto set for this scenario in Fig. 7b. With corresponding colors, we visualize the matching path representations of the Pareto set's extreme points and its knee point in Fig. 10.

The visible paths have different qualities. The Pareto set's extreme point paths are optimized regarding the respective objective. However, they can achieve sub-optimal function values for the remaining objectives. For example, a noise-optimal path runs at higher elevations above the city, resulting in high energy consumption. The first four rows of Table 3 approve this coherence. Every column shows the

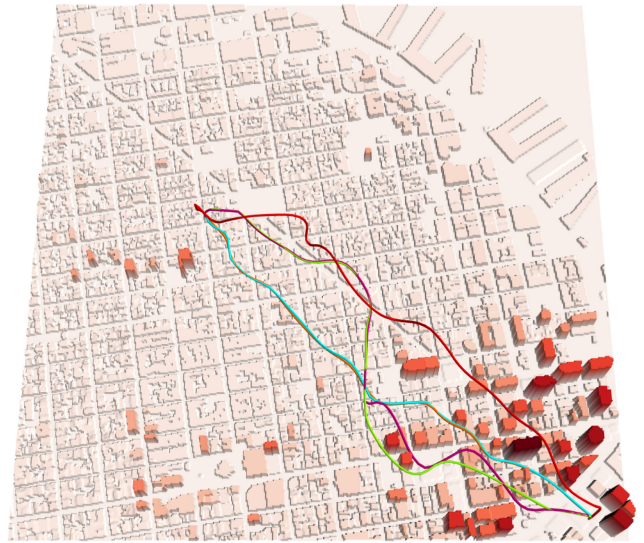


FIGURE 10. For an exemplary scenario, we plotted the paths belonging to the Pareto set's extreme points (radio: green, risk: blue, energy: orange, noise: red) and the knee point (violet) into the three-dimensional height map.

TABLE 3. Deviation from the best obtained fitness value.

Solution	Radio	Risk	Energy	Noise
Best radio solution	0%	212%	32%	22%
Best risk solution	204%	0%	20%	2143%
Best energy solution	210%	0%	0%	3094%
Best noise solution	349%	920%	512%	0%
Knee point solution	11%	103%	35%	17%

respective path's relative deviation to the corresponding best solution. For example, in terms of energy, the described noise-optimal path deviates from the energy-optimal path by 512%. A user will rarely select an extreme point as a solution. A good trade-off solution could be the knee point. The fifth row in Table 3 depicts the relative deviations of the knee point solution to the respective best solutions.

Discussion: Although the extreme points of a Pareto front will likely play a minor role in most real applications, it is advantageous to initially provide good approximations of the same to the optimizer. This way, the limits of the meaningful search space (i.e., the part of the search space where one can expect to find non-dominated solutions) are already defined. Consequently, this improves the chances of the evolutionary algorithm finding non-dominated solutions in highly diverse parts of the search space more efficiently. In summary, one must be aware that our framework does not generate one solution but a Pareto set of solutions, i.e., many possible paths. The decision on which path the drone should finally fly along is up to the user. For this task, preference-based selection algorithms or simple methods like knee-point selection can be used.

VI. CONCLUSION

We have introduced a hybrid framework that consists of a Dijkstra optimization stage and an evolutionary optimization

stage to tackle many-objective path planning problems for aerial vehicles in the three-dimensional urban space.

Contrary to the idea of other hybrid optimizers that first apply a metaheuristic or randomized search and then improve the found solutions by a local search, we change the order of exact and metaheuristic optimizers. Concerning many-objective path planning for smooth three-dimensional paths, we have shown in a broad analysis that this combination of a first exact optimization stage and a second metaheuristic optimization stage can save computational resources and improve the quality of obtained Pareto sets.

We have identified a crucial hyperparameter and described the problems that come along with the hybrid setup and the associated differences in the grid scaling and the path representation. For the stated problems, we provided solutions and showed their effectiveness in two additional analyses. The framework is applicable to arbitrary geographic areas and new objectives can be included. But, it is tailored to grid-based or at least grid-transformable objectives. There may be objectives that do not fulfill this restriction. For these, the preprocessing stage can be skipped. Another limitation is the size of the operation space, which is implicitly bounded by the Dijkstra algorithm. Thus, larger planning areas may need to be divided into sub-areas. The result of the proposed many-objective path planning framework is a Pareto set of paths balancing differently between the formulated objectives that may represent the interests of different stakeholders. Accordingly, the user can choose an appropriate path from the Pareto set.

With our approach, we are able to efficiently find paths that connect a defined start and endpoint with regard to different objectives. However, in the future we expect aerial transportation services in a city to require more than one single path but rather a complete network in which they are able to move freely. In the next step, we plan to use our framework to compute such connected aerial corridor networks.

REFERENCES

- [1] S. F. W. Lim, X. Jin, and J. S. Srail, "Consumer-driven e-commerce: A literature review, design framework, and research agenda on last-mile logistics models," *Int. J. Phys. Distrib. Logist. Manag.*, vol. 48, no. 3, pp. 308–332, 2018.
- [2] N. Boysen, S. Fedtke, and S. Schwerdfeger, "Last-mile delivery concepts: A survey from an operational research perspective," *OR Spectr.*, vol. 43, no. 1, pp. 1–58, 2020.
- [3] N. E. Neef, K. Kastner, M. Schmidt, and S. Schmidt, "On optimizing driving patterns of autonomous cargo bikes as a function of distance and speed—A psychological study," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 592–601, 2022.
- [4] I. Zubin, N. Van Oort, A. Van Binsbergen, and B. Van Arem, "Deployment scenarios for first/last-mile operations with driverless shuttles based on literature review and stakeholder survey," *IEEE Open J. Intell. Transp. Syst.*, vol. 2, pp. 322–337, 2021.
- [5] M. Heutger and M. Kückelhaus, "Unmanned aerial vehicles in logistics a DHL perspective on implications and use cases for the logistics industry," DHL Custom. Solut. Innovat., Troisdorf, Germany, Rep. 1, 2014. Accessed: Oct. 31, 2022. [Online]. Available: <https://www.dhl.com/discover/content/dam/dhl/downloads/interim/full/dhl-trend-report-uav.pdf>
- [6] G. Grandl et al., "The future of vertical mobility sizing the market for passenger, inspection, and goods services until 2035," Porsche Consult., Munich, Germany, Rep. 2, 2018. Accessed: Oct. 31, 2022. [Online]. Available: <https://fedotov.co/wp-content/uploads/2018/03/Future-of-Vertical-Mobility.pdf>
- [7] A. Steinberg, M. Cardei, and I. Cardei, "UAS batch path planning with a space-time graph," *IEEE Open J. Intell. Transp. Syst.*, vol. 2, pp. 60–72, 2021.
- [8] R. Papa, I. Cardei, and M. Cardei, "Generalized path planning for UTM systems with a space-time graph," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 351–368, 2022.
- [9] Wingcopter, "Wingcopter provides worldwide drone solutions for commercial and humanitarian applications to improve and save lives." Accessed: Nov. 1, 2022. [Online]. Available: <https://wingcopter.com/>
- [10] P. D. Vascik, H. Balakrishnan, and R. J. Hansman, "Assessment of air traffic control for urban air mobility and unmanned systems," in *Proc. ICRA*, 2018, pp. 1–9.
- [11] A. Bauranov and J. Rakas, "Designing airspace for urban air mobility: A review of concepts and approaches," *Progr. Aerosp. Sci.*, vol. 125, pp. 1–27, Aug. 2021.
- [12] E. Sunil et al., "Metropolis: Relating airspace structure and capacity for extreme traffic densities," in *Proc. ATM R&D Seminar*, 2015, pp. 1–11.
- [13] D.-S. Jang, C. A. Ippolito, S. Sankaraman, and V. Stepanyan, "Concepts of airspace structures and system analysis for UAS traffic flows for urban areas," in *Proc. AIAA Inf. Syst. AIAA Infotech Aerosp.*, 2017, pp. 1–15.
- [14] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [15] T. Back, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. New York, NY, USA: Oxford Univ. Press, 1996, ch. 2, pp. 106–130.
- [16] M. Emmerich and A. H. Deutz, "A tutorial on multiobjective optimization: Fundamentals and evolutionary methods," *Nat. Comput.*, vol. 17, no. 3, pp. 585–609, 2018.
- [17] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, "Evolutionary algorithm based offline/Online path planner for UAV navigation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 6, pp. 898–912, Dec. 2003.
- [18] J. Rubio-Hervas, A. Gupta, and Y.-S. Ong, "Data-driven risk assessment and multicriteria optimization of UAV operations," *Aerosp. Sci. Tech.*, vol. 77, pp. 510–523, Jun. 2018.
- [19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [20] S. Ghambari, M. Golabi, J. Lepagnot, M. Bréviillers, L. Jourdan, and L. Idoumghar, "An enhanced NSGA-II for multiobjective UAV path planning in urban environments," in *Proc. ICTAI*, 2020, pp. 106–111.
- [21] N. Sadallah, S. Yahiaoui, A. Bendjoudi, and N. Nouali-Taboudjemat, "Multi-objective offline and Online path planning for UAVs under dynamic urban environment," *Int. J. Intell. Robot. Appl.*, vol. 6, no. 1, pp. 119–138, 2021.
- [22] N. Hohmann, M. Bujny, J. Adamy, and M. Olhofer, "Hybrid evolutionary approach to multi-objective path planning for UAVs," in *Proc. SSCI*, 2021, pp. 1–8.
- [23] L. Piegl and W. Tiller, "Rational B-spline curves and surfaces," in *The NURBS Book*, 2nd ed. Berlin, Germany: Springer, 1997, ch. 4, pp. 117–138.
- [24] L. Piegl and W. Tiller, "Standards and data exchange," in *The NURBS Book*, 2nd ed. Berlin, Germany: Springer, 1997, ch. 12, pp. 571–580.
- [25] J. S. Reid, "Drone flight—What does basic physics say?" 2020. Accessed: Feb. 11, 2022. [Online]. Available: <https://homepages.abdn.ac.uk/nph120/meteo/DroneFlight.pdf>
- [26] A. J. Torija, Z. Li, and R. H. Self, "Effects of a hovering unmanned aerial vehicle on urban soundscapes perception," *Transp. Res. D Transpo. Environ.*, vol. 78, Jan. 2020, Art. no. 102195.
- [27] N. Hohmann, M. Bujny, J. Adamy, and M. Olhofer, "Multi-objective 3D path planning for UAVs in large-scale urban scenarios," in *Proc. CEC*, 2022, pp. 1–8.
- [28] L. Piegl and W. Tiller, "Curve and surface fitting," in *The NURBS Book*, 2nd ed. Berlin, Germany: Springer, 1997, ch. 9, pp. 410–413.
- [29] S. Das, S. Maity, B.-Y. Qu, and P. N. Suganthan, "Real-parameter evolutionary multimodal optimization—A survey of the state-of-the-art," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 71–88, 2011.

- [30] X. Li, M. G. Epitropakis, K. Deb, and A. Engelbrecht, "Seeking multiple solutions: An updated survey on niching methods and their applications," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 518–538, Aug. 2017.
- [31] K. Deb and S. Agrawal, "A niched-penalty approach for constraint handling in genetic algorithms," in *Proc. ICANNGA*, 1999, pp. 235–243.
- [32] D. J. Poole and C. B. Allen, "Constrained niching using differential evolution," *Swarm Evol. Comput.*, vol. 44, pp. 74–100, Feb. 2019.
- [33] K. Deb and A. Saha, "Multimodal optimization using a bi-objective evolutionary algorithm," *Evol. Comput.*, vol. 20, no. 1, pp. 27–62, 2012.
- [34] "OpenCellid contributors." Accessed: May 23, 2022. [Online]. Available: <https://www.opencellid.org/downloads.php>
- [35] "OpenStreetMap contributors." Accessed: Sep. 6, 2022. [Online]. Available: <https://www.openstreetmap.org>
- [36] OpenStreetMap. "San Francisco area around telegraph hill." Accessed: Sep. 6, 2022. [Online]. Available: <https://www.openstreetmap.org/search?query=san20francisco#map=16/37.8035/-122.4068>
- [37] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.
- [38] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.
- [39] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.



NIKOLAS HOHMANN received the B.Sc. degree in electrical engineering and information technology and the M.Sc. degree in automation technology from the Technical University of Darmstadt, Germany, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree with the Technical University of Darmstadt and Honda Research Institute Europe GmbH, Offenbach/Main, Germany. His current research interests involve applications of evolutionary optimization regarding multiple objectives, path

and trajectory planning for multiple agents, and dynamic optimization, in particular in the context of traffic scenarios for unmanned aerial vehicles.



SEBASTIAN BRULIN received the B.Sc. and M.Sc. degrees in mechanical and process engineering from the Technical University of Darmstadt, Germany, in 2013 and 2016, respectively, and the Ph.D. degree from the Institute for Fluid Mechanics and Aerodynamics, Technical University of Darmstadt in 2020. Since 2021, he has been working as a Senior Scientist with the Complex Systems Optimization and Analysis Group, Honda Research Institute Europe. His research interests include investigations of multi-agent system dynamics and multidisciplinary optimization in transportation applications.



JÜRGEN ADAMY received the Diploma and Dr.-Ing. degrees in electrical engineering from the Technical University of Dortmund, Germany, in 1987 and 1991, respectively. From 1992 to 1998, he was an Engineer and a Manager in the area of Control Applications with Siemens AG, Erlangen, Germany. In 1998, he became a Full Professor with the Technical University of Darmstadt, where he is the Head of Control Methods and Intelligent Systems Laboratory. His research interests are in the areas of nonlinear control, fuzzy systems, and mobile robots.



MARKUS OLHOFER graduated in electrical engineering in 2000, and the Dr.-Ing. degree from the Institute for Neural Computation, Ruhr-Universität Bochum, Germany. He has been a Chief Scientist with Honda Research Institute Europe GmbH since 2010, and responsible for the Complex Systems Optimization and Analysis Group of the institute. He joined the Future Technology Research Division, Honda R&D Europe (Deutschland) GmbH in 1998. since 2001, he has been working with Honda Research Institute Europe GmbH.