IEEE Open Journal of
Intelligent Transportation
Systems

Received 20 March 2024; revised 12 June 2024 and 17 July 2024; accepted 10 August 2024. Date of publication 13 August 2024; date of current version 25 September 2024.

Digital Object Identifier 10.1109/OJITS.2024.3443170

Multi-Objective Optimization of Urban Air Transportation Networks Under Social Considerations

NIKOLAS HOHMANN [1], SEBASTIAN BRULIN [2], JÜRGEN ADAMY [1], AND MARKUS OLHOFER [2]

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

[2] Complex Systems Optimization 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. The work of Nikolas Hohmann was supported by the Honda Research Institute Europe (HRI-EU).

This article has supplementary downloadable material available at https://doi.org/10.1109/OJITS.2024.3443170, provided by the authors.

ABSTRACT This work proposes and investigates a solution approach to the urban air transportation network optimization problem, considering the perspectives of different stakeholders, including societal interests. Given logistic hub positions and a set of optimized paths connecting them pairwise, we aim for a Pareto-optimal and three-dimensional air corridor network structure. This work demonstrates a way to merge the given paths into a network and provides a framework to optimize the network further regarding multiple objectives. It proposes three objective functions that evaluate the network from the economic perspectives of network providers and users and the city residents' social point of view. Using geospatial data from Frankfurt, Germany, we conducted different experiments including and excluding the social objective function under a varying input set of pre-optimized paths. Our analysis showed that taking social aspects into account results in traffic networks whose increase in social acceptance far outweighs the extra monetary costs. We conclude that it is beneficial to integrate social criteria into optimization problems when the solutions obtained are the basis for decisions in the area of conflict between the economy and human welfare.

INDEX TERMS Evolutionary algorithm, multi-objective, social, optimization, traffic network, transportation, UAM, UAV, unmanned aerial vehicle, urban, urban air mobility.

I. INTRODUCTION

THE RANGE of concepts for Urban Air Mobility (UAM) infrastructure design is broad. Over the last years, several different ideas and proposals have emerged from both government-supported (FAA [1], NASA [2], SESAR U-SPACE [3], DLR [4]) and industrial (Airbus [5], Embraer [6], Uber [7]) stakeholders. A good overview is provided in the review paper by Bauranov and Rakas [8].

Jang et al. [2] consider three types of airspace structures in their concept: sky-lanes, sky-tubes, and sky-corridors, each with varying degrees of freedom for the aerial vehicles. These structures aim to ensure safety while reducing the need for heavy investment in technology and infrastructure. Tests conducted on various structures demonstrate that a more structured environment increases safety and simplicity at the expense of throughput. Geister and Korn [4] argue that the greater the number of vehicles in an aerial mobility system, the stronger the requirement for adhering to pre-determined flight paths. Sunil et al. [9] propose different types of air spaces that are 1) full mix, 2) layers, 3) zones, and 4) tubes, ranging from completely unrestricted to fully restricted. Their simulation results show that a moderate separation like in the layer concept shows a better overall performance than the others regarding capacity, safety, and efficiency.

Other approaches from the literature examine especially flight corridor concepts in more detail. Denham et al. [10] investigate different corridor structures (i.e., line segment, intersection, roundabout) geometrically and in simulation

The review of this article was arranged by Associate Editor Fangfang Zheng.

concerning their capacity. Cummings et al. [11] examine different corridor network configurations on two scenarios concerning a throughput metric. Fedrigo [12] introduces three urban air traffic flow models. He considers the flow through a corridor structure, through a corridor with tracks, and the traffic flow while entering or exiting a corridor. Pang et al. [13] introduce the so-called *AirMatrix* concept that is a separation of the three-dimensional space into cubes. The cubes can be of different sizes allowing for a denser sampling in crowded low-speed airspace and a sparse sampling in open high-speed airspace. Corridors connect areas of different resolutions.

We emphasize two important observations that can be made when looking at recent literature on UAM infrastructure design.

1) *The social is secondary to the economic:* Bauranov and Rakas [8] notice that most evaluated airspace concept studies use idealized network representations that commonly evaluate the proposed airspace structures concerning economic quality criteria. In most cases, the aim is to increase capacity or throughput to allow as many vehicles as possible to fly simultaneously. In some cases, the studies are extended to include the aspect of safety, which leads to trade-off considerations between economic efficiency and safety aspects. Bauranov et al. criticize that social factors play a subordinate role in the examined Urban Air Mobility infrastructure design concepts. However, the noise produced by drones constitutes the major acceptance threshold for UAM transportation systems and thus for their success [14].
2) *Straight-line connections are a biased standard:* In the field of network optimization, traffic networks are most of the time modeled as graphs, with their edges representing (air)ways. Consequently, the (air)ways are usually assumed to be straight-line connections between logistic hubs [15]. Straight-line networks could be assumed energy-optimal. They might be a sufficient assumption when networks are optimized regarding economic objectives. However, they are biased as soon as a social objective is introduced.

Building on previous research [16], [17], we want to consider not only economic aspects but also social requirements in the design of aerial traffic networks. We therefore identify three requirements, each of which assesses the traffic network from a different point of view. Next to the costs to maintain the traffic network (network provider's behalf) and the time to traverse it (network user's behalf), we also optimize the transport network concerning social acceptance towards the noise generated by the traffic network's participants (network's residents' behalf). We are given a set of vertiports randomly distributed throughout the city. It is assumed that we have optimized paths between each pair of vertiports. Each path already has an assigned social attribute that quantifies its social compatibility. In the first step, called *Path-merging*, the single paths are combined into a graph that models the traffic network. The graph's nodes represent intersections and vertiports in the traffic network, whereas the edges represent aerial corridors. By construction, this graph contains redundant connections that might be unnecessary depending on someone's requirements for a traffic network. Therefore, in the second *Network Optimization* step, the traffic network is optimized regarding multiple formulated objectives by either removing edges from the graph or by re-integrating already deleted edges. Finally, we obtain a Pareto set of traffic networks that satisfy the formulated objectives to varying degrees. A decision maker can then choose a traffic network configuration that fits his or her needs. The selected network can be used as input to a routing and scheduling algorithm [18], [19] that calculates time windows for delivery tasks and trajectories through the network. This work thus forms the basis for unmanned aircraft system traffic management (UTM).

The contribution of this study is twofold and based on the two stated observations.

1) First, we integrate a social objective function into the network optimization process and investigate networks that fulfill economic and social objectives to identify the trade-off costs.
2) Secondly, we represent a traffic network as a graph, not exclusively consisting of straight-line corridors but also socially pre-optimized and therefore in general curved paths. As a second question, we want to assess the performance of this more detailed representation modeling, especially regarding the newly introduced social objective.

The remainder of this article is structured as follows. In Section II, we give an overview of related research in trajectory clustering and network optimization. In Section III, we then present our developed framework for *Path-merging* and *Network Optimization*. The evaluation results are presented and discussed in Section IV, before concluding in Section V.

II. RELATED WORK

Our approach combines a *Path-merging* with a *Network Optimization* step. In the following, we present literature from both domains.

A. PATH-MERGING

Trajectory clustering is a subdomain of the broad field of trajectory data mining. For a good overview, we refer the interested reader to the paper of Zheng [20]. The trajectory clustering problem searches for common movement patterns in (segmented) trajectory data, groups them into clusters of similar features, and - in some cases - eventually generates a representative trajectory for a found cluster. The term *trajectory* is not equally defined in all publications. For some, a trajectory is equivalent to a path, which is two or three-dimensional spatial point data. Others use it to refer to a path with time information. Researchers use trajectory clustering algorithms for example to analyze aircraft trajectories nearby airports [21], [22].

One famous trajectory clustering algorithm is TRACLUS by Lee et al. [23]. They propose a three-step approach that consists of a segmentation step, a clustering step, and a representation step. First, they partition paths into sub-paths by splitting them apart at designated characteristic points while smoothing the path segments. They minimize the approximation error between path and smoothed sub-paths and the number of sub-paths. A heuristic solves this two-objective optimization problem. Then, the line segments are grouped into clusters by applying a density-based clustering algorithm (adapted DBSCAN [24] algorithm). Last, a representative for each group of line segments is found by a geometrical approach that builds a representative path by computing average coordinates of intersected line segments when sweeping a vertical line alongside the cluster's major axis. However, the TRACLUS algorithm does not apply to our problem, as there still is the problem of deriving a graph from the clustered trajectories.

An example of a trajectory-to-graph conversion is the approach by Chen et al. [25] to find the most popular route between two locations by observing historical GPS data. In the first step, they calculate a transfer network from the trajectory data. The nodes of the transfer graph represent intersections in the trajectory data. The edges of the graph are then assigned transfer probabilities that increase with the edge's frequency of use for a given destination. Finally, the obtained weighted graph can be used to find the most popular route in a breadth-first search.

The map construction problem [26] can be seen as another application-related field of trajectory data mining. Given the GPS data of a vehicle traversing an unknown traffic map, it aims to reconstruct the graph of the underlying traffic map. Trajectory-to-graph conversion and map construction algorithms handle two-dimensional noisy path data to construct a graph. Thus, these approaches use two-dimensional geometric calculations that can not be applied to three-dimensional spatial paths as we handle them. Nevertheless, these approaches inspired us to develop a *Path-merging* strategy tailored to our needs, which is presented in Section III-B.

B. NETWORK OPTIMIZATION

A good overview of the literature in the domain of network optimization can be found in the article by Ding et al. [27]. However, they emphasize urban *ground* traffic networks. For example, Gastner and Newman [28] take a set of initial logistic hub positions as given and create two-dimensional traffic networks by minimizing the average distance between all node pairs subject to a restricted budget. Li et al. [29] also assume a set of initial node positions that are additionally structured in a lattice graph. They examine how the insertion of edges between non-neighboring nodes improves the overall travel time through the network. Instead of the network structure, Chen et al. [30] optimize the flow through the network by adapting the edge capacities in a bi-objective optimization problem regarding total travel time and construction costs as objectives.

To our knowledge, there is little research on UAM, thus *aerial*, traffic network optimization. Related work can be found by He et al. [31], who build multiple dependent air traffic routes by sequentially running a shortest path algorithm for different start and end points. The planning of every new path considers the previously planned paths to not interfere with them. The paths are optimized regarding UAV energy consumption, risk, and airspace occupancy, which means that the algorithm tries to overlap different air corridors' buffer zones. However, there is no optimization on the network level: A pair of vertiports will always be connected by an isolated path. There is no feature to merge spatially shared path segments of two paths into a combined air corridor. In our work, we aim for a Pareto-optimal network structure instead of a set of unconnected paths, which has several advantages. First, by identifying common flight paths between several vertiports and replacing direct connections, the aerial corridor length of the complete network and thus the infrastructure costs can be reduced. Second, adding a new vertiport to the UAM system does not require planning many new paths for every existing vertiport. Rather, the new vertiport must only be connected to the nearest corridor in the existing traffic network.

In the research area of multi-agent navigation, Mai et al. [32] optimize a two-dimensional network. Instead of assuming logistic hubs as starting points, they aim for a homogeneous coverage of the entire design space. As there is no cost for the presence of an edge, the network they generate does not represent a physical transportation network but rather virtual connection possibilities. Here, the design principle of considering the entire airspace as a design space applies. We take a different design approach and pursue a growing thus adaptive design concept. Considering governmental approval processes and the gradual gaining of acceptance by the public, we believe it is very likely that UAM pilot projects will only realize single routes between a few logistic hubs in the city. Then, these individual routes could successively be expanded to include more vertiports and new connections. Slowly, a transportation network would emerge spanning larger parts of the city. For this reason, we propose an additive design of the aerial traffic network in our work, which finds its origin in individual flight paths assembled into a network.

The insights we gained from the presented studies are all incorporated into the design of our multi-objective network optimization framework. To sum up, our approach differs from others presented in that it optimizes a *three-dimensional* spatial network, takes *social criteria* into account, allows *curved paths* instead of straight lines as network corridors, starts the optimization with an *initial pre-calculated graph* (derived from *Path-merging*) instead of a set of nodes, and is conceived as a *growing network design* process.

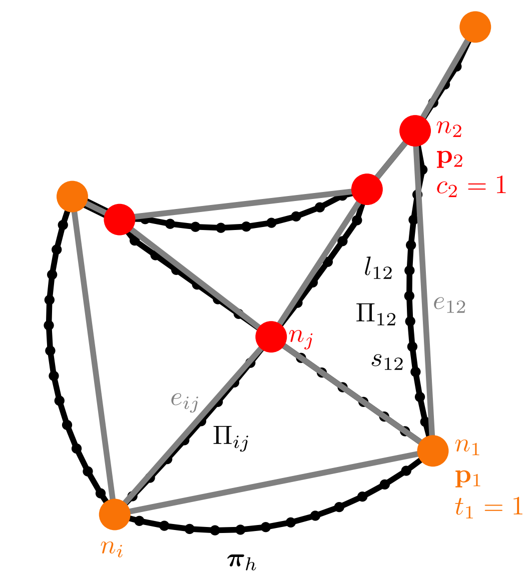


FIGURE 1. Visualization of the graph structure $G = \{N, E\}$ used as traffic network representation. All nodes $n_i \in N$ of the graph are shown as large circles, with vertiport nodes ($t_i = 1$) in orange and crossing nodes ($c_i = 1$) in red. Moreover, each node is assigned a three-dimensional spatial position $\mathbf{p}_i$. The edges $e_{ij} \in E$ of the graph are gray. The 3D path information Π_{ij} stored in each edge is shown in black. Each path has a length l_{ij} and a social attribute s_{ij}. The small black dots represent the waypoints $\boldsymbol{\pi}_h$ of the paths.

III. SYSTEM MODEL

We begin by describing the tackled problem in Section III-A, before we present the proposed *Path-merging* approach in Section III-B. The objective functions are then introduced in Section III-C and the constraints in Section III-D. These form the *Network Optimization* problem, whose solution is described in Section III-E.

A. PROBLEM DEFINITION

Let us assume a continuous and cubic space $\mathbb{D} = \left[x_{\min} \;\; x_{\max}\right] \times \left[y_{\min} \;\; y_{\max}\right] \times \left[z_{\min} \;\; z_{\max}\right] \subset \mathbb{R}^3$. We represent a traffic network utilizing a three-dimensional spatial graph representation $G = \{N, E\}$ visualized as two-dimensional projection in Fig. 1. A graph G consists of a set of nodes $N = \{n_1, \ldots, n_{|N|}\}$ and a set of $|E|$ edges $E = \{e_{ij}\}$. Every node n_i is assigned information

- about its three-dimensional spatial position $\mathbf{p}_i = \left[x_i \; y_i \; z_i\right] \in \mathbb{D}$,
- whether it represents a vertiport

$$t_i = \begin{cases} 1, \text{ if vertiport}, \\ 0, \text{ else}, \end{cases} \tag{1}$$

- and whether it represents a crossing in the network

$$c_i = \begin{cases} 1, \text{ if crossing}, \\ 0, \text{ else}. \end{cases} \tag{2}$$

An edge $e_{ij} = (n_i, n_j)$ is undirected and connects two nodes. Every edge holds information

- on a path Π_{ij} that connects $\mathbf{p}_i$ and $\mathbf{p}_j$ with a sequence of three-dimensional points $\boldsymbol{\pi} \in \mathbb{D}$

$$\Pi_{ij} = \left[\boldsymbol{\pi}_1 = \mathbf{p}_i, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_{|\Pi_{ij}|} = \mathbf{p}_j\right], \tag{3}$$

- and on a social cost weight s_{ij}.

Note that there may exist more than one edge between two nodes. Furthermore, we denote the Euclidean distance between two adjacent nodes n_i and n_j as $|e_{ij}| = ||\mathbf{p}_i - \mathbf{p}_j||_2$, whereas the length of the underlying path Π_{ij} is labeled

$$l_{ij} = \sum_{k=2}^{|\Pi_{ij}|} ||\boldsymbol{\pi}_k - \boldsymbol{\pi}_{k-1}||_2. \tag{4}$$

We want to emphasize that in general $|e_{ij}| \neq l_{ij}$ yields. Unless otherwise stated, edge length in the following refers to l_{ij}.

We start with an operational space $\mathbb{D}$ and a set of vertiport positions $N_\text{T} = \{\mathbf{p}_{\text{T},1}, \ldots, \mathbf{p}_{\text{T},|N_\text{T}|}\}$, determining where transport agents can take off and land in the network. Defining all $|C| = \binom{|N_\text{T}|}{2}$ vertiport pair combinations (without repetition) as $C = \{(n_k, n_l) \,\forall\, n_k \neq n_l \in N_\text{T}\}$, we assume to have already optimized paths Π_i between every vertiport pair $(n_k, n_l) \in C$ following a many-objective path planning approach [33]. Given this set of optimized paths $P = \{\Pi_1, \ldots, \Pi_{|C|}\}$, we pursue a two-fold approach in this work. In the first *Path-merging* step, described in Section III-B, we derive a network representation (i.e., a graph G) from the set of paths. In the second *Network Optimization* step, as proposed in the following Section III-C, III-D, and III-E, our goal is to find a set of traffic networks that approximate the Pareto front of a multi-objective optimization problem, which consists of $E = 3$ objectives and $F_\text{eq} = 1 + |N_\text{T}|$ equality constraints.

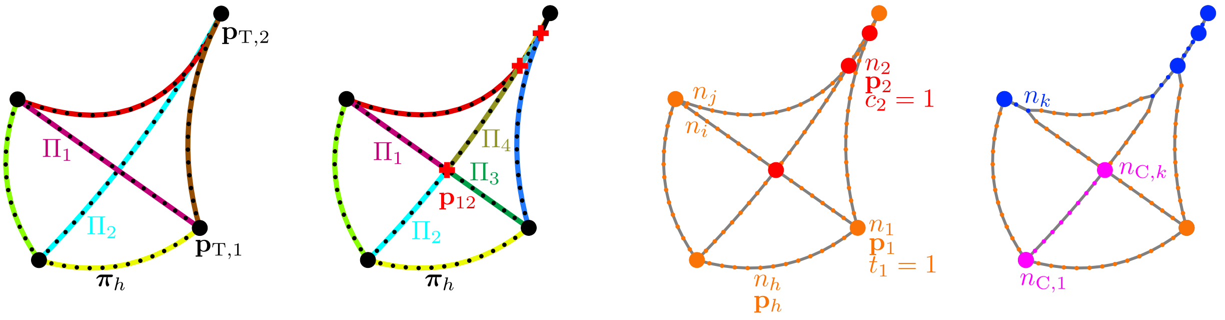
B. PATH-MERGING

The given input set of three-dimensional spatial paths P, as visualized in Fig. 2a, consists of connections between all vertiport pairs whose paths have already been pre-optimized. Thus, they often overlap and form a network structure, at least visually (see the red, blue, and brown paths in Fig. 2a). In the *Path-merging* step, we want to model this network structure by calculating a spatial graph representation G, as shown in Fig. 1, as output. The *Path-merging* process consists of five steps, each of which is explained in the following subsections. We denote paths in the initial path set $\Pi_i \in P$ with a single index, not to be confused with the path segments Π_{ij} (double index) in the derived graph G.

We assume that the paths form centerlines of cylindrical tubes (i.e., corridors) of diameter d_P in which UAVs can move.

1) INTERSECTION FINDING

We start with an initial set of paths P as shown in Fig. 2a. Where two paths intersect, the later network should be nested. Therefore, we calculate the intersection points $\mathbf{p}_{ij} \in \mathbb{D}$ between all paths Π_i and Π_j in P. We assume that two



(a) Exemplary colored paths Π_i between all pairs of vertiport positions $\mathbf{p}_{\mathrm{T},k}$ (black circles). The paths build the initial set of paths P, which is input into the *Path-merging* process. A path consists of several waypoints $\boldsymbol{\pi}_h$ marked as black dots.

(b) The same set of paths P after the intersection finding step of the *Path-merging* process. Paths has been cut apart at identified crossing positions $\mathbf{p}_{ij}$ marked as red crosses.

(c) The graph G after the graph conversion step. All round elements are nodes. Every node n_h stores the position information of the former waypoint $\boldsymbol{\pi}_h$ in $\mathbf{p}_h$. Orange circles visualize vertiport nodes ($t_i = 1$), and red circles show crossing nodes ($c_i = 1$). Nodes are connected by edges shown in gray.

(d) The graph G after the node contraction step. Nodes that are too close together (e.g., n_i and n_j in Fig. 2c) have been contracted and now build a new node (e.g., n_k) visualized in blue. One exemplary linear chain from node $n_{\mathrm{C},1}$ to node $n_{\mathrm{C},k}$ is highlighted in pink.

FIGURE 2. Visualization of the *Path-merging* process applied on an exemplary set of paths. For reasons of clarity, not all elements are labeled. Vertiport positions are visualized as black circles, and intersection positions as red crosses. In the set of paths ((a) and (b)), the single paths are colored differently, their waypoints are black dots. Generally in the graphs ((c) and (d)), vertiport nodes are visualized as orange circles, crossing nodes as red circles. All other nodes are orange dots. The graphs' edges are gray lines.

paths intersect when the closest distance between both falls below d_{P}, which means that their cylindrical shells intersect. If two paths intersect, the intersection point $\mathbf{p}_{ij}$ is included in both paths as a waypoint $\boldsymbol{\pi}$, and the two intersecting paths are split up into four separate paths as visualized in Fig. 2b.

2) GRAPH CONVERSION

In the next step, the path set, as visualized in Fig. 2b, is transferred into a graph structure by introducing a node n_h with $\mathbf{p}_h = \boldsymbol{\pi}_h$ for every waypoint $\boldsymbol{\pi}_h$. Only one node is created, if several paths share the same waypoint (e.g., at a vertiport position). Also, the vertiport flags t_i and crossing flags c_i are set accordingly. An edge e_{ij} is created for every pair of neighboring waypoints $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_{i+1}$ for every path in P, whereas the edge information Π_{ij} and s_{ij} remains empty for now. Figure 2c shows the resulting graph.

3) NODE CONTRACTION

The graph structure, as shown in Fig. 2c, may contain nodes that were assigned to two separate path segments before the graph conversion, but whose distance is less than the corridor diameter d_{P}. Thus, the nodes can be assigned to one and the same corridor and are therefore merged (i.e., contracted). As an example, this is visualized by the nodes n_i and n_j in Fig. 2c. Two nodes n_i and n_j collapse into a single one n_k, when $||\mathbf{p}_i - \mathbf{p}_j||_2 < d_{\mathrm{P}}$ applies. If one of the nodes to be contracted is a vertiport node or a crossing node the position of the new node n_k becomes that of the vertiport or crossing node. Otherwise, $\mathbf{p}_k$ is calculated by the means of

$$\mathbf{p}_k = \bar{X},\ \text{where}\, X = \{\mathbf{p}_l \,\forall\, n_l \in G[n_i] \cup G\big[n_j\big]\}, \tag{5}$$

and $G[n]$ denotes the neighborhood of n in G. The exemplary graph after node contraction is visualized in Fig. 2d in which the contracted nodes are marked in blue.

4) GRAPH REDUCTION

The graph, as in Fig. 2d, now contains sequences of nodes where each inner node n has only two adjacent edges, i.e., degree $d(n) = 2$. We call such a sequence linear chain $G_{\mathrm{C}} = \{N_{\mathrm{C}}, E_{\mathrm{C}}\}$ with $E_{\mathrm{C}} = \{(n_{\mathrm{C},1}, n_{\mathrm{C},2}), (n_{\mathrm{C},2}, n_{\mathrm{C},3}), \ldots, (n_{\mathrm{C},k-1}, n_{\mathrm{C},k})\}$ in the following. An exemplary linear chain is visualized with pink nodes in Fig. 2d. The optimizer will later remove single, gray visualized edges from the graph. But, removing an edge from a linear chain would result in the unnecessary creation of two dead-ends in the transport network. Instead, it would be meaningful if the optimizer removes a complete linear chain. Therefore, we resolve linear chains into a single new edge $e_{ij} = (n_{\mathrm{C},1}, n_{\mathrm{C},k})$. This means that the linear chain's nodes are deleted (and thus its edges). However, it is important to note that the position information of the linear chain's nodes is stored in the new edge's path $\Pi_{ij} = [\mathbf{p}_{\mathrm{C},1}, \mathbf{p}_{\mathrm{C},2}, \ldots, \mathbf{p}_{\mathrm{C},k}]$ of length l_{ij} and is thus still available. Reducing all linear chains in the exemplary graph in Fig. 2d ultimately leads to the desired graph structure shown in Fig. 1. Its gray-colored edges e_{ij} are subject to the later *Network Optimization*. Its paths' spatial information Π_{ij} (black lines) can be used to retrieve the 'original', i.e., non-reduced, curved network after the *Network Optimization*.

5) GRAPH FINALIZING

Each path Π_k in the input set of paths P originally had a social attribute s_k. However, the paths were spatially modified in the steps 1) until 4) of the *Path-merging* procedure, so the social attributes s_{ij} of each path segment Π_{ij} must now be recomputed. The calculation itself is explained in Section III-C3. In the following, we call the final network, i.e., the output of the *Graph-merging*, initial graph $G_0 = \{N_0, E_0\}$ because it serves as input into the *Network Optimization*. The set of vertiport positions can now be expressed as $N_\mathrm{T} = \{\mathbf{p}_i \,\forall\, n_i \in N_0 \,|\, t_i = 1\}$.

C. OBJECTIVES

Here, we introduce the objectives for evaluating a traffic network during *Network Optimization*. The objectives are designed to assess the quality of a traffic network from the perspective of a traffic network operator (maintenance), a traffic network user (travel cost), and the residents living underneath the traffic network (social cost). Beforehand, we give some definitions for objective function normalization. We introduce the sum of edge lengths in the initial graph G_0 as $L_0 = \sum_{e_{ij} \in E_0} l_{ij}$, and the sum of social cost weights as $S_0 = \sum_{e_{ij} \in E_0} s_{ij}$. Furthermore, given a set of terminal nodes $N_\tau \subseteq N$ in a graph $G = \{N, E\}$ with a set of weighted edges E, a Steiner tree [34] $G_\mathrm{ST} = \{N_\mathrm{ST} \subseteq N, E_\mathrm{ST} \subseteq E\}$ is a connected subset of G that contains all terminal nodes $N_\tau \subseteq N_\mathrm{ST}$ and minimizes the sum of its edges' weights $\min \sum_{e_{ij} \in E_\mathrm{ST}} w_{ij}$. Setting the set of vertiport nodes as terminal nodes $N_\tau = N_\mathrm{T}$, we denote the initial graph's Steiner tree regarding ende length $G_\mathrm{ST,L} = \{N_\mathrm{ST,L}, E_\mathrm{ST,L}\}$. Summing its edges' length attributes, yields $L_\mathrm{ST,L} = \sum_{e_{ij} \in E_\mathrm{ST,L}} l_{ij}$. Moreover, we introduce $\mathcal{R}_{\{\cdot\}}(G, n_k, n_l)$ as operator (e.g., Dijkstra's algorithm [35]) that finds the shortest route from n_k to n_l regarding an edge attribute $\{\cdot\} \in \{l, s, |e|\}$ in a graph G, resulting in a sequence of edges $R_{G,n_k,n_l,\{\cdot\}}$.

1) MAINTENANCE COST

A UAM traffic network equipped with monitoring and potentially also navigation technology allows the operation of UAVs with few sensors of their own. Therefore, we assume in our optimization problem that a larger network means more maintenance and operating costs for the network operator. Consequently, we define the maintenance cost objective function that evaluates a graph $G = \{N, E\}$ as

$$f_\mathrm{M}(G) = \frac{\sum_{e_{ij} \in E} l_{ij} - L_\mathrm{ST,L}}{L_0 - L_\mathrm{ST,L}}. \tag{6}$$

The normalization constants L_0 and $L_\mathrm{ST,L}$ guarantee that $0 \leq f_\mathrm{M}(G) \leq 1$ applies.

2) TRAVEL COST

To assess the network's capability to route traffic efficiently, we adopt a well-known metric from the graph domain [36]: The route factor (also detour index) between two nodes n_k and n_l in a graph can be expressed as the quotient of the shortest route cost through the graph $r_{|\mathrm{e}|}(G, n_k, n_l) = \sum_{e_{ij} \in R_{G,n_k,n_l,|e|}} |e_{ij}|$ and the direct Euclidean distance between n_k and n_l

$$Q(n_k, n_l) = \frac{r_{|\mathrm{e}|}(G, n_k, n_l)}{||\mathbf{p}_k - \mathbf{p}_l||_2}. \tag{7}$$

Thus, there is an efficient (i.e., direct) connection between the two nodes, if $Q(n_k, n_l) = 1$ applies. The longer the detour, the greater the route factor. The considered traffic networks generally do not have straight connections between two nodes, but curved paths. In the route factor definition (7), we, therefore, replace the Euclidean distance between n_k and n_l with the shortest route cost between both in the initial graph G_0. Furthermore, by forming the negative reciprocal of (7) and adding one, we obtain an objective function having its best value in zero and its worst in one. Finally, for our travel cost objective function, we average the adapted route factor definition for all connections C yielding

$$f_\mathrm{T}(G) = \frac{1}{|C|} \sum_{(n_k,n_l) \in C} \frac{r_\mathrm{l}(G, n_k, n_l) - r_\mathrm{l}(G_0, n_k, n_l)}{r_\mathrm{l}(G, n_k, n_l)} \tag{8}$$

which ensures $0 \leq f_\mathrm{T}(G) \leq 1$. The function $r_\mathrm{l}(G, n_k, n_l) = \sum_{e_{ij} \in R_{G,n_k,n_l,l}} l_{ij}$ sums the length attributes of the shortest path's edges from n_k to n_l through G. The smaller the travel cost values $f_\mathrm{T}(G)$ of a graph G, the shorter its established connections between all vertiport pairs compared to the shortest connections in the initial graph G_0.

3) SOCIAL COST

We proceed similarly with the design of the social cost objective function. For each connection $(n_k, n_l) \in C$, we compare the most socially acceptable route through G with the social optimum through the initial graph G_0 resulting in

$$f_\mathrm{S}(G) = \frac{1}{|C|} \sum_{(n_k,n_l) \in C} \frac{r_\mathrm{s}(G, n_k, n_l) - r_\mathrm{s}(G_0, n_k, n_l)}{r_\mathrm{s}(G, n_k, n_l)}, \tag{9}$$

with $r_\mathrm{s}(G, n_k, n_l) = \sum_{e_{ij} \in R_{G,n_k,n_l,s}} s_{ij}$ being the totalized social cost values of the most social path from n_k to n_l through G. The relationship $0 \leq f_\mathrm{S}(G) \leq 1$ also applies here. A decreasing social cost value $f_\mathrm{S}(G)$ means that the graph G establishes connections between vertiport pairs that are perceived as less noisy. The social cost weight s_{ij}, which is the core of the objective function, can generally be seen as a placeholder for any quantifiable social criterion such as privacy or safety [37]. In our work, it represents the aviation noise that city residents perceive near a flight path. We calculate s_{ij} for every path Π_{ij} by applying a simplified noise model to it. The noise model used is deduced from the inverse-square law according to which the noise intensity decreases with increasing distance and can be taken from a previous work [33] in detail. Generally speaking, a flight path has a lower (i.e., better) social cost weight if it tends to run at higher altitudes, or over ground traffic roads as the drone noise then vanishes in the ground traffic noise [38].

D. CONSTRAINTS

During *Network Optimization*, the optimizer will delete edges from the initial graph G_0 to create new traffic networks. This might result in graphs G that do not meet the requirements of a reasonable transportation network. We formulate these requirements in the following equality constraint functions.

1) CONNECTIVITY

A connected component H is a subset of an undirected graph G with a path between any pair of nodes. A graph can thus be written as a union of its $|H|$ connected components $G = H_1 \cup H_2 \cup \ldots \cup H_{|H|}$. For a traffic network, we therefore ensure its connectivity by defining the connectivity equality constraint

$$h_{\mathrm{C}}(G) = |H| - 1 \overset{!}{=} 0. \tag{10}$$

2) VERTIPORT INCLUSION

When edges are deleted from a graph during optimization, the resulting graph may no longer include important nodes (i.e., vertiport nodes) while still being connected. We define a binary variable b_i for every node $n_i \in N_0$ indicating whether the node is included in the graph G ($b_i = 1$) or not ($b_i = 0$). Incorporating $|N_{\mathrm{T}}|$ equality constraint functions

$$h_{\mathrm{T},j}(G) = b_j - 1 \overset{!}{=} 0, \quad \forall \arg n_j \in N_{\mathrm{T}} \tag{11}$$

into the optimization problem, we ensure that all vertiport nodes are part of the traffic network.

E. EVOLUTIONARY ALGORITHM DESIGN

For the *Network Optimization*, the initial graph G_0 is encoded into a binary optimization vector

$$\mathcal{E}(G_0) = \mathrm{v}_0 = \left[v_1 = 1 \;\; v_2 = 1 \;\ldots\; v_{|E_0|} = 1\right] \tag{12}$$

of length $|E_0|$. Inversely to the encoding operator $\mathcal{E}$, any solution v can again be decoded into the respective graph by applying the decoding operator $G = \mathcal{D}(\mathrm{v}) = \mathcal{E}^{-1}(\mathrm{v})$. To obtain a new network G, the optimizer can either delete an edge e_{ij} by setting the corresponding $v_k = 0$ or adding a previously deleted edge by setting $v_k = 1$ again. The complete optimization problem tackled in this article can be described as

$$\min_{\mathrm{v} \in \mathbb{V}} \begin{cases} f_{\mathrm{M}}(\mathrm{v}), \\ f_{\mathrm{T}}(\mathrm{v}), \\ f_{\mathrm{S}}(\mathrm{v}), \end{cases} \tag{13}$$

$$\text{s.t.} \begin{cases} h_{\mathrm{C}}(\mathrm{v}), \\ h_{\mathrm{T},j}(\mathrm{v}), \quad \forall \arg n_j \in N_{\mathrm{T}}. \end{cases} \tag{14}$$

For the sake of simplicity, the notations $f(\mathrm{v})$ and $f(G)$ are used interchangeably in the following without explicitly mentioning decoding. Since the formulated optimization problem is multi-objective and non-convex, we apply the metaheuristic evolutionary algorithm NSGA3 [39] provided by the `pymoo` framework [40]. A flowchart of the evolutionary optimization procedure is depicted in Fig. 3. The associated initialization routine, mutation, and repair operator are presented in the following. With the selected encoding (12), a crossover operator generates non-feasible solutions (e.g., unconnected networks with dead ends) with a very high probability. These solutions would have to be repaired with high computational effort. Since those repaired solutions can also be calculated with the mutation operator, a crossover operation is bypassed setting its probability of occurrence in our framework to $p_{\mathrm{x}} = 0$.

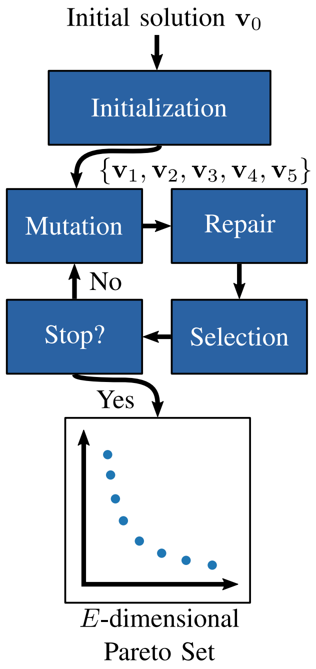

Initial solution $\mathbf{v}_0$
Initialization
$\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{v}_5\}$
Mutation
Repair
No
Stop?
Selection
Yes
E-dimensional
Pareto Set

FIGURE 3. Flowchart diagram of the utilized evolutionary algorithm. From the encoded initial graph (i.e., v_0), the solutions $\{\mathrm{v}_1, \ldots, \mathrm{v}_5\}$ are generated, which are input into the iterative optimization. During the mutation, new solutions are generated by either removing edges or re-inserting them (if they are contained in G_0). Constraint-violating networks will be repaired. During selection, the networks are evaluated by calculating their objective function values. By applying a selection scheme to all new and old solutions, the next generation of solutions is obtained. If the termination criterion does not hold, the process is repeated. Otherwise, the result is returned, being a Pareto set of trade-off networks.

1) INITIALIZATION

The optimization is initialized with five solutions $\{\mathrm{v}_1, \ldots, \mathrm{v}_5\}$ with $\mathrm{v}_{\mathrm{i}} = \mathcal{E}(\mathrm{G}_{\mathrm{i}})$ that are expected to be (near-)optimal for at least one objective:

(a) The Steiner tree regarding the length attribute $G_1 = G_{\mathrm{ST,L}}$ obtains optimal values for the maintenance objective function $f_{\mathrm{M}}(G_1) = 0$.

(b) The Steiner tree regarding the social attribute $G_2 = G_{\mathrm{ST,S}}$ is expected to be an interesting initial solution as it contains only socially favored edges while being characterized by small maintenance costs.

(c) To obtain the Pareto set's extreme point regarding travel cost (i.e., $f_{\mathrm{T}} = 0$), we introduce a graph that unites the shortest paths (regarding l) between all vertiport pair combinations C, yielding $G_3 = \{N_3, E_3\}$ with $E_3 = \bigcup_{(n_k,n_l) \in C} R_{G_0,n_k,n_l,l}$.

(d) In the same way, we add the union of all connections' shortest paths through G_0 regarding the social attribute to the initial solutions $G_4 = \{N_4, E_4\}$ with $E_4 = \bigcup_{(n_k,n_l) \in C} R_{G_0,n_k,n_l,s}$. This gives us the extreme point regarding social cost $f_{\mathrm{S}}(G_4) = 0$.

(e) Last, we can calculate another non-dominated solution by taking the union of G_3 and G_4 as an initial solution, being $G_5 = \{N_5, E_5\}$ with $E_5 = E_3 \cup E_4$. This solution is characterized by $f_T(G_5) = f_S(G_5) = 0$, but consists of more edges than G_3 and G_4 and is, therefore, worse regarding the maintenance cost.

2) MUTATION

The mutation process consists of two parts.

(a) Firstly, an individual (i.e., a solution v) is subject to a delete mutation with a probability of $p_{\text{mut,del}}$. The delete mutation removes an edge e_{ij} from $\mathcal{D}(\text{v})$ with a probability of $p_{\text{mut,ind}}$. We set the probability of deleting an edge inversely proportional to the number of edges in the network $p_{\text{mut,ind}} = 1/\sum_{\text{v}} v_k$. This is a reasonable choice, as each additional edge deletion means a larger jump in the search space potentially overshooting a desired minimum in the objective space.

(b) Secondly, an add mutation is applied to an individual with a probability of $p_{\text{mut,add}}$. This mutation randomly draws from the set of connections $(n_k, n_l) \in C$ and then randomly either adds the shortest path $R_{G_0,n_k,n_l,l}$ or the best path regarding the social attribute $R_{G_0,n_k,n_l,s}$ to the graph $\mathcal{D}(\text{v})$.

3) REPAIR OPERATOR

During optimization, the evolutionary algorithm creates solutions that may be improved or made feasible by incorporating domain knowledge. This can be done by defining a repair operator. It repairs infeasible solutions or solutions that must have an obvious (induced from the graph structure) similar solution that Pareto dominates the solution to be repaired.

(a) We start by repairing dead-ends that emerged in the traffic network during optimization. Given a node n_i in the graph G that is a dead-end (i.e., its degree is $d(n_i) = 1$) but no vertiport ($t_i = 0$), we delete the node n_i together with the connected linear chain.

(b) Then, we repair solutions that do not satisfy the connectivity constraint (10) by determining the biggest connected component H_0 in terms of number of contained vertiports. We connect every other connected component H_i to H_0 by choosing a random vertiport node n_i in H_i (or if there is none, any random node) and connect it via the shortest possible path to the nearest vertiport in H_0. Note that the terms “shortest” and “nearest” apply with a probability of 50% concerning the length or the social attribute of the graph’s edges.

(c) Finally, we repair solutions that violate the vertiport inclusion constraint (11). For that to happen, any vertiport that is not part of the current graph $\mathcal{D}(\text{v})$ is again connected to the graph by either the length-related or socially shortest path to the nearest vertiport in $\mathcal{D}(\text{v})$.

TABLE 1. Parameters.

Parameter	Symbol	Value
Add mutation probability	$p_{\text{mut,add}}$	0.5
Delete mutation probability	$p_{\text{mut,del}}$	0.5
Individual delete mut. prob.	$p_{\text{mut,ind}}$	$1/\sum_{\text{v}} v_k$
Crossover probability	p_{x}	0
Population size	s_{pop}	100

TABLE 2. Scenario Specifications.

Parameter	Symbol	Value
Origin latitude	Lat	50.1002 °N
Origin longitude	Long	8.666 °E
Map dimensions	$x_{\text{min}},\ x_{\text{max}}$	0 m, 3010 m
	$y_{\text{min}},\ y_{\text{max}}$	0 m, 3000 m
	$z_{\text{min}},\ z_{\text{max}}$	0 m, 300 m
Corridor diameter	d_{P}	5 m
Number of vertiports	$\lvert N_{\text{T}}\rvert$	16
Vertiport pair combinations	$\lvert C\rvert$	120

IV. EVALUATION

In this section, we describe the experiments conducted to answer the research questions 1), and 2) presented in Section I. Therefore, in Section IV-A, we explain the scenario setup, before describing the different experiments in Section IV-B. In Section IV-C, we present and discuss the results of the experiments.

A. SETUP

We describe our evaluation’s setup consisting of the parameters of the proposed optimization algorithm in Section IV-A1, and an exemplary city where the aerial traffic network is planned in Section IV-A2.

1) ALGORITHM PARAMETERS

We provide the optimization algorithm’s parameters in Table 1. The gray highlighted parameters are default parameters of the used state-of-the-art algorithm [39]. The rest of the parameters were determined empirically.

2) SCENARIO

As an example city for our evaluation, we choose a section of Frankfurt that is visualized in Fig. 4. The $|N_T|$ vertiports, visualized as orange circles, were randomly distributed over the entire area excluding water areas as possible locations. Any specifications considering the chosen scenario can be drawn from Table 2.

B. EXPERIMENTS

For the same given vertiport locations, we conduct three experiments that are three independent *Path-merging* and *Network Optimization* runs with varying settings. We run the network optimizations in all three experiments for 2000 iterations, but the optimizations converge earlier. We adopt the termination criterion from Blank and Deb [42] to determine convergence. Table 3 gives an overview of

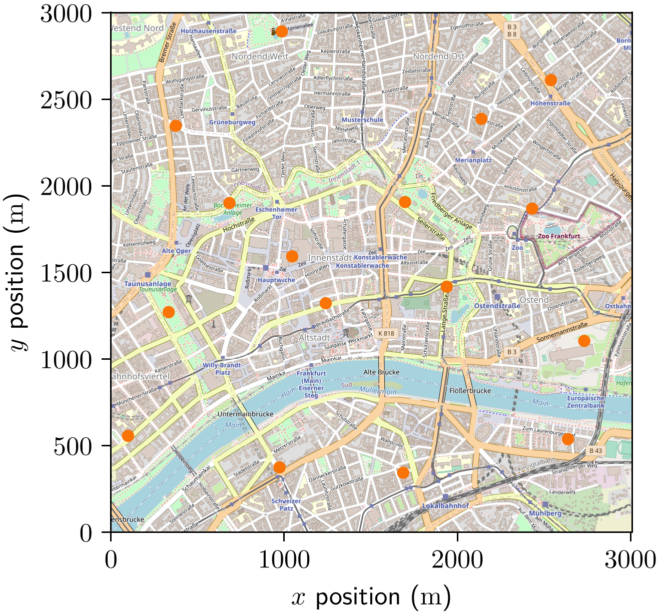


FIGURE 4. The map extract of Frankfurt [41] that was used to calculate the set of paths. The orange points denote the vertiport positions that were randomly sampled within the map. The z-component of the vertiport positions corresponds to the height of the building at the respective location.

TABLE 3. Experiments & Convergence.

Exp.	Initial path set	E	Objectives	Iterations until Convergence
S2	Straight	**2**	f_{M}, f_{T}	705
S3	Straight	**3**	f_{M}, f_{T}, f_{S}	809
SC3	**S**traight & **C**urved	**3**	f_{M}, f_{T}, f_{S}	1074

the different runs and the number of iterations until they converge respectively.

1) In the first experiment, abbreviated **S2** in the following, the initial path set P, which is input into the *Path-merging*, mainly consists of straight-line connections between the vertiports. Figure 5 shows a two-dimensional projection of the complete path set after *Path-merging*. The paths were derived by $|C|$ independent energy-optimal path optimizations [33]. The curves of some paths are caused by flying around static obstacles (i.e., higher buildings). During the *Network Optimization* step, the aerial traffic networks are only optimized regarding the two economic objectives f_{M} (maintenance cost) and f_{T} (travel cost).
2) In the second experiment **S3**, the *Path-merging* is conducted on the same set of straight paths. However, in the *Network Optimization* step, we plug the social criterium f_{S} as the third objective function into the optimization problem. The additional dimension in the objective space results in an expected increase in convergence time, which is 15 % in this case.
3) In the third experiment **SC3**, the *Path-merging* is initialized with $2|C|$ paths, namely the $|C|$ straight paths as used before and additional $|C|$ paths that had been optimized regarding the social cost weight. We obtained these by conducting $|C|$ independent noise-optimal path optimizations [33] based on the city map shown in Fig. 4. To differentiate these pre-optimized paths from straight paths, we refer to them as social or curved paths. A two-dimensional projection of the set of curved paths after *Path-merging* is visualized in Fig. 6. Due to the larger search space compared to the experiment **S3**, the optimization takes 33 % longer to converge.

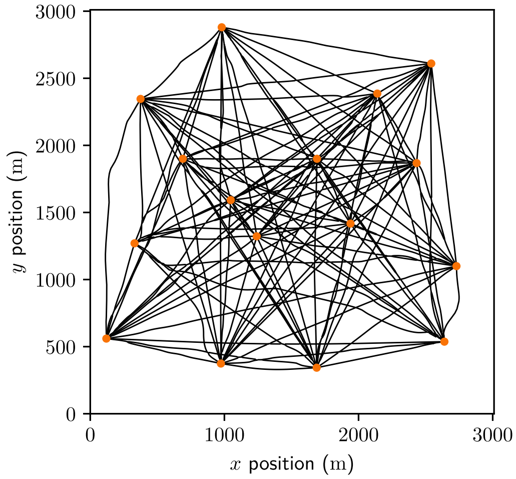


FIGURE 5. The projected straight-path network obtained after applying the *Path-merging* process to the set of paths that connects all vertiports with straight lines while circumventing static obstacles.

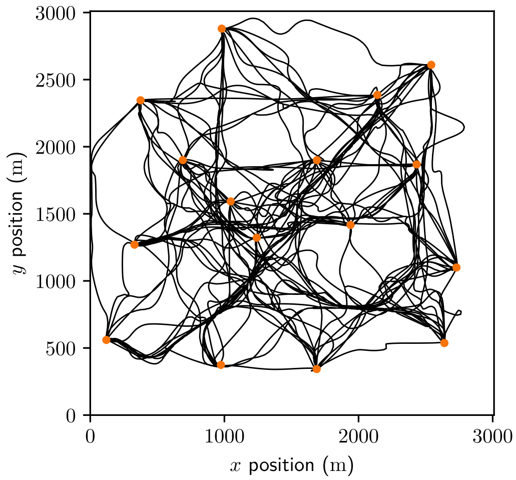


FIGURE 6. The projected curved-path network obtained after applying the *Path-merging* process to the set of paths that was pre-optimized regarding social cost weights based on the city map shown in Fig. 4.

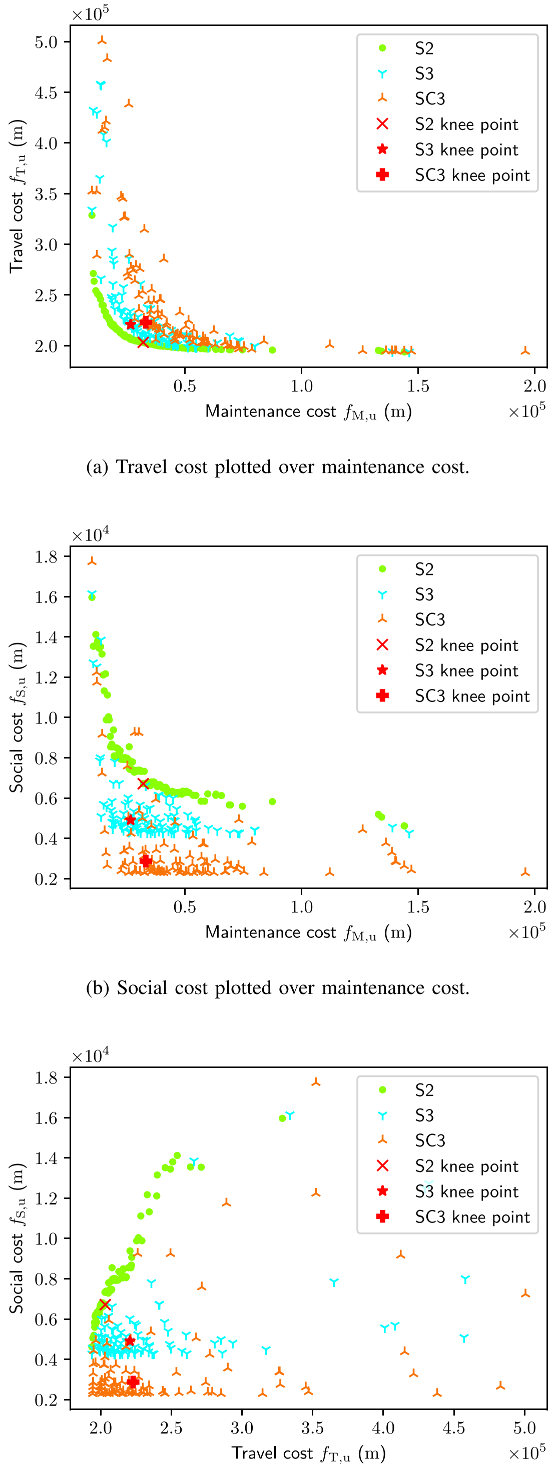
C. RESULTS & DISCUSSION

We now compare the results from the three experiments to obtain answers to the research questions posed in this article's introduction. For a fair cross-experiment comparison, the unnormalized objective functions $f_{\mathrm{M,u}}$, $f_{\mathrm{T,u}}$, and $f_{\mathrm{S,u}}$ must be used. These correspond to the objective functions (6), (8), and (9) without the normalization constants. All three of these unnormalized cost functions are given in meter since $f_{\mathrm{M,u}}$ is the sum of all path lengths, $f_{\mathrm{T,u}}$ adds up the path lengths of the shortest paths between all vertiport pairs, and $f_{\mathrm{S,u}}$ sums edge attributes s_{ij} that originate from a path integral over a unitless grid map [33].

1) ECONOMIC VS. SOCIO-ECONOMIC NETWORK OPTIMIZATION

First, we want to investigate the effects of including a social objective function f_{S} into the multi-objective network optimization problem. This social factor is often neglected in related work. What effect does this have on the obtained Pareto sets of traffic networks? We, therefore, compare the traffic networks obtained in **S2** with those in **S3**. As this means comparing a two-dimensional Pareto set with a three-dimensional Pareto set, we evaluate a posteriori the solutions of **S2** regarding $f_{\mathrm{S,u}}$. The obtained three-dimensional objective space for **S2** (green) and **S3** (blue) is visualized as projections in Fig. 7. Figure 7a shows the objectives that were used in both experiments. We qualitatively observe that for maintenance costs of $f_{\mathrm{M,u}} < 0.3 \times 10^5$ the solutions of **S2** dominate those of **S3** while otherwise the solutions lie on top of each other. Looking at the projections in Fig. 7b and Fig. 7c, we see a large gap between the solutions obtained by both experiments. The **S3** approach has found better solutions regarding the social costs $f_{\mathrm{S,u}}$ than the **S2** approach. We want to quantify the differences between the visualized solutions from **S2** (green) and from **S3** (blue). Therefore, we compare every network obtained in **S2** with every network calculated in **S3** regarding their respective travel and social costs by calculating their relative differences (**S2** is the baseline). The resulting boxplot is shown in Fig. 8a. We see that on average (dashed lines) the approach **S3**, which includes the social objective, produces solutions that are 13 % worse than those of **S2** regarding travel costs, and 27 % better regarding social costs. In Fig. 8a, the left (green) box range indicates that three-quarters of all networks from **S3** are only at most 16 % more expensive, one-fourth of the networks are even less expensive than the networks from **S2**. The right (blue) box of Fig. 8a shows that 75 % of the networks calculated in **S3** are at least 20 % more social, and a quarter of the networks is even at least 42 % more social than the networks from **S2**.

Discussion: What are the benefits and costs of integrating social criteria into network optimization? First, we get solutions that are socially more acceptable. On average, our networks achieve a 27 % reduction in noise. However, we also note a 13 % increase in monetary costs, assuming that time and money are proportional. By integrating a further objective function, differently shaped transport networks are created which, on average, establish longer flight corridors between two vertiports than the shortest possible connection. On the one hand, this causes longer travel times for the network users. On the other hand, these longer corridors are then spanned over areas of the city where they are perceived as less annoying by the residents living under the aerial traffic network. The choice of a solution from the Pareto set, also called *Decision Making*, is crucial as it



(a) Travel cost plotted over maintenance cost.

(b) Social cost plotted over maintenance cost.

(c) Social cost plotted over travel cost.

FIGURE 7. The projections of the three Pareto sets obtained by the two-objective optimization on the straight-path network (S2), the three-objective optimization on the straight-path network (S3), and the three-objective optimization on the straight and curved-path network (SC3). Each Pareto set's knee point is highlighted in red.

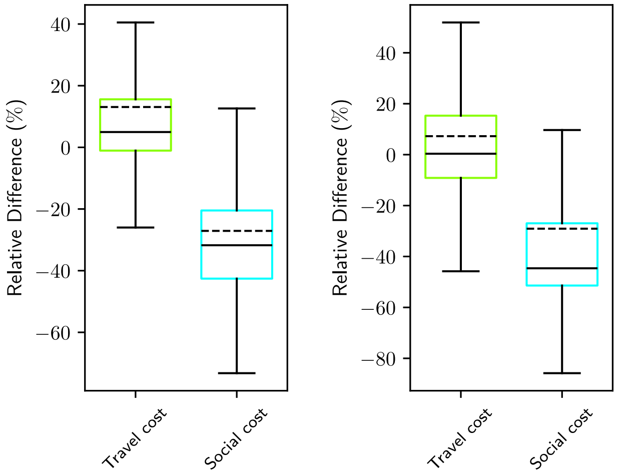
significantly impacts both social attractiveness and monetary costs.

A decision can be made based on Pareto set metrics or preference-based algorithms that go beyond the scope of this work. The interested reader is referred to the overview section by Schmitt [43]. The so-called knee point, generally realizing a good trade-off between all objectives, is often selected as the preferred solution from the Pareto set (red-colored markers in Fig. 7). There are several ways to calculate the knee point [43]. We adopt the method by Sun et al. [44] and determine the knee point as the solution with the smallest distance to a Utopia point, which is in our case the origin of the objective space. When we compare the **S3** knee point with the **S2** knee point (baseline), the travel costs get 6.6 % worse, while the social attractiveness increases by 26.3 %.

2) ECONOMIC VS. SOCIO-ECONOMIC PRE-OPTIMIZATION

Secondly, we want to investigate the influence on the network optimization if we deviate from the restriction of straight connections, as is often done in the literature. Initializing with straight connections is an intuitive way to find good economic solutions. However, this strategy tends to neglect the social aspect. What happens when we calculate social paths beforehand and make them available during *Path-merging*? We will compare the resulting networks of experiment **S3** (three-objective optimization on the set of straight paths) with those of **SC3** (three-objective optimization on the set of straight and curved paths) to find an answer. The Pareto set projections of both experiments are visualized in Fig. 7 in different colors for **S3** (blue) and **SC3** (orange). Qualitatively, Fig. 7b and Fig. 7c already show the large difference in the social costs of the networks obtained by the two experiments. The quantitative comparison between the networks generated by both approaches results in the boxplot in Fig. 8b. The experiment **S3** is the baseline for calculating the relative differences regarding travel cost and social cost between all traffic networks of **S3** and all traffic networks of **SC3**. The results indicate that on average (dashed lines) the solutions found on the combined straight-and-curved-path network are 7 % worse regarding travel cost and 29 % better regarding social cost than the solutions found within the only straight-path network. On the one hand, the range of the left (green) box in Fig. 8b visualizes that 75 % of all networks from **SC3** are only at most 16 % more expensive than the networks from **S3**. On the other hand, the right (blue) box of Fig. 8b shows that three-quarters of the networks calculated in **SC3** are at least 27 % more social, while 25 % are even at least 51 % more social than the networks obtained in **S3**.

Discussion: Extra effort is necessary to optimize a traffic network following approach **SC3**. This includes the a priori optimization of paths between all vertiports regarding social costs. In addition, the network optimization itself requires more time to converge (Table 3). In terms of social acceptability, the extra effort is worth it as the increase in social acceptance is greater than the increase in monetary costs. The difference becomes evident when comparing the knee point solutions from both approaches. The knee point solution of approach **SC3** requires 1.7 % more travel time but is 47.7 % better regarding the social criterion than the knee point solution of **S3** (baseline).



(a) Results for the comparison between the traffic networks from the optimization run **S2** (two objectives) without the social objective and the traffic networks from the run **S3** (three objectives) including the social objective.

(b) Results for the comparison between the traffic networks from the optimization run **S3** without pre-optimized paths (straight-path network) and the traffic networks from the run **SC3** with pre-optimized paths (straight and curved-path network) paths.

FIGURE 8. Results for the two inter-experiment comparisons S2 vs. S3, and S3 vs. SC3. The traffic networks from the respective first experiment are compared to those from the respective second experiment in terms of travel costs and social costs. The relative differences are visualized using boxplots, with the respective first experiment serving as the baseline. The box spans from the data's first quartile to the third quartile. The black straight line within the box represents the median, and the dashed line is the mean of the data. Since we are comparing the results of a minimization problem, 'the lower the better' applies here.

To provide a comprehensive analysis, we have also included the comparison results between **S2** and **SC3**. On average, the solutions of a three-criteria optimization with an initial set of both straight and curved paths are 17 % more expensive but 51 % more socially beneficial than the networks obtained with a two-criteria optimization on a set of only straight paths.

3) VISUALIZATION AND FURTHER EXAMINATION OF SC3

Finally, we show the visualization of some selected solutions obtained from approach **SC3** in Fig. 9. For reasons of clarity, we limit ourselves to solutions $\{v \,|\, f_M(v) < 0.05 \cdot f_M(v_5)\}$ whose maintenance costs are less than 5 % of the initial solution v_5, otherwise the high number of edges would make it difficult to recognize anything in the networks plotted on top of each other. From the remaining solutions, we visualize the solution with the least maintenance costs in violet, the solution with the best travel costs in black, the solution with

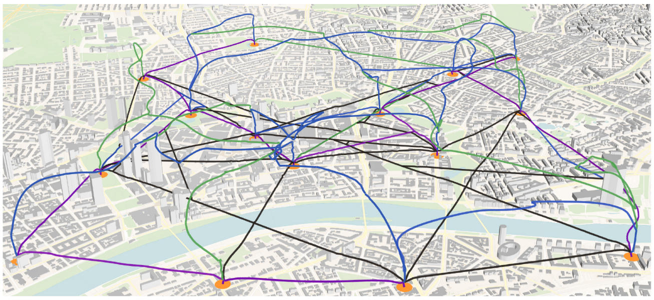


FIGURE 9. Visualization [46] of different aerial traffic networks obtained by selecting differently preferenced solutions from the Pareto set of solutions obtained by SC3. The network with the least maintenance costs is displayed in violet, the one with the smallest travel costs in black, the network with the largest social acceptability is shown in green, and the knee point (trade-off between all three objectives) network is visualized in blue.

TABLE 4. Deviation from the best obtained objective function value and relation between additional economic effort and resulting social benefit.

Solution	$f_{\mathrm{M,u}}$	$f_{\mathrm{T,u}}$	$f_{\mathrm{S,u}}$
Least maintenance cost network	0 %	56 %	672 %
	+5 %		-8 %
	+10 %		-16 %
	+15 %		-24 %
	+20 %		-32 %
Least travel cost network	180 %	0 %	303 %
		+5 %	-22 %
		+10 %	-44 %
		+15 %	-66 %
		+20 %	-74 %
Best social cost network	155 %	94 %	0 %
Knee point solution	123 %	53 %	4 %

the best social costs in green, and the knee point as a trade-off solution in blue. In addition, a video file [45] visualizing the respective networks (same color coding) is provided in the digital supplementary material of this article.

Table 4 illustrates how the (near-)optimal solutions for one objective may not perform optimally regarding the other objectives. It is shown how the traffic networks specified in each row deviate from the best-obtained solution determined by the column. Moreover, the observed correlation between economic investment and social benefit is displayed. As an example, we look at the least maintenance cost network, visualized in violet in Fig. 9, which equals the Steiner tree $G_{\mathrm{ST,L}}$. It connects all 16 vertiports by a set of edges of minimal maintenance costs (i.e., edge length). Consequently, between any two vertiports, only one route exists that is generally not the shortest or most social one available in the initial graph G_0. Looking at Table 4, we can thus observe that its objective function value regarding travel time is 56 % worse than this of the least travel cost network and its social cost value is 672 % worse than that of the best social cost network. Based on this solution, increasing the allowed maintenance costs by 5 % results in an 8 % improvement in social acceptance. Thus, if a decision maker decides to accept additional maintenance costs of 15 %, the resulting network is 24 % more social than the least maintenance cost network. The best travel cost network, depicted in black in Fig. 9, consists mainly of straight edges that connect the vertiports directly, allowing for the quickest possible travel between the hubs. However, as these fast lanes are located near the ground, this comes at the expense of social acceptance, which is 303 % worse than the best social cost network. If a decision maker were to cause network users to have 10 % higher travel costs, the resulting network would be 44 % quieter than the network with the lowest travel costs. The best social cost network, visualized in green in Fig. 9, mainly includes high-altitude transportation corridors that avoid areas of high social aversion. Due to the longer corridors and the detours taken, both the maintenance and travel costs increase by 155 % and 94 % respectively. The knee point network, shown in blue in Fig. 9, is a possible solution that realizes a balance between the competing objectives. The twelfth row in Table 4 displays the relative differences between the knee point solution and the respective best solutions. The knee point traffic network combines straight direct connections and curved corridors at higher altitudes, thus achieving a compromise between established objective functions.

V. CONCLUSION

This research aimed to examine the feasibility and the effects of integrating a social perspective into the design process of aerial traffic networks in the domain of Urban Air Mobility (UAM) applications. We developed a multi-objective network optimization framework to find Pareto-optimal three-dimensional urban air traffic networks given a set of potentially pre-optimized paths. Contrary to other approaches, we also looked at this problem from a social point of view. We first analyzed the impact of a social criterion in the actual network optimization. We then investigated the effects of providing more social paths in the initial set of paths before the optimization step. The evaluation results suggested that including social objectives in UAM infrastructure optimization is beneficial. Even a small increased economic effort can lead to much greater social acceptance.

This work has provided the algorithmic framework for further interesting studies. On the one hand, we would like to investigate the dependence of the obtained results on the city (i.e., other maps). On the other hand, simulating more realistic objective functions that involve a tighter coupling to the city structure appears desirable.

In the medium term, we are also interested in breaking up static networks in favor of dynamic structures that change with the time of day or weather conditions: In this work, the initial sets of straight and curved paths used for network assembly and optimization were obtained from independent many-objective path optimizations. The path optimizations resulted in Pareto sets of paths. Extreme points (i.e., unit vectors as selection weights) were then selected from these path Pareto sets to form the initial path sets of straight and curved paths. Calling for a dynamic approach, we want to investigate how to quickly adapt an optimized traffic network when the selection weights are updated and an evolvable approach is required.

REFERENCES

[1] P. Fontaine (Federal Aviat. Admin., Washington, DC, USA). *Urban Air Mobility (UAM) Concept of Operations*. 2023. Accessed: Feb. 27, 2024. [Online] Available: https://www.faa.gov/sites/faa.gov/files/Urb%20A%20Mobili%%28U%%20Conce%20%20Operatio%202.0_1.pdf

[2] D.-S. Jang, C. A. Ippolito, S. Sankararaman, and V. Stepanyan, "Concepts of airspace structures and system analysis for UAS traffic flows for urban areas,"'in *Proc AIAA*, 2017, PP. 1–15.

[3] (Publi. Off. Eur. Union, Luxembourg City, Luxembourg). *U-Space Concept of Operations (ConOps)—Fourth Edition: Single European Sky ATM Research 3 Joint Undertaking*. 2023. Accessed: Feb. 28, 2024. [Online]. Available: https://op.europa.eu/s/zg7g

[4] D. Geister and B. Korn (German Aerosp. Center, Cologne, Germany). *DLR Blueprint—Concept for Urban Airspace Integration*. 2017. Accessed: Feb. 27, 2024. [Online] Available: https://www.dlr.de/de/medien/publikationen/sonstige-publikationen/2017/blueprint-concept-for-urban-airspace-integration_2933/@@download/file

[5] K. Balakrishnan, J. Polastre, J. Mooberry, R. Golding, and P. Sachs (A^3 Airbus, San Francisco, CA, USA). *Blueprint for the Sky: The Roadmap for the Safe Integration of Autonomous Aircraft*. 2018. Accessed: Feb. 28, 2024. [Online]. Available: https://www.airbus.com/sites/g/files/jlcbta136/files/2022-07/Airbus-whitepaper-integration-autonomous-vehicles.pdf

[6] (EmbraerX, Fort Lauderdale, FL, USA). *Flight Plan 2030: An Air Traffic Management Concept for Urban Air Mobility*. 2019. Accessed: Feb. 28, 2024. [Online]. Available: https://daflwcl3bnxyt.cloudfront.net/m/4e5924f5de45fd3a/original/embraerx-whitepaper-flightplan2030.pdf

[7] J. Holden and N. Goel (Uber Technol. Inc., San Francisco, CA, USA). *Fast-Forwarding to a Future of on-Demand Urban Air Transportation*. 2016. Accessed: Feb. 28, 2024. [Online]. Available: https://d1nyezh1ys8wfo.cloudfront.net/static/PDFs/Eleva%2BWhitepaper.pdf?uclick_id=688eb213-8550-46cd-b1a9-4ba0fe26fb6d

[8] A. Bauranov and J. Rakas, "Designing airspace for urban air mobility: A review of concepts and approaches," *Progr. Aerosp. Sci.*, vol. 125, Aug. 2021, Art. no. 100726.

[9] E. Sunil et al., "Metropolis: Relating airspace structure and capacity for extreme traffic densities," in *Proc. ATM R D Semin.*, 2015, pp. 1–11.

[10] C. L. Denham, W. G. Cummings, and J. C. Smith, "Theoretical and simulated capacity of urban air mobility airspace characteristics," in *Proc. SciTech Forum*, 2023, p. 546.

[11] W. G. Cummings, C. L. Denham, and J. C. Smith, "Effect of airspace characteristics on urban air mobility airspace capacity," in *Proc. SciTech Forum*, 2023, p. 545.

[12] A. Fedrigo, "A series of macroscopic models for urban air mobility traffic flow," in *Proc. SciTech Forum*, 2023, p. 785.

[13] B. Pang, W. Dai, T. Ra, and K. H. Low, "A concept of airspace configuration and operational rules for UAS in current airspace," in *Proc. DASC*, 2020, pp. 1–9.

[14] V. Bulusu, V. Polishchuk, and L. Sedov, "Noise estimation for future large-scale small UAS operations," in *Proc. Inter-Noise Noise-Con Congr. Conf.*, vol. 254, 2017, pp. 864–871.

[15] S. Brulin and M. Olhofer, "Bi-level network design for UAM vertiport allocation using activity-based transport simulations," in *Proc. EVTeC*, 2023, pp. 1–4.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 1–41, 2015.

[21] M. Gariel, A. N. Srivastava, and E. Feron, "Trajectory clustering and an application to airspace monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1511–1524, Dec. 2011.

[22] P. Tampakis, N. Pelekis, N. Andrienko, G. Andrienko, G. Fuchs, and Y. Theodoridis, "Time-aware sub-trajectory clustering in hermes@PostgreSQL," in *Proc. ICDE*, 2018, pp. 1581–1584.

[23] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: A partition-and-group framework," in *Proc. SIGMOD*, 2007, pp. 593–604.

[24] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 96, 1996, pp. 226–231.

[25] Z. Chen, H. T. Shen, and X. Zhou, "Discovering popular routes from trajectories," in *Proc. ICDE*, 2011, pp. 900–911.

[26] M. Ahmed, S. Karagiorgou, D. Pfoser, and C. Wenk, "A comparison and evaluation of map construction algorithms using vehicle tracking data," *Geoinformatica*, vol. 19, pp. 601–632, Jul. 2015.

[27] R. Ding et al., "Application of complex networks theory in urban traffic network researches," *Netw. Spat. Econ.*, vol. 19, pp. 1281–1317, Dec. 2019.

[28] M. T. Gastner and M. E. Newman, "The spatial structure of networks," *Eur. Phys. J. B, Condens. Matter*, vol. 49, pp. 247–252, Jan. 2006.

[29] G. Li, S. D. S. Reis, A. A. Moreira, S. Havlin, H. E. Stanley, and J. S. Andrade Jr., "Towards design principles for optimal transport networks," *Phys. Rev. Lett.*, vol. 104, no. 1, 2010, Art. no. 18701.

[30] A. Chen, J. Kim, S. Lee, and Y. Kim, "Stochastic multi-objective models for network design problem," *Expert Syst. Appl.*, vol. 37, no. 2, pp. 1608–1619, 2010.

[31] X. He, F. He, L. Li, L. Zhang, and G. Xiao, "A route network planning method for urban air delivery," *Transp. Res. E, Logist. Transp. Rev.*, vol. 166, Oct. 2022, Art. no. 102872.

[32] S. Mai, M. Deubel, and S. Mostaghim, "Multi-objective roadmap optimization for multiagent navigation," in *Proc. CEC*, 2022, pp. 1–8.

[33] N. Hohmann, S. Brulin, J. Adamy, and M. Olhofer, "Three-dimensional urban path planning for aerial vehicles regarding many objectives," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 639–652, 2023.

[34] F. K. Hwang and D. S. Richards, "Steiner tree problems," *Networks*, vol. 22, no. 1, pp. 55–89, 1992.

[35] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959.

[36] M. Barthélemy, "Spatial networks," *Phys. Rep.*, vol. 499, nos. 1–3, pp. 1–101, 2011.

[37] B. Rao, A. G. Gopi, and R. Maione, "The societal impact of commercial drones," *Technol. Soc.*, vol. 45, pp. 83–90, May 2016.

[38] A. J. Torija, Z. Li, and R. H. Self, "Effects of a hovering unmanned aerial vehicle on urban soundscapes perception," *Transp. Res. D, Transp. Environ.*, vol. 78, Jan. 2020, Art. no. 102195.

[39] 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.

[40] J. Blank and K. Deb, "Pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89497–89509, 2020.

[41] (OpenStreetMap, Cambridge, U.K.). *Frankfurt City Center*. Accessed: Dec. 1, 2020. [Online]. Available: https://www.openstreetmap.org/export#map=14/50.1138/8.6869

[42] J. Blank and K. Deb, "A running performance metric and termination criterion for evaluating evolutionary multi-and many-objective optimization algorithms," in *Proc. CEC*, 2020, pp. 1–8.

[43] T. Schmitt, "A posteriori decision making strategies," in *Multi-Objective Building Energy Management Optimization with Model Predictive Control*. Darmstadt, Germany: Technische Universität Darmstadt, 2022, ch. 4.2.2, pp. 81–84. [Online]. Available: http://tuprints.ulb.tu-darmstadt.de/22344/

[44] G. Sun, G. Li, S. Zhou, H. Li, S. Hou, and Q. Li, "Crashworthiness design of vehicle by using multiobjective robust optimization," *Struct. Multidiscipl. Optim.*, vol. 44, pp. 99–110, Jul. 2011.

[45] "Animation tool for 3D and satellite images from Google earth." Google Earth Studio. Accessed: Feb. 26, 2024. [Online]. Available: https://earth.google.com/studio/docs/

[46] "Geospatial analysis tool." Kepler.gl. Accessed: Feb. 26, 2024. [Online]. Available: https://kepler.gl/demo

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, where he is currently pursuing the Ph.D. degree in collaboration with the Honda Research Institute Europe GmbH, Offenbach am Main, Germany. His current research interests involve multi-objective optimization, path planning, and network optimization, particularly 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 and the Ph.D. degree from the Institute for Fluid Mechanics and Aerodynamics, Technical University of Darmstadt, Germany, in 2013, 2016, and 2020, respectively. Since 2021, he has been working as a Senior Scientist with the Honda Research Institute Europe within the Complex Systems Optimisation and Analysis Group. 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, where he became a Full Professor since 1998 and the Head of the Control Methods and Intelligent Systems Laboratory. From 1992 to 1998, he was an Engineer and the Manager with the area of Control Applications at Siemens AG, Erlangen, Germany. His research interests are in the areas of nonlinear control, intelligent systems, and mobile robots.

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