Transportation Research Part C 143 (2022) 103808

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Traffic management and resource allocation for UAV-based parcel delivery in low-altitude urban space

Ang Li [a,*], Mark Hansen [a], Bo Zou [b,a]

[a] Department of Civil and Environmental Engineering, University of California Berkeley, United States
[b] Department of Civil, Materials, and Environmental Engineering, University of Illinois Chicago, United States

ARTICLE INFO

Keywords:
Unmanned Aircraft Systems Traffic Management (UTM)
Urban parcel delivery
UAV path planning
UAV Conflict Detection and Resolution (CD&R)
Resource allocation
Mechanism design

ABSTRACT

This research proposes a framework of Unmanned Aircraft Vehicles (UAV) system traffic management in the context of parcel delivery in low-altitude urban airspace, including clustering-based UAV path planning, Unmanned Aircraft System Traffic Management (UTM) with conflict detection and resolution (CD&R), and mechanism design for airspace resource allocation. For UAV path planning, we develop a procedure by first clustering a large variety of obstacles that arise from building heights and terrain topology and can impede UAV flying. Based on the clustered obstacles, Saturated Fast-Marching Square (Saturated FM2) algorithm is then employed to generate optimal and alternative paths for each UAV mission. While identifying the optimal and alternative paths does not consider UAV traffic interactions, several traffic management models are proposed to efficiently allocate spatial and temporal airspace resources to UAV missions. The UTM models determine the departure time and the path to take for each UAV flight while resolving path conflicts from different perspectives. Specifically, four UTM models are proposed: Sequential Delay (SD) Model, Sequential Delay/Reroute (SDR) Model, Full Optimization (FO) Model, and Batch Optimization (BO) Model. Among the four models, the BO model is of particular interest as it strikes a balance between seeking a system optimum solution and maintaining computational tractability. Given that traffic management requires private information from UAV operators, the Vickrey-Clarke-Groves (VCG) mechanism is further adapted to the UTM context, in which airspace resource allocation is performed in conjunction with a payment scheme to incentivize truthful private information reporting by UAV operators. Extensive numerical analysis is conducted with San Francisco as the case study area. The results show the effectiveness of the proposed framework, particularly the scalability of the BO model. We also find that payment by a UAV flight under the adapted VCG mechanism depends critically on traffic density and the extent of interaction the UAV flight has with other flights.

1. Introduction

The explosive growth in e-commerce, increasing urgency of de-carbonization, and rapid advances in technologies and the gig economy are driving dramatic changes in urban package delivery. In New York city, for example, more than 1.5 million packages were delivered daily in 2019 (Haag and Hu, 2019). As a result of the large volume, we are seeing an increasing number of delivery trucks and vans entering and driving around cities every day, contributing to greater traffic congestion, air pollution, noise, road deterioration,

* Corresponding author.
E-mail addresses: angli@berkeley.edu (A. Li), mhansen@ce.berkeley.edu (M. Hansen), bzou@uic.edu (B. Zou).

https://doi.org/10.1016/j.trc.2022.103808
Received 8 July 2021; Received in revised form 24 May 2022; Accepted 13 July 2022
Available online 11 August 2022

and safety concerns. Meanwhile, many delivery service providers, including Amazon, UPS, and DHL, have opened urban warehouses near or in city centers from which last-mile deliveries to customers are performed (Haag and Hu, 2019; Young, L. 2020a; Young, L. 2020b). The large number of online orders from restaurants, stores, and urban warehouses – especially during the COVID-19 pandemic – along with the expectation of delivery within an order of hours (SupplyChainBrain, 2020) have exacerbated the need for new delivery solutions to meet the demand.

While various innovations have been considered (Kafle et al., 2017; Ranieri et al., 2018; Le et al., 2019; van Duin et al., 2020; González-Varona et al., 2020), delivery by Unmanned Aircraft Vehicles (UAV), or drones, is increasingly perceived as an integral part of the future solution for urban freight movement to provide fast, point-to-point deliveries. Established logistics companies and startups have already begun using drones in package delivery: FedEx Express and Wing Aviation have completed their first scheduled commercial-to-residential drone deliveries in the US (Norman, H., 2019). In 2019, UPS got the first broad FAA approval for drone delivery (Josephs, L., 2019). Amazon has also received such approval for its Prime Air drone delivery fleet (Palmer, A., 2020), whose cost of delivery per package is estimated to be only two thirds or less of traditional ground vehicle-based delivery (Sudbury and Hutchinson, 2016).

UAV delivery has several advantages over traditional parcel delivery, such as shorter delivery times, lower maintenance costs, and environmental friendliness (Lee et al, 2016; Lim and Jung, 2017). When deployed at a sufficiently large scale, UAV delivery may also help reduce road safety risks and mitigate traffic congestion on the ground. Moreover, as a result of the COVID-19 pandemic, the trend of shifting from physical stores to online shopping has been accelerated by roughly five years (Perez, S., 2020). It thereby imposes even greater pressure on logistics service providers, which are struggling in soliciting sufficient workers to meet the delivery requests. In the midst of the pandemic, delivery workers are exposed to a high risk of virus infection due to frequent contacts with store staffs and customers during order pickup and delivery. UAV delivery, which is contact free, can help mitigate the spread of the virus as well as solve the imbalance between order demand and delivery capacity, if not for the current pandemic, then perhaps for a future one.

With potentially large demand for UAV delivery in the foreseeable future, the need for and importance of efficiently managing UAV traffic in urban airspace is arising. Many countries have already started developing UAV Traffic Management (UTM) methods (Kopardekar et al, 2016; Unmanned Airspace, 2019). The subject has attracted many interests in the research community (Labib et al, 2019; Ho et al, 2019a). While there is a body of literature on how to optimally configure urban delivery systems with UAVs, most existing research does not consider the UAV delivery problem in the context of traffic management. Yet the ability to safely and efficiently resolve conflicts will become increasingly urgent as the use of UAVs for urban package delivery and other purposes intensifies. To our knowledge, there has been little research on integrating traffic management with UAV delivery. Study of strategic traffic management in the context of an UAV delivery system, in particular the systematic efficient allocation of congestible airspace resources to UAVs, is particularly lacking. Prior work on planning UAV delivery systems has focused on problems such as UAV vertiports facility location problem (Vascik and Hansman, 2019; Fadhil, 2018; Rath and Chow, 2019), maximizing the number of delivered packages by UAV while satisfying battery consumption constraints (Kim et al, 2020), and evaluation of collision risk in small UAV systems (Weinert et al, 2018). There has been limited research on Conflict Detection and Resolution (CD&R) models for high traffic densities of UAVs. While some In-Flight CD&R methods have been recently proposed and studied (Foina et al, 2015; Yang et al, 2016; Ho et al, 2019c), strategic Pre-Flight CD&R methods remain mostly unexplored (Ho et al, 2019b). None of the studies are concerned with identifying and efficiently resolving path conflicts in a UAV system that features the high traffic intensities that may be required to meaningfully relieve road congestion or reduce pandemic spread.

This paper attempts to fill this gap. We propose a systematic approach for assigning departure times and paths to UAVs with heterogeneous cost functions traveling between a pre-determined set of origins and destinations with fixed desired departure times, in a manner that resolves all UAV conflicts while maintaining safe distances from airborne obstacles. In addition, since solving this problem requires information from individual UAVs, which is likely to be private and unknown *a priori* to the system planner/central controller, it is important to ensure that individual UAV operators report private information truthfully. To accomplish the above, we develop a suite of methods which includes clustering-based UAV path planning, hybrid time/route-based traffic management modeling that embeds conflict resolution, and mechanism design to elicit truthful information from UAV operators. Given a set of specified UAV missions and operator preferences, we propose and compare several traffic management models of varying sophistication and complexity to schedule and route the UAV flights in a manner that avoids conflicts. In particular, we propose models that can yield high-quality, near-optimal solutions for UAV scheduling and routing while requiring short computation time. Furthermore, to elicit truthful preference information from the operators, we adapt the Vickrey-Clarke-Groves (VCG) mechanism to our context by which UAV operators pay for their missions when their UAV trajectories involve conflicts with other missions.

This rest of the paper is organized as follows. Section 2 reviews the related work on UTM. The methodology adopted in this paper is described in Section 3. Section 4 presents the modeling and simulation results using San Francisco as the case study context. Conclusions are summarized in Section 5.

2. Literature review

UAVs have the potential to significantly reduce the cost and time of making last-mile deliveries. To investigate this, many researchers have worked on routing UAVs in the context of Vehicle Routing Problems, where UAVs are utilized in conjunction with trucks in last-mile delivery (Murray and Chu, 2015; Murray and Raj, 2020). Multi-trip vehicle routing problems for UAV delivery that reduce delivery time and cost are proposed in Dorling et al. (2017). Ferrandez et al. (2016) investigate the effectiveness of UAV delivery by looking at the reduction in overall delivery time and energy for an integrated truck-UAV network compared with truck-only delivery system. Boysen et al. (2018) schedule UAV deliveries for truck routes considering trucks as mobile landing and take-off

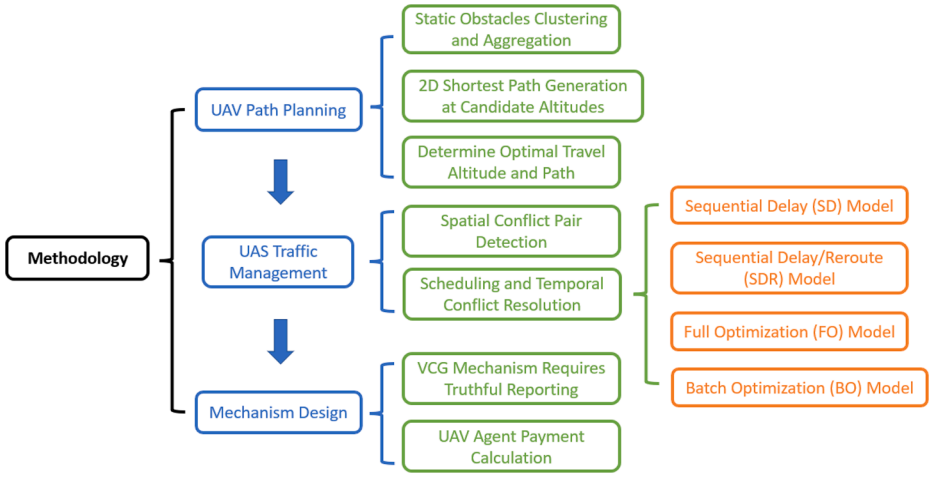

Fig. 1. Methodology overview

platforms.

Besides routing, path planning is of vital importance in UAV delivery, so that UAVs can avoid obstacles while accomplishing missions. Sathyaraj et al. (2008) compare multiple algorithms, including Dijkstra's algorithm, Bellman Ford's algorithm, Floyd-Warshall's algorithm, and A* algorithm, for UAVs path planning. The authors find that A* algorithm performs better than the others. A* with Jump Point Search (JPS) and A* with Rectangular Symmetry Reduction (RSR) are improved path planning algorithms based on A* (Botea, Mueller and Schaeffer, 2004; Harabor, Botea and Kilby, 2011; Harabor and Grastien, 2011). Duchoň et al. (2014) find that A* (JPS) has better performance than A* (RSR). Fast-Marching methods have also been found to yield consistent, accurate, and highly efficient solutions in path planning around obstacles (Sethian, J.A., 1999) and thus can be adopted for UAV path planning (Valero-Gomez et al., 2013). In addition, Rapidly Exploring Random Trees (RRTs) have been used to search for a path for a multirotor (Janjoš et al., 2017). Optimal control is another widely used trajectory-optimization approach in aviation. Ng et al. (2014) use optimal control to develop flight trajectories that approximately minimize aircraft travel time and fuel burn. Our previous work compared several path planning algorithms and concluded that Fast-Marching methods outperform several other path planning algorithms (Li and Hansen, 2020).

Several studies have been conducted on other aspects of system planning for UAV delivery. (Torabbeigi, Lim and Kim, 2020) address the strategic operation planning of UAV-based parcel delivery problems considering payload effects on battery consumption rate. A distance-constrained mobile hierarchical facility location problem is adapted in (Shavarani et al, 2018) to determine the optimal number and locations of launch and recharge stations for UAVs, with the objective of minimizing system costs.

With potentially large UAV delivery demand, managing UAV traffic in urban airspace becomes necessary. NASA (Kopardekar et al, 2016) and the Netherlands Aerospace Centre (Unmanned Airspace, 2019) have been developing UTM systems, as have agencies in other parts of Europe, Singapore, and Korea. Kim et al. (2020) employs a Mixed-Integer Linear Programming (MILP) model for generating UAV delivery task sequence to maximize number of delivered packages using building rooftops in cities. Weinert et al (2018) analytically estimate the collision risk of small unmanned aircrafts at low altitudes in nonterminal airspace. Weibel et al. (2012) derived preliminary separation standards for UAVs analytically in terms of both time and distance. A simulation framework is presented in (Peinecke and Kuenz, 2017) that can load a given urban airspace with a specified demand or frequency of delivery UAVs and resolve conflicts without introducing too much delay for delivery. Though many UTM concepts are presented, UAV safety separation and conflict resolution standards are studied, and strategic planning models for UTM are proposed, no research has investigated systematic and efficient allocation of low-altitude urban airspace to UAVs. Determining which conflict flight is going to take action to avoid conflict, whether a UAV flight is to be delayed or rerouted to avoid conflict, how to efficiently assign delays, and which path to take if an UAV flight is rerouted to avoid conflict, are examples of allocation problems that have not been studied.

An important aspect of resource allocation is the design of economic mechanism that involves agents who act in their own interests and possibly against the interest of the designer. The VCG mechanism is a generic truthful mechanism for achieving a socially-optimal solution. It incentivizes every agent to truthfully report their preference information. Existing research uses mechanism design for flight slot assignment and manned Air Traffic Management (ATM). Ball et al (2020) define and investigate quantity-contingent auctions as a mechanism for allocating airport arrival and departure slots. Hoffman et al (2012) propose a mechanism design for assigning flight reroutes and ground delays which is flexible enough to admit real-time operational changes made by the FAA and changing preferences of NAS users. Zou et al (2015) propose a mechanism design for public parking slot assignment in an environment empowered by ubiquitous communications between infrastructure and vehicles. Mechanism design for efficient airspace resource allocation in the context of UAV delivery has not yet been explored. We also intend to fill this gap in the present paper.

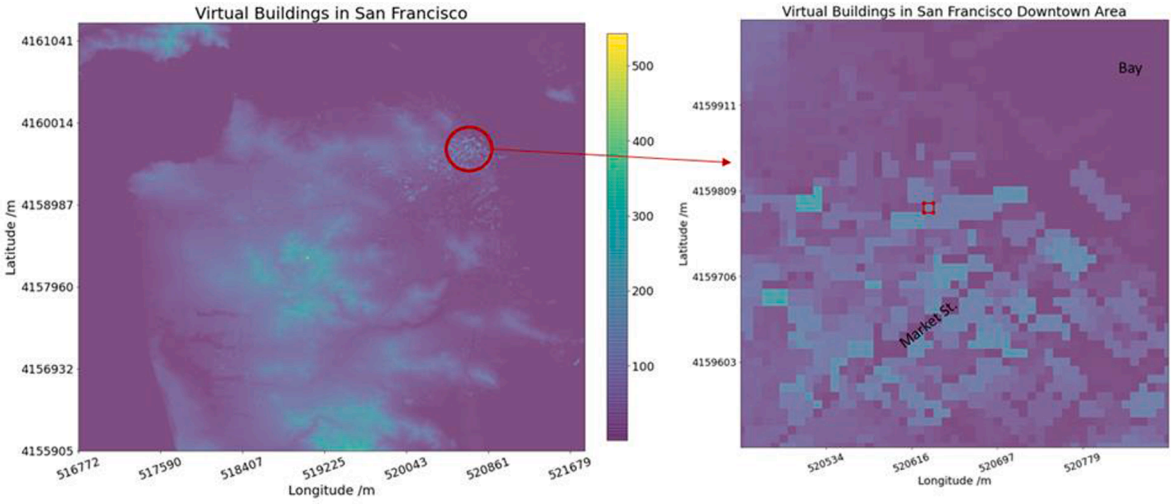

Fig. 2. San Francisco virtual buildings in one arc-sec grid system representing all static obstacles including both topological and above-ground obstacles

3. Methodology

We consider a system for strategic UTM whereby, prior to departure, flight paths and, if necessary, departure delays are assigned so that conflicts (i.e. loss of separation) between UAVs are eliminated. In practice, this strategic management layer would be complemented by a tactical layer, presumably based on automated self-separation, which would be used to resolve conflicts introduced by stochastic factors such as winds, control errors, and UAV performance variability that are not explicitly considered in the strategic layer. The strategic layer can be more or less conservative, and thus place lesser or greater burden on the tactical layer, by varying the criteria used to determine conflicts. While the interaction between these two layers is an important research area, the focus of this research is solely on the strategic layer.

Our methodology includes four components required to simulate and evaluate the proposed strategic UTM including UAV path planning, conflict detection, strategic conflict resolution, and mechanism design. Fig. 1 shows the overview of the methodology. Four different UTM models are proposed for UAV flight scheduling and conflict resolution. Some of the models require true flight operator preferences information, leading to the need for the mechanism design component.

3.1. Deterministic clustering-based single UAV path planning

We assume a set of UAV flights, each with a given origin, destination and desired departure time. For each flight we must determine a set of desirable 3D routes. The path planning approach is simplified by assuming a set of discrete altitudes, and finding the shortest 2D path at each altitude. The optimal travel altitude is determined by minimizing a linear cost function associated with both horizontal and vertical flying. In order to generate the most representative altitude candidates based on the topography of terrain and the elevation of building obstacles, we employ a clustering approach to characterize the height and proximity of the numerous static obstacles in a dense urban area. Based on the generated altitude candidates set, horizontal shortest paths that avoid obstacles are then be generated for each altitude candidate. We compare the vertical and horizontal costs to decide the optimal travel altitude and 2D cruise path at that optimal altitude. We also identify an optimal 2D path at a different altitude, which may be used in order to resolve conflicts.

3.1.1. Static obstacles clustering for altitude candidates

A set of altitude candidates is generated by clustering static obstacles. Static obstacles in urban airspace include topological obstacles (e.g. hills) and above-ground obstacles (e.g. buildings). A 10-meter keep-out geofence distance is added to above-ground obstacles to reduce the risk of collisions with buildings. UAVs also maintain 10-meter keep-in geofence distance. We combine the gridded elevation maps of topological and above-ground obstacles, which are represented by elevation at many grid points on a one arc-sec grid system, as the elevation map of all static obstacles. Fig. 2 shows the elevation map of all static obstacles in San Francisco. Then, we generate virtual buildings to represent all obstacles in the study area, including topological and above-ground obstacles. Each grid in the elevation map is regarded as a virtual building (e.g. the red square in Fig. 2 represents one such virtual building). The elevation of each virtual building is the maximum elevation of the four grid vertices.

The K-means clustering algorithm is applied to perform clustering over all virtual buildings. Through clustering, a very large number of elemental static obstacles will be represented as a much smaller number of clusters. Obstacles in each cluster are similar in height and location. Since the K-means clustering algorithm requires the same feature dimensions for every obstacle observation, each

Table 1
Algorithm for combining altitudes

Input	Elevations of all clusters *Hcluster*
Output	Altitude candidates set *Hcandidate*
1	Sort *Hcluster* in descending order
2	$Alt = Hcluster[0] =$ the first element in *Hcluster*
3	Add *Alt* to *Hcandidate*
4	**While** $Alt - 30 \geq Min(Hcluster)$:
5	$Alt =$ the first element in the sorted subset of *Hcluster* that is lower than $Alt - 30$
6	Add *Alt* to *Hcandidate*
7	**End while**

virtual building is characterized by nine features: X and Y coordinates of four grid vertices plus virtual building elevation. While all variables are measured in meters, elevation is unique because it determines obstacle height. For this reason, the elevations of virtual buildings are rescaled with appropriate rescaling factors.

After clustering, the elevation of a cluster is the maximum elevation of virtual buildings in the cluster. With 30-meter vertical separation distance (Ren et al, 2017) between small UAVs, we combine the altitudes of clusters that are within 30-meter vertical distance. Table 1 shows the algorithm to combine altitudes. We use the combined altitudes as our altitude candidates set for horizontal path planning.

3.1.2. Optimal horizontal travel route

Shortest horizontal travel paths are generated at altitude candidates obtained from subsection 3.1.1 for each UAV mission. The Saturated Fast-Marching Square (Saturated FM2) algorithm (Valero-Gomez et al., 2013) is used to generate the shortest cruise path that avoids static obstacles. Compared to the traditional Dijkstra algorithm or A* algorithm, Fast Marching methods (FM) replace the graph update by a local resolution of gradient descent, instead of only considering standard eight directions of neighbors, which significantly reduces the grid bias. After comparing several shortest path algorithms, Li and Hansen (2020) found that FM is the most efficient method in terms of both computation time and solution quality for UAV horizontal path planning. Compared with traditional Fast Marching Algorithm, Saturated FM2 imposes a speed penalty that increases continuously with proximity to obstacles within a specified threshold distance. We use the distance map to all obstacles as a speed map and assure the maximum allowable speed once a UAV is at least 10 m away from obstacles. In most cases, a UAV will maintain 10-meter keep-in geofence distance from obstacles to avoid a speed penalty. However, a UAV may trespass this keep-in geofence and stay within 10 m to an obstacle (whose boundary includes a 10-meter keep-out geofence) if the resulting travel time penalty is less than the extra travel time of another possible path.

3.1.3. Determining optimal travel altitude and path

To decide the optimal travel altitude, we must balance the horizontal path length savings from cruising at a higher altitude (with fewer obstacles) against the additional cost of ascent and descent. Several existing studies have used horizontal and vertical costs to measure path costs in UAV path planning. For example, Scherer et al. (2009) use horizontal and vertical cost ratio, and Bagherian and Alos (2015) use horizontal and vertical path length costs for UAV trajectory planning.

In this study, we assume that all UAV flight demands are generated from buildings, and origins and destinations are on building rooftops. We assign the elevation of origin and destination rooftop points as the origin and destination altitudes of the corresponding flight. For each flight, the shortest horizontal path is generated at each candidate altitude from subsection 3.1.1. Total travel cost including ascent, horizontal travel, and descent costs is calculated at all candidate altitudes. Flying costs are assumed to be proportional to the distance of flying, with the unit-distance cost ratio among ascent, descent, and horizontal flying to be 2.04: 1.53: 1 following Liu et al. (2017). The best flying path corresponds to the flying altitude with the lowest total cost. An additional flying path is further identified which corresponds to the horizontal flying altitude with the second-lowest total cost. The latter path is considered as an alternative to resolve potential conflicts between UAV paths.

3.2. Traffic management models

In this section, we propose four traffic management models of varying sophistication, each of which aims to efficiently schedule and route each UAV flight while resolving UAV path conflicts. We start by detecting flight pairs whose candidate paths create spatial conflicts. A spatial conflict is also as a temporal conflict if, based on the desired departure times of the two conflict flights, there would be simultaneous occupancy of the conflict region. The temporal conflicts must be resolved by means of either departure delay or an alternative path. Traffic management models will decide how to assign delay or to reroute flight to resolve each temporal conflict.

The four proposed models, in order of simplicity, are the sequential delay (SD) model, the sequential delay/reroute (SDR) model, and two optimization models. The SD model assigns flights a priority order, and resolves a conflict between two flights by delaying the flight with lower priority. A desirable feature of this model is that it requires no inputs other than the desired routes and the flight sequence, which might be established, for example, by the order in which UAV flight requests are received by the traffic manager. The SD model also serves as a benchmark for the other models. The SDR model is also based on flight priority, but may resolve a conflict either by delaying a flight or assigning the flight a different path (or possibly both). This model requires UAV operators to submit cost information so that the model can choose the conflict resolution strategy with the least cost.

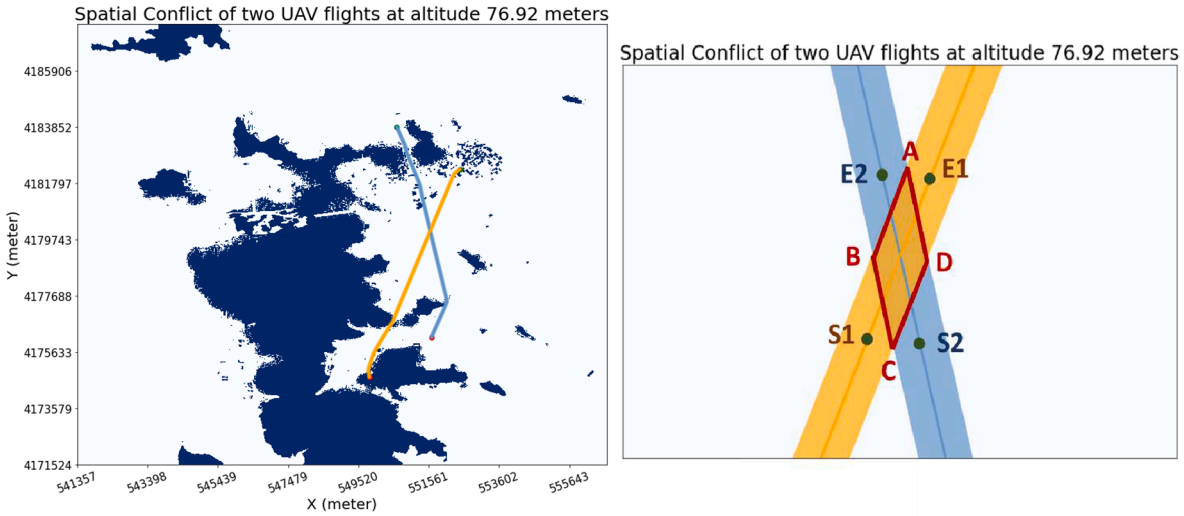

Fig. 3. An illustrative example of a pair of spatially conflict flight paths

Table 2
Algorithm for detecting spatial conflict pairs.

Input	Optimal path and alternative paths of N UAV flights in the system
Output	Spatial conflict flight pairs *Spatial_conflict_pairs*, with travel time from origin to the entry and exit points of conflict region for each spatial conflict flight
1	Add 10-meter buffer to each UAV flight path
2	*Spatial_conflict_pairs* $= []$
3	**For** any two paths of any two different flights:
4	**If** two paths are at the same altitude and overlapped:
5	Define the intersected region as conflict region
6	Find entry and exit points of conflict region for two paths $(S1, E1, S2, E2)$
7	Calculate travel time from origin to entry and exit points of the conflict region
8	Add IDs of the two flights, corresponding path IDs, and travel time to the entry and exit points of conflict region to *Spatial_conflict_pairs*
9	**End if**
10	**End for**

In contrast to the sequential models, the optimization models assign delays and paths in order to minimize the total cost of eliminating all conflicts. For these models, the traffic manager requires additional cost information for each UAV mission. Further, to assure that operators provide truthful information, this model must be paired with a mechanism design that is incentive compatible. One version - the full optimization model - considers all flights at once, which yields the true optimal solution but can be computationally intensive. This motivates a variant of the model which assigns flights in batches that are constructed so that most conflicts are between flights in the same batch. We term the full and batch optimization models as FO and BO respectively. Flight conflicts between batches are resolved sequentially based on batch priority. We will show later that BO will yield still very high-quality solution, while running much faster than the FO.

A critical underlying assumption of SDR, FO and BO is that there exists a central manager who has certain cost information for all UAV flights. FO and BO require more information than SDR. For example, the unit delay cost of an UAV flight, which may depend on the urgency of the package to be delivered. With the delay cost information, traffic management models can assign departure delay or alternative route to UAV flights while minimizing system total cost. SDR only needs the ratio of unit delay cost to unit path cost for each UAV flight, while FO and BO need more information to compare costs among different UAV flights. In addition, though UAVs could hover during flight, assigning departure delay can reduce the number of UAVs hovering in the airspace (Neufville and Odoni, 2003), reducing power consumption and risk. Self-separation to avoid potential conflicts/collisions is not considered at this stage, but can be applied in the tactical traffic management layer, as discussed above.

3.2.1. Spatial conflict detection

Spatial conflict flight pairs are detected as input for traffic management models. Any two flights with paths that spatially conflict form a spatial conflict flight pair. Both the optimal and the alternative paths of the flights are considered when identifying spatial conflict pairs. If more than two flights are in conflict, the conflicts will be represented by multiple pair-wise conflicts.

Fig. 3 illustrates how a spatial conflict is identified. In the figure, the two flights (represented by the yellow and blue paths) are northbound. A 10-meter buffer is added to each UAV path. If flight 1 taking the orange path and flight 2 taking the blue path are at the same altitude, then their path overlap indicates a spatial conflict. The overlap area ABCD is defined as the conflict region. S_i and E_i are the entry and exit points of flight i, $i = 1,2$. Point C is defined as the boundary point of the conflict region closest to the origin of flight 1.

Table 3
Algorithm of the SD model

INPUT	Flight priority order *order*; Scheduled departure time $S[i]$ for flight $i \in F$;
OUTPUT	Assigned departure time $t[i]$ for $i \in F$
1	Define an empty set of assigned flights: *assigned* $= [\,]$
2	Add the first flight ID in *order*to *assigned* with scheduled departure time
3	**For** flight f in *order* starting from the second flight:
4	Check with each assigned flight in *assigned* and get subset *s_con*that are spatially conflict with flight f.
5	Define an empty set *TemConflict* to store temporal conflicts between any assigned flight and flight f.
6	Define an empty set *TriggerConflict* to store newly triggered temporal conflicts from delay assignment
7	*TemConflict* $=$ Select flights in *assigned* that are in temporal conflict with f.
8	Calculate the delay of f to avoid conflict with any flight i, *delay*$[i]$, $\forall$ *flight i in TemConflict*.
9	*Delaynow* $= Max_{i \in TemConflict}(delay[i])$
10	*TriggerConflict* $=$ Select flights in *s_con* that enters conflict region after f exits.
11	No temporal conflict occurs at this point between f and flights in *TriggerConflict*, but these spatial conflicts may become temporal conflicts after f is delayed. Calculate the delay upper bound of f that will not trigger any of these spatial conflicts to become temporal conflict *delayub*.
12	**If** *TriggerConflict* is empty or *Delaynow* $<$ *delayub*:
13	Assign delay *Delaynow* to flight f
14	End current iteration and go to next flight at step 3
15	**Else:**
16	Stay in current flight iteration, replace the departure time of flight$f$: $s[i] = s[i] +$ *Delaynow*, then go to step 4
17	**End if**
18	**End for**

Table 4
Algorithm of the SDR model

Input	Flight priority order*order*; Scheduled departure time $S[i]$ for flight $i \in F$; Alternative paths set P of each UAV flight
Output	Assigned departure time $t[i]$ for $i \in F$
	Assigned path of each UAV flight $p[i]$for $i \in F$. $p[i] = 0$ if taking optimal path, 1 if taking second alternative path, 2 for third…
1	Define an empty set of assigned flights: *assigned* $= [\,]$
2	Add the first flight ID in *order*to *assigned* with scheduled departure time and optimal path
3	**For** flight f in *order*:
4	Check with each assigned flight in *assigned* and get subset *s_con*that spatially conflict with flight f considering all alternative paths.
5	Define an empty set *TemConflict* to store temporal conflicts between any assigned flight and flight f
6	Define an empty set *TriggerConflict* to store newly triggered temporal conflicts from delay assignment
7	**For** each assigned flight f` in *s_con*:
8	**If** f and f` are in the conflict region at the same time:
9	f will be delayed for time d and enter conflict region 10 seconds after f` exits
10	Add f and f` taking corresponding paths to *TemConflict*
11	**Else if** f exits conflict region 10 seconds before f` enters:
12	No temporal conflict occurs at this point, but this spatial conflict may become temporal conflict if flight is delayed because of other spatial conflicts. Calculate the delay upper bound that will not trigger this spatial conflict to become temporal conflict *delayub*$[f,p]$.
13	Add f taking corresponding path and *delayub*$[f,p]$ to *TriggerConflict*
14	**Else if** f` exits conflict region 10 seconds before f enters:
15	No temporal conflict.
16	**End if**
17	**End for**
18	Define an empty set *Pathcost*
19	**For** any path $j \in P$ of flight f:
20	*PathConflict* $=$ select subset from *TemConflict* where f takes path$j$
21	Delay of taking path j with all conflicts resolved$=$ maximum delay in *PathConflict*
22	Add the sum of delay cost and path cost taking path j to *Pathcost*
23	**End for**
24	Assign the path with smallest total cost to f, $p[f] = np.argmin(Pathcost)$
25	**If** the delay of taking path $p[f] <$*Min* $(delayub[f,p])$ $\forall (f,p) \in TriggerConflict$:
26	End current iteration and go to next flight at step 3
27	**Else:**
28	Stay in current iteration, replace the departure time of flight$f$: $s[i] = s[i] +$ delay of taking path $p[f]$, then go to step 4
29	**End if**
30	**End for**

Point A is defined as the boundary point of the conflict region closest to the destination of flight 1. As such, S1 is the orthogonal projection point of point C to the path of flight 1. E1 is the orthogonal projection point of point A to the path of flight 1. S2 and E2 are similarly identified.

Note that two spatially conflict flights may temporally conflict as well depending on when will they pass through the conflict region. One way to resolve a conflict is to delay a flight at its departure end. The amount of departure delay for a flight is determined such that no more than one flight will be in the conflict region at a given time. More specifically, one of the spatial conflict flights must exit the conflict region (i.e. passing the exit point) before the other one enters the conflict region (i.e. passing the start point). 10s are

Table 5
Definitions of sets, decision variables, and parameters in FO model

Sets	
F	Set of flights
P	Set of alternative flight paths
C	Set of predetermined spatial conflict flight pairs
Decision variables	
I_{ij}	Binary variable, equals to 1 if flight $i \in F$ takes path $j \in P$
D_i	Delay cost of flight $i \in F$
t_i	Assigned departure time of flight $i \in F$
z_c	Binary variable where $z_c = I_{ij} \bullet I_{mn}$, for $c = (i,j,m,n) \in C, \forall\ i,\ m \in F;\ j,\ n \in P$. $z_c = 1$ if flights i and m take paths j and n ($I_{ij} = 1, I_{mn} = 1$), which are in conflict
X_c	Binary variable where $c = (i,j,m,n) \in C. X_c = 1$ if four conditions are met: 1) flight i takes path j; 2) flight m takes path n; 3) paths j and n are in spatial conflict; and 4) flight i exits the conflict region before flight m enters the conflict region
x_c	Binary variable. Same definition as X_c except that flight m exits the conflict region before flight i enters the conflict region
Parameters	
c_{ij}	Path cost of flight $i \in F$ taking path $j \in P$
r_i	Delay cost coefficient of flight $i \in F$ when delay is less than a threshold d
R_i	Delay cost coefficient of flight $i \in F$ when delay is greater than a threshold d
b_c	Travel time from path origin to the start point of conflict region for flight i in $c = (i,j,m,n) \in C$
e_c	Travel time from path origin to the exit point of conflict region for flight i in $c = (i,j,m,n) \in C$
B_c	Travel time from path origin to the start point of conflict region for flight m in $c = (i,j,m,n) \in C$
E_c	Travel time from path origin to the exit point of conflict region for flight m in $c = (i,j,m,n) \in C$
s_i	Scheduled departure time of flight $i \in F$

further added as a buffer, so that an entry of one conflict flight cannot occur until 10s after the other conflict has exited. The time that each flight needs to travel from its origin to the entry point of a conflict region (e.g. for flight 1 in Fig. 3, the time from the origin to S1) and the time needed to pass through the conflict region (S1 to E1) are computed for each detected spatial conflict flight pair. The traffic management models assign delays and alternative paths in order to avoid temporal conflicts. Table 2 describes the algorithm of detecting all spatial conflict pairs in the system.

3.2.2. Sequential delay (SD) model

In the SD model, each flight is assumed to take its optimal path and is randomly assigned a priority order. The order is assigned randomly although other ordering schemes, such as based on scheduled departure time, could also be used. The SD model assigns a departure time to each UAV flight that avoids conflicts with higher priority flights iteratively. Table 3 provides a detailed description of traffic assignment under the SD model.

3.2.3. Sequential Delay/Reroute (SDR) model

Similar to SD model, SDR model is also based on flight priority order, but may resolve a conflict either by delaying a flight or assigning the flight a different path (or possibly both). As discussed in subsection 3.1, each flight is given two alternative paths: the shortest path at the optimal altitude and the shortest path at the second-best altitude. A conflict may be resolved by assigning a path at the second-best altitude, or by introducing some departure delay. The SDR model iteratively returns the assigned departure time and path to each UAV flight, one flight at a time per the priority order. At each iteration, we check the temporal conflicts of the current flight with each of the assigned flights, which have higher priority order. The departure time and path with the least cost will be assigned to the current flight. See Table 4 for the detailed description of the algorithm for performing SDR traffic management model.

3.2.4. Full optimization (FO) model

The FO model efficiently assigns path and departure time to each UAV mission to minimize total cost. The UAV path planning method from subsection 3.1 is also applied in FO to generate two alternative paths for each flight: the shortest path at the optimal altitude and the shortest path at the second-best altitude. With the above setup, we now present the formulation of the FO model.

Let us first specify the sets involved in the formulation. We use F to denote the set of UAV missions (flights) in the system. P denotes the set of paths each UAV mission can choose from. C denotes the set of conflict UAV flight pairs, which includes all the spatial conflict flight pairs as determined in subsection 3.1. Each element in set C includes IDs of the two spatially conflict UAV flights and the corresponding conflict paths. Each flight has two alternative paths. In addition to sets, we also introduce the decision variables and parameters, as shown in Table 5.

The formulation of the MILP is shown below. The objective function is the system total cost, including delay cost of all UAV missions, $\sum_{i\in F} D_i$, and path cost, $\sum_{i\in F, j\in P} c_{ij} I_{ij}$.

$$Min \sum_{i\in F} D_i + \sum_{i\in F,\ j\in P} c_{ij} I_{ij}$$

s.t.

$$D_i = \begin{cases} r_i(t_i - s_i) & if\ t_i - s_i \le d \\ r_i d + R_i(t_i - s_i - d) & if\ t_i - s_i > d \end{cases} \qquad \forall i \in F \tag{1}$$

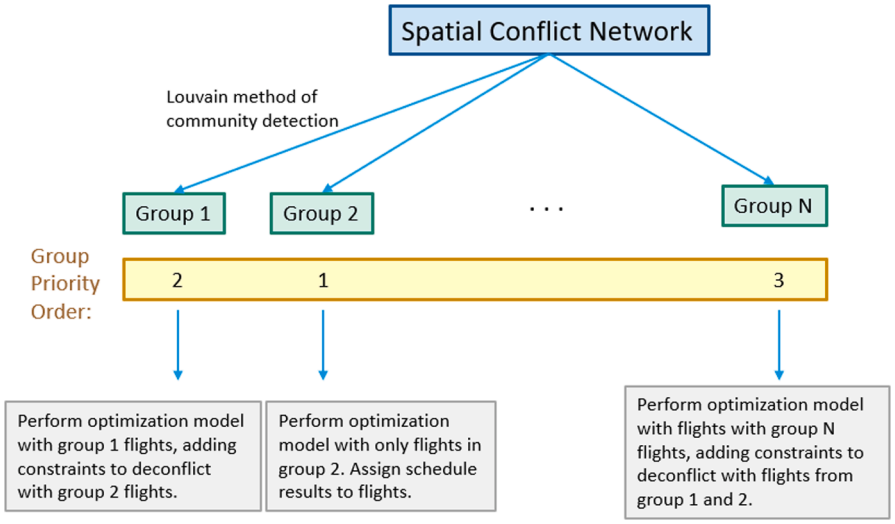

Fig. 4. The BO model

$$t_i + e_c + t_0 \le M(1 - X_c) + B_c + t_m \qquad \forall c = (i,j,m,n) \in C;\ i,\ j \in F;\ j,\ n \in P \tag{2}$$

$$t_m + E_c + t_0 \le M(1 - x_c) + b_c + t_i \qquad \forall c = (i,j,m,n) \in C;\ i,\ j \in F;\ j,\ n \in P \tag{3}$$

$$z_c = I_{ij} I_{mn} \qquad \forall c = (i,j,m,n) \in C;\ i,\ j \in F;\ j,\ n \in P \tag{4}$$

$$X_c + x_c = z_c \qquad \forall c = (i,j,m,n) \in C;\ i,\ j \in F;\ j,\ n \in P \tag{5}$$

$$t_i \ge s_i \qquad \forall i \in F \tag{6}$$

$$\sum_{j \in P} I_{ij} = 1 \qquad \forall i \in F \tag{7}$$

Constraint (1) specifies a piecewise linear delay cost function for each flight $i \in F$, with delay cost coefficient r_i when delay is less than threshold d. The delay cost coefficient becomes R_i when delay is larger than d. Constraints (2) and (3) resolve possible temporal conflicts from predetermined spatial conflict flight pairs. One of the big M terms, $M(1 - X_c)$ in (2) and $M(1 - x_c)$ in (3), will be cancelled and either one of two flights in spatial conflict may be delayed to avoid temporal conflict. If the big M term of constraint (2) is cancelled, flight i will pass the conflict region before flight m, and conversely if the big M term of (3) is cancelled. t_0 in constraints (2) and (3) is a safety separation time, which is the additional time delayed flight needs to wait to enter conflict region after the other flight exits. $t_0 = 10s$ in our research. Constraint (4) stipulates that if flights i and m take conflict paths j and n, z_c equals to one. Otherwise, $z_c = 0$. Constraint (5) requires that if flight i taking path j is spatially conflict with flight m taking path n, one of X_c and x_c equals to one. Constraint (6) means that the assigned departure time of any flight i must be no earlier than the flight's scheduled departure time. Constraint (7) ensures that an UAV flight will end up choosing one path.

In the formulation below, the nonlinear terms in the original optimization, i.e. constraints (1) and (4), are linearized and replaced by linear constraints (8)-(9) and constraints (10)-(12).

$$D_i \ge r_i(t_i - s_i) \qquad \forall i \in F \tag{8}$$

$$D_i \ge r_i d + R_i(t_i - s_i - d) \qquad \forall i \in F \tag{9}$$

$$z_c \le I_{ij} \qquad \forall c = (i,j,m,n) \in C;\ i,\ j \in F;\ j,\ n \in P \tag{10}$$

$$z_c \le I_{mn} \qquad \forall c = (i,j,m,n) \in C;\ i,\ j \in F;\ j,\ n \in P \tag{11}$$

$$z_c + 1 \ge I_{ij} + I_{mn} \qquad \forall c = (i,j,m,n) \in C;\ i,\ j \in F;\ j,\ n \in P \tag{12}$$

3.2.5. Batch optimization (BO) model

Our numerical experiments confirm that the computation time needed for solving the linearized integer program of the FO model (with constraints (8)-(12)) grows dramatically as N becomes large, to the extent that is impractical for strategic UTM. To overcome this and to solve scheduling problems with a large number of delayed flights (Samà et al., 2013), we propose the BO model to manage traffic with a large number of UAV flights. The BO model reduces the computational burden of the FO model by scheduling and routing

Table 6
Algorithm of the fourth step in BO model

Input	Group priority order; UAV flights group partition
Output	Schedules and assignments
1	*Assigned_schedules* = [ ]
2	Perform the optimization model on flights in the first group which has the highest priority, ignoring spatial conflicts with flights from other groups. Add flight assignment results to *Assigned_schedules*
3	**For** group i in priority order (from the second group):
4	Add constraints to optimization model to avoid temporal conflicts with flights in *Assigned_schedules*
5	Perform the optimization model on flights in group i with added constraints
6	Add flight assignment results to *Assigned_schedules*
7	**End for**

flights in groups. Specifically, UAV flights are clustered into groups based on a hierarchical decomposition of a "conflict network" formed by the flights through network community detection algorithms. Then we run the FO model described in subsection 3.2.4 for each group, based on the priority order which is determined according to the number of spatial conflicts that flights in a group have with flights from other groups.

Fig. 4 illustrates the algorithm for solving the BO model, which consists of four steps. In the first step, we assign scheduled departure time and optimal paths to those flights which have no spatial conflict with any other flights in the system. Then, we create a node-link spatial conflict network for the rest of flights, where a node denotes a UAV flight and a link indicates a spatial conflict between two UAV flights. Any two UAV flights with their paths overlapped in the airspace regardless of time will be connected by a link. The network is an undirected and weighted graph. The weight is determined by the travel time passing through the conflict regions of the two associated flights. Since each UAV flight has multiple alternative paths, all possible combinations of alternative path pairs will be checked for spatial conflicts. The weight is the sum of travel time passing through conflict regions if there exist multiple conflict path pairs between two UAV flights.

Once the spatial conflict network is constructed, the second step clusters flights in the network into different groups by the Louvain Method of Community Detection (Blondel et al, 2008), which is a percolation-inspired heuristic for community detection. The heuristic proceeds by iterating between two phases: the first phase is a greedy algorithm that places each node in a different community. Initially, there are as many communities as the number of nodes. Then each node is moved to its neighbor's community that gives the maximum gain in modularity. This is repeated randomly for all nodes until no more modularity improvement can be made. The second phase builds a new graph where the communities from the first phase are collapsed into a single node, and the edges between communities collapse into single edges between the respective nodes with equivalent weights. Then the first phase is repeated on this reduced graph, and the iterations continue until no changes occur. In this heuristic, modularity is a metric that measures how much community structure a network has. For a graph with c communities, its modularity is defined as:

$$Q = \frac{1}{2m}\sum_{v,w}\left[A_{v,w} - \frac{k_v k_w}{2m}\right]\delta(c_v, c_w) \tag{13}$$

where v and w denote nodes in the graph; $A_{v,w}$ represents the weight of the edge between v and w; k_v is the sum of the weights of the edges attached to v; c_v is the community to which v is assigned; the δ-function $\delta(u,v)$ is 1 if $u = v$ and 0 otherwise; $m = \frac{1}{2}\sum_{v,w}A_{v,w}$.

The third step assigns a priority order to each flight group. Different metrics may be used to assign the priority order, such as the sum of degree centrality within a cluster, the average scheduled departure time of cluster, and the cost increase of a cluster. The latter is measured by the cost difference between results from applying the optimization model on flights within the cluster (ignoring spatial conflicts with flights in other clusters) and the cost of all flights in the cluster taking their optimal paths and departing with no delay. The results using the three metrics will be compared in the next section.

In the fourth step, we incrementally optimize the routes and schedules of each flight group, based on the group priority order. At each iteration, we first add constraints to remove conflicts of flights in the current group with any flight in the groups of higher priority order. These constraints allow flights in the current group to avoid conflicts with determined schedules of flights in higher priority group by either delay or taking alternative paths. With the constraints, then we apply the optimization model to the flight group under consideration. The overall algorithm is described in Table 6.

3.3. Mechanism design

A key assumption made in the previous models is that a central controller has full and correct information about unit flight delay and path cost, which is private and will be provided by UAV operators. However, a possible issue may arise that the UAV operators misreport their private information in hopes to get more favorable routes or lower delay. Similar arguments are made in other contexts, such as parking and airport slot auction and allocation (Zou et al, 2015; Ball et al, 2020). To address the information misreporting issue, we propose to adapt the VCG mechanism (Vickrey, 1961; Clarke, 1971; Groves, 1973) which integrates traffic assignment with an accompanying payment scheme, so that each UAV operator has the incentive to truthfully reveal its private information. In other words, when the central controller aims to minimize system total cost based on private information submitted by UAV operators, no operator will find it beneficial to misreport—reporting private information other than the truth would not reduce the operator's cost,

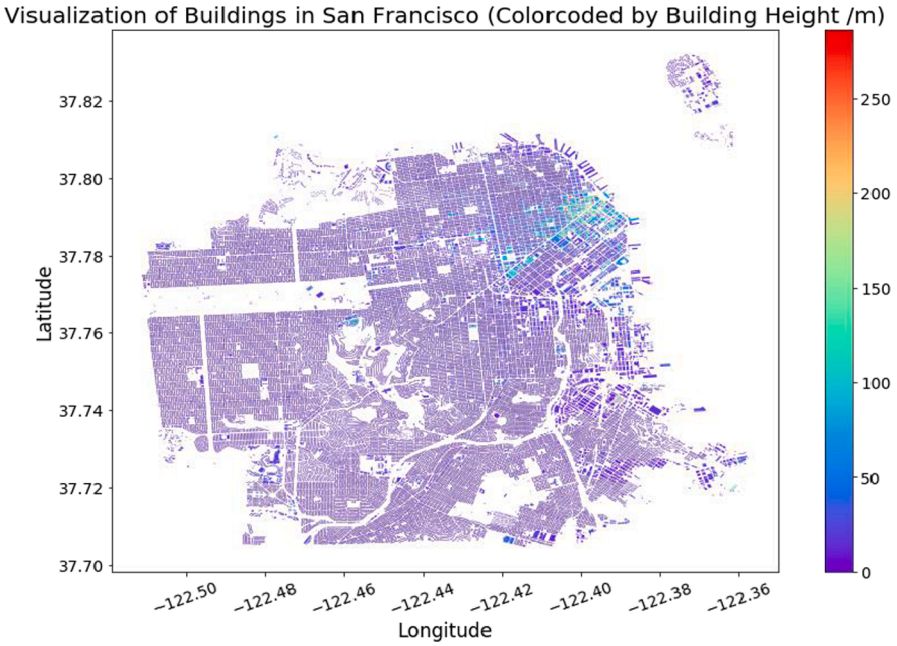

Fig. 5. San Francisco building height distribution

which consists of UAV flying cost and the payment. For example, if an operator reported a higher-than-actual delay cost, the system would assign a lower delay, but would also extract a higher payment from the operator, which would be no less than offset the delay cost saving.

More specifically, we assume that the central controller collects information from each UAV flight i on unit delay cost (r_i and R_i in the model formation in subsection 3.2.4), path cost by taking path j (c_{ij}) and scheduled departure time (s_i). The possible paths of each UAV flight, generated from section 3.1, are also submitted to the central controller. Based on the submitted information, the central controller assigns each UAV flight a specific path and departure time. In the meantime, the central controller asks for a payment from each UAV operator for the assigned path and departure time. In the case of one operator operating one UAV flight, the payment of each UAV operator (flight) i, $p(i)$, is calculated as:

$$p(i) = C_{\text{all}} - C_{\text{all}}(i) - C_{-i} \tag{14}$$

where C_{all} is the optimal system cost. $C_{\text{all}}(i)$ is the flying cost of UAV flight i in C_{all}. Thus, $C_{\text{all}} - C_{\text{all}}(i)$ is the system cost for all flights excluding flight i, while flight i is present in the system. C_{-i} is the optimal system cost while flight i is not present in the system. Thus, $p(i) = C_{\text{all}} - C_{\text{all}}(i) - C_{-i}$ gives the cost difference for all other flights with and without flight i being present—in other words, the "externality" of flight i on the rest of the system. In the case that one operator operates multiple UAV flights, the operator's payment will be the cost difference for all other flights with and without all the flights belonging to the operator being present. In this case, $C_{\text{all}}(i)$ in Eq. (14) would denote the flying cost of all UAV flights of operator i.

4. Simulation and experiments

The section presents the UTM model results in a simulated case study of UAV traffic in San Francisco. We simulate demand that may include food delivery from restaurants to customers and documents or package delivery between clients in San Francisco. Demand is assumed to be related to building height. The traffic management models developed in Section 3 are applied and compared. In addition, payments needed under the traffic management mechanism in subsection 3.3 are generated and analyzed for each UAV flight, based on the BO model for which their computation is tractable.

4.1. Data description

This paper takes San Francisco (SF) as the case study area. Building footprints data and topographic data of SF are used. The building footprints data is obtained from DataSF (DataSF, 2020). This dataset contains above-ground-level (AGL) height and boundary contour of structure roof and tree canopy. Zonal statistics are available for each building footprint based on a gridded representation of the vector polygon, discrete for every structure. We use the maximum value of 50 cm square grids sampled in each building's zone as the AGL building height.

Considering SF's hilly topology, it is necessary to merge the above AGL data with topographic data to obtain heights over sea level.

Table 7
Algorithm for demand generation

Input	Above-ground-level height $H[i]$ for building $i \in$ all San Francisco buildings; Top-view area $A[i]$ of building $i \in$ all San Francisco buildings; Number of OD pairs to be generated N
Output	A set of OD pairs
1	**For** building $i \in$ all San Francisco buildings:
2	Calculate a frequency-related factor $freq[i]$ that indicating the demand points frequency generated on each building: $freq[i] = (2 + max(H[i] - 4.3, 0)/4.3) \bullet A[i]$
3	Calculate cumulative value of $freq[i]$: $cumfreq[i] = \sum_{j=0}^{i} freq[j]$
4	**End for**
5	o = generate a set of $2N$ random values uniformly in range $(0, max(cumfreq[i]))$
6	d = generate a set of $2N$ random values uniformly in range $(0, max(cumfreq[i]))$ with different random seed
7	Define an empty set for OD coordinates: $ODpair = [\,]$
8	**For** k in $(0,\ 2N)$:
9	O_{index} is the index of the first building in the list satisfying $cumfreq[i] \geq o[k], \forall\ i \in$ all buildings, where the building list is in ascending order of $cumfreq$.
10	Origin point coordinate = Centroid of the footprint polygon of the building with index O_{index}
11	D_{index} is the index of the first building in the list satisfying $cumfreq[i] \geq d[k], \forall\ i \in$ all buildings, where the building list is in ascending order of $cumfreq$.
12	Destination point coordinate = Centroid of the footprint polygon of the building with index D_{index}
13	Add origin and destination coordinates to $ODpair$
14	**End for**
16	Calculate distance between origin and destination points for each OD in $ODpair$
18	Final demand OD pairs = $ODpair$ with distance between OD in range [1km, 10km]

Note: $freq[i]$ is proportional to the volume of building i, which indicates that buildings with larger volume generate more delivery demand. The algorithm generates OD points by uniformly randomly selecting from a sequence of consecutive numbers and assigns demand point to the corresponding building associated with the selected number, where there are $freq[i]$ consecutive numbers in the sequence representing building i. (E.g., for building 0, if $freq[0] = 3$, in the sequence of consecutive numbers, '0, 1, 2' represents demand from the first building. If number 2 is chosen randomly from the sequence under an iteration, there will be one demand point generated on building 0.)

The topographic data of SF is obtained from the National Oceanic and Atmospheric Administration (NOAA) Central California 1 Arc-second NAVD 88 Coastal Digital Elevation Model (DEM). Topographic elevation in this dataset is measured with reference to the vertical tidal datum of North American Vertical Datum of 1988 (NAVD 88). The original data with latitudes and longitudes in the geographic coordinate system of WGS 84 / EPSG 32610 UTM zone 10 N are converted to a Cartesian coordinate system based on a 1-meter grid.

We combine the two datasets to get the elevation of static obstacle with respect to NAVD 88 with topography instead of the original AGL building height. This results in a gridded discrete representation of elevation that incorporates both topography and buildings, at a grid resolution of 1 arc-second.

4.2. OD simulation

Origin and destination demand points for buildings are generated across the city. We assume that all UAV origin and destination points are on building rooftops. Taller buildings are assumed to generate more demand. The building height distribution is visualized in Fig. 5. Buildings in San Francisco are colored by above-ground-level building height.

In this paper, we assume a linear relationship between delivery demand generation/attraction and building height: $D = [K + \alpha(H - 1)]A$, where K is the demand for the first floor of building per square meter; H is the number of building floors; α is the demand per floor above the first floor per square meter; A is the area of building footprint. We assume $K > \alpha$ because retailer stores or restaurants are often located on the first floors. Specifically, based on Bard and Jarrah (2013), we set $K = 2\alpha$. We assume 4.3 m as the building floor height (Wikipedia, 2020). Other demand function types are also analyzed and compared including random demand and quadratic relationship between demand and building volume D. In quadratic case, OD points are more concentrated in SF downtown area with high-rise buildings, and UAV trip length is much shorter. It also seems to have the most conflicts and congestion. We only show results of linear demand function in the paper.

We randomly generate origin and destination points based on building height distribution such that the distance between origin and destination is between 1000 m and 10,000 m. The lower bound of this range reflects our estimate of the minimum distance at which an UAV may be used instead of some surface mode. The upper bound are based on UAV performance limits, including endurance and speed, with a margin for reaching a recharging point. The complete algorithm of OD generation is presented in Table 7. Though we only generate one-way UAV traffic, the random assignment of origin and destination points and different delay cost coefficient values (see Section 4.4) is intended to capture both "inbound" and "outbound" UAV delivery flights. 1000 UAV missions are generated for analysis.

4.3. Path planning results

UAV path planning is performed for each mission by generating horizontal obstacle-free paths at different altitudes and selecting the two altitudes that offer the lowest-cost paths. As discussed in Section 3.1, we represent obstacles by performing cluster analysis on

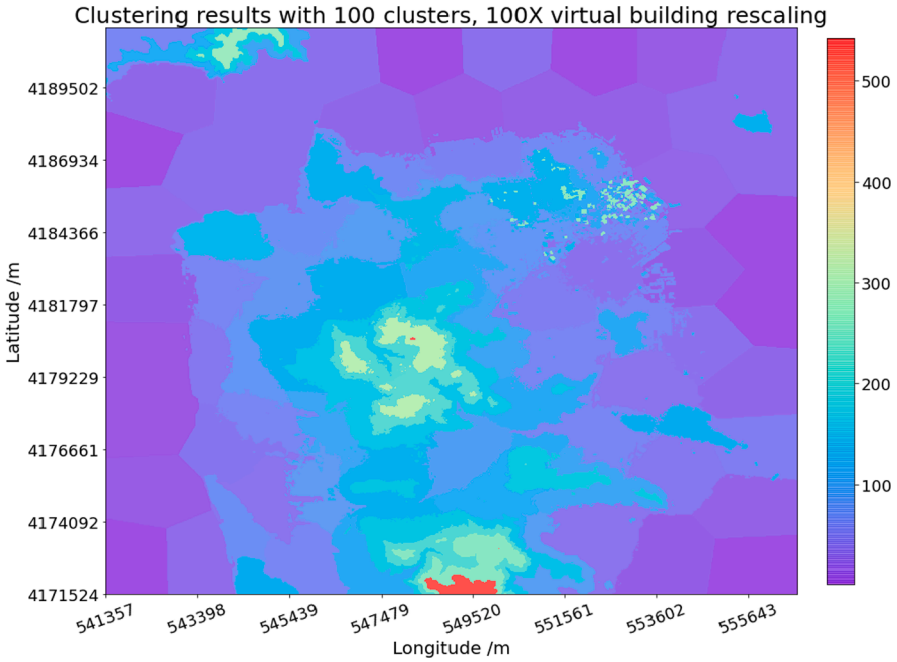

Fig. 6. Clustering results of San Francisco virtual buildings with 100 clusters and 100 times rescaling on virtual building height. Clusters are colored by elevation in meters

individual grid squares. We present the clustering results of 100 clusters in Fig. 6.

In Fig. 6, each cluster is colored by cluster elevation relative to NAVD 88 water level. Ocean areas are colored in purple. The virtual building height is highly influenced by topography especially for places with large topographical elevation changes. Twin peaks and San Bruno Mountain area have the highest obstacle elevation and are colored in red and light green. We also find sparse little obstacles colored in light green in the commercial area of San Francisco near Market Street, corresponding to high-rise buildings in the CBD area. The cluster elevation is determined by the maximum virtual building vertices' elevation within cluster. The final set of altitude candidates is reduced from 100 altitudes to 10 by combining cluster elevations within 30-meter vertical distance. The final 10 altitudes are 542, 310, 264, 215, 172, 139, 108, 77, 46, 12 m above NAVD 88 water levels.

As mentioned earlier, the horizontal shortest paths for each UAV mission are generated by the Saturated FM2 algorithm at each candidate altitude. Fig. 7 shows the horizontal shortest obstacle-free path results of an OD pair, at the altitudes of 215, 172, 139, 108, 77, and 46 m. The saturated FM2 algorithm is applied with 10-meter safe distance, which is the distance from the closest obstacle at which the maximum speed can be reached. Thus, the algorithm discourages flights from coming within 10 m of obstacles. The step size is set to be one arc-second. It can be seen that lower altitudes require more horizontal maneuvers to avoid obstacles, although vertical travel cost is less. By accounting for the cost of ascent, horizontal travel, and descent, the optimal flying altitude of this UAV mission is found to be 108 m. The second-optimal flying altitude is 139 m.

The distribution of the optimal altitudes for the 1,000 UAV flights is shown in Fig. 8. It can be seen that 108 m is the desired altitude for most flights. No flight prefers to fly at 12, 310, or 542 m, reflecting the large number of obstacles in the former case and the high vertical cost for minimal savings in horizonal cost in the latter two cases. Fig. 9 shows the distribution of the fractional cost difference between optimal and second-best paths to the optimal path cost, across the 1,000 UAV flights. The distribution is displayed using boxplots by the optimal flying altitude. The typical cost penalty from using the second-best path is 5 percent, but there is substantial variability, suggesting the advantage of re-assigning altitudes based on the cost penalties for individual flights. We also observe that UAV flights with higher least-cost altitudes tend to have lower cost difference fraction. Overall, the cost difference fraction of flights with optimal altitude at 264 m is the lowest among all altitudes, since most horizontal paths reach great circle distance between origin and destination starting from altitude 215 m.

4.4. Comparison of traffic management models

This subsection performs numerical analysis to compare the proposed four traffic management models. The following model parameters are adopted. For piecewise linear delay cost function, each UAV will be assigned a unit delay cost (r_i in model formulation) randomly picked from a uniform distribution in range $(0, 0.3\$/\text{min}]$ (Liu et al., 2019) to capture different urgency of UAV missions, when delay is less than five minutes. If delay exceeds five minutes, the unit delay cost will double (R_i). Though we do not directly consider inbound and outbound UAV deliveries separately, where half missions are with package and half are empty UAVs, the variability in r_i captures this as well as other factors. The unit horizontal path cost for each UAV is set to $\$0.2/\text{km}$ based on UAV

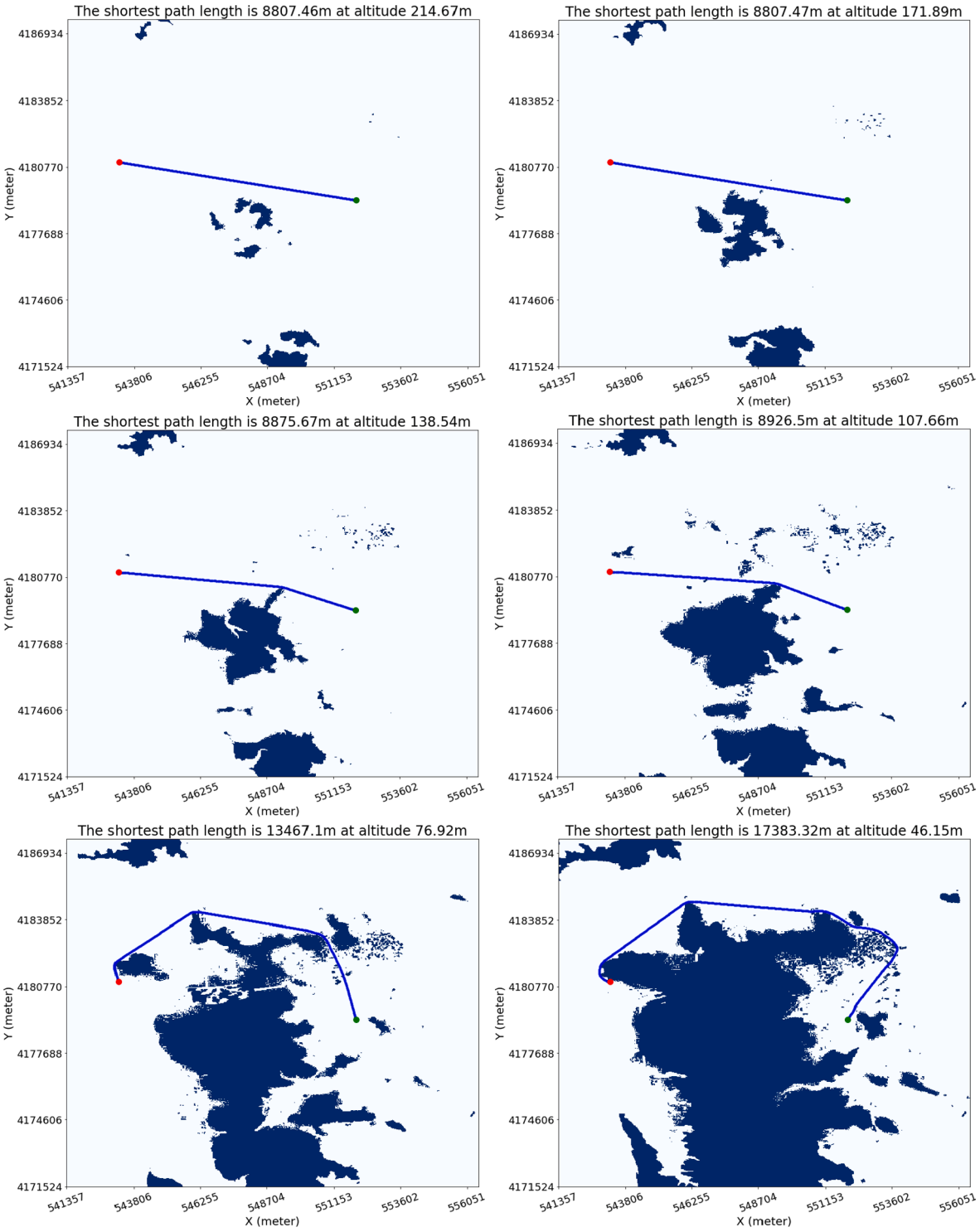

Fig. 7. Shortest horizontal obstacle-free path results by Saturated FM2 of one OD example

operational cost including equipment and software cost, labor cost, maintenance cost, and electricity cost (Sudbury and Hutchinson, 2016; Stolaroff et al, 2018).

We first consider a set of smaller-scale problems with 200 missions, for which the FO method is tractable. These problems enable us to compare the BO model results with the true optimum. Recall that for each UAV mission, both the optimal path and the second-best path at different altitudes, are generated as two alternative paths. Therefore, spatial conflict detection is performed pairwise on a total

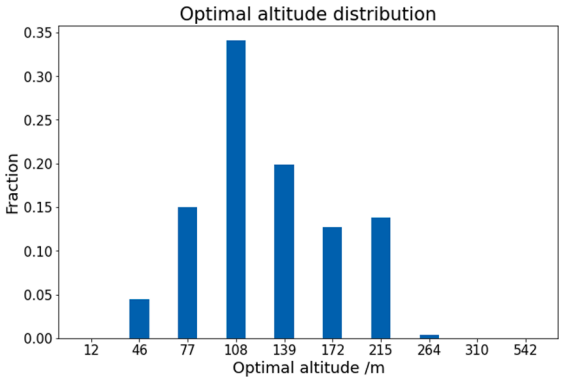

Fig. 8. Distribution of optimal altitudes for 1000 flights

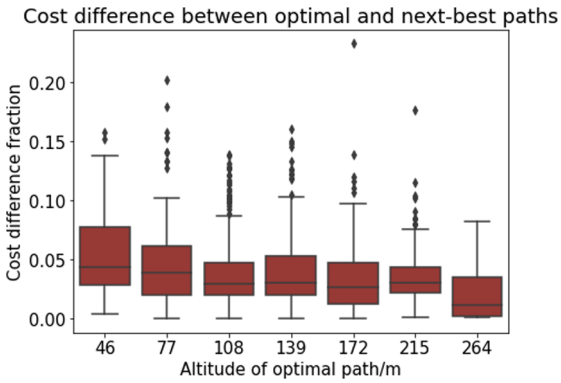

Fig. 9. Cost difference at different altitudes

Table 8
Comparison results of four traffic management models

		Length of simulation period (min)			
		60	**30**	**15**	**5**
Congestion cost ($)	FO	0.2	0.5	0.6	1.0
	BO	0.3	0.5	0.9	1.7
	SDR	0.4	1.6	3.6	4.2
	SD	5.8	15.6	19.7	20.6
% of congestion cost reduced from SD	FO	96.9%	96.8%	97.0%	95.1%
	BO	94.5%	96.5%	95.5%	91.5%
	SDR	93.7%	89.4%	81.9%	79.6%
% of delay reduced from SD	FO	93%	97%	92%	95%
	BO	91%	96%	97%	93%
	SDR	94%	87%	78%	87%
# delayed flights	FO	5	11	14	22
	BO	6	11	12	22
	SDR	6	12	18	19
	SD	18	28	31	57
# flights taking second-best path	FO	8	18	21	37
	BO	11	23	28	42
	SDR	11	19	20	43

of 400 UAV paths. In total, 4371 spatial conflicts are detected. For each spatial conflict path pair, travel time of each flight from its origin to the entry and exit points of the conflict region is computed, with 10-second safety buffer added. In other words, there must be at least 10 s between when one UAV departs the conflict region and the other enters it.

The scheduled departure time of the 200 UAV flights is randomly assigned within a time period. To simulate different traffic intensities, we consider different lengths of the time periods, of 60, 30, 15, and 5 min. To make sensible comparison between different period lengths, each flight's scheduled departure time changes proportionately with the period length. To illustrate, if the scheduled

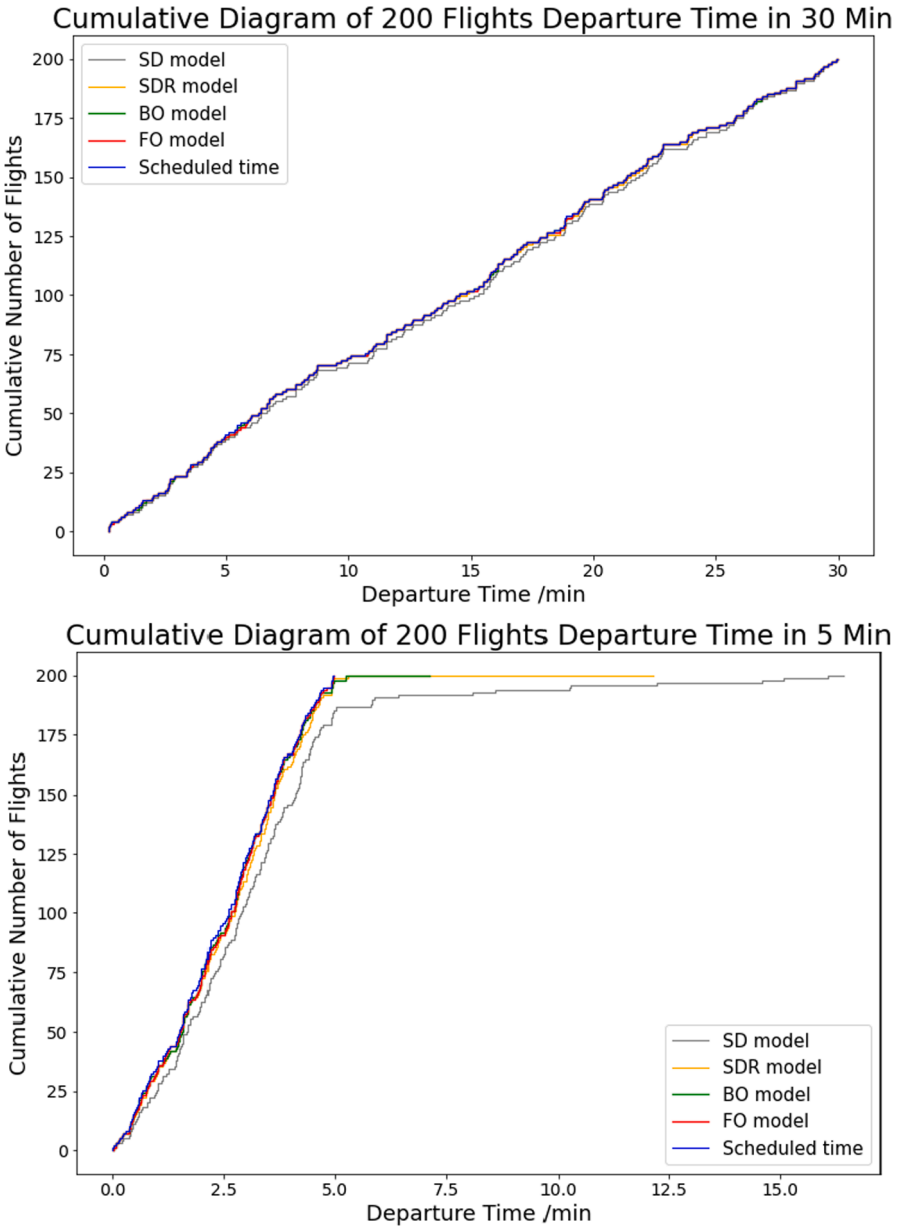

Fig. 10. Cumulative diagrams of UAVs assigned departure time from four models in 30- and 5-min cases

departure time of a UAV flight is at time 0:30 h for the total time period of one hour (60 min), then the same flight's scheduled departure time would be at 0:15 h if the total time period changes to half an hour (30 min). Note that the cases with time periods shorter than average UAV travel time (12 min for simulated UAV missions) are not realistic, since they represent "pulses" of high demand with no demand earlier or later. These cases are still considered so that the performance of the four models can be compared under congested conditions. Aside from the time period length, the same UAV flight schedules are used for the four traffic management models. When performing the BO model, we consider the sum of degree centrality within a cluster as the basis for determining priority order of clustered flight groups. Our experiments reveal that this metric yields a lower system cost than the other two alternatives (average scheduled departure time of a cluster, and cost increase of a cluster) mentioned in Section 3.2.5.

Table 8 shows the results. The congestion cost is calculated as the system cost by four proposed UTM models minus the ideal system cost, which is the system cost if all UAV flights taking their optimal paths without departure delay ignoring conflicts. The congestion cost reveals the additional cost of addressing conflicts by both departure delay and rerouting. As expected, the FO model yields the smallest congestion cost, which includes ground delay and use of second-best paths, while the SD model yields the largest. Also as expected, the congestion cost increases as the simulation period becomes shorter. Congestion cost is greatly reduced when the SDR model is employed, and further reduced when the optimization models are used. Comparing the latter, we find, as expected, that the

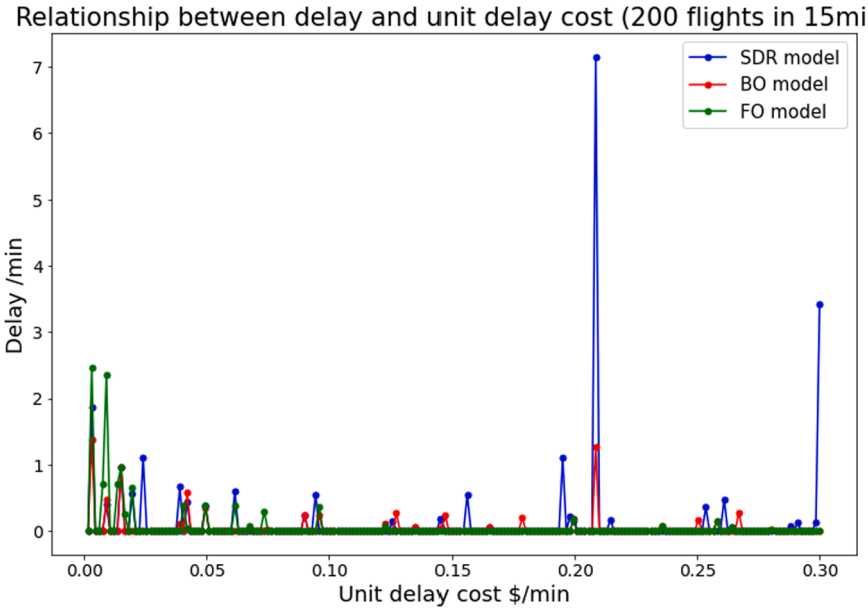

Fig. 11. Relationship between assigned delay and flight unit delay cost with 200 flights in 15 min

Table 9
Average model runtime (in seconds) over five runs, each with 200 flights

Average runtime	Length of simulation period (min)			
	60	**30**	**15**	**5**
FO	3.7	5.1	7.4	22.1
BO	4.9	5.1	5.3	5.9
SDR	20.6	20.3	20.1	21.0
SD	4.2	4.2	4.3	4.4

FO model yields slightly lower congestion cost than the BO model. The difference is small however, indicating that the latter is an acceptable alternative in light of its substantially lower computational burden. The percentages of congestion cost saving of the three models, which are calculated as the difference of the congestion cost with respect to the SD model divided by the congestion cost of SD model, are also presented. The FO model and the BO model have congestion cost savings of above 90% in all four lengths of the simulation periods. In contrast, due to its limited ability to take account of flight-specific cost information when resolving conflicts, the percentage of congestion cost savings for the SDR model diminishes as the simulation period becomes shorter. Overall, the comparison results suggest that while the BO model yields a slightly suboptimal solution, the cost penalty is modest. We will therefore focus on the BO model when we consider the 1000 flights case below.

The cumulative diagrams of the assigned UAV departure time for simulation period lengths of 30 and 5 min are presented in Fig. 10. In the 30-min case, the cumulative plots of four traffic management models are very close. However, the difference amplifies in the 5-min case as UAV traffic density increases. In the SDR model, the UAV traffic is saturated in the 5-min case: some flights are substantially delayed departing beyond the end of the 5-min period, as evidenced by the long 'tail' with some flights delayed by more than 10 min. A deeper look into the trajectories reveals that flights with large delay on the 'tail' are mostly sharing long path corridors with other flights. We will further explain the 'tail' in the next subsection.

Fig. 11 shows the relationship between the amount of delay and the unit delay cost (r_i in Eq. (1)) across the 200 flights in the case of 15-min time period for three models. For the FO model (plotted in green), very little delay is assigned to UAV flights unless they have very low delay cost. The SDR model results do not exhibit the same trend. For the BO model, though some delays are assigned to flights with moderate unit delay cost, no large delay is assigned to flights with high unit delay cost.

Table 9 shows the average runtime of the four models over five runs, each run with a different group of 200 UAV flights, for simulation periods of 60 min, 30 min, 15 min, and 5 min. We used a laptop with Intel i7-8750H CPU, 6 cores (2.2 GHz) and 16 GB RAM. The average runtime of the BO, SDR, and SD models remains rather constant for different simulation period lengths, but not the FO model. The runtime of SDR model is much higher than SD model because not only SDR model has twice amount of flight paths but also it computes best schedules for both possible paths correspondingly before determining the path. Though the average runtime of the FO model for a 5-min simulation period is similar to that of SDR model, the FO model runtime is expected to increase dramatically as UAV traffic becomes denser. Our experiments show that the computation time of the FO model takes more than 10 hours for the case with 1000 flights in a 30-minute period. The drastic increase in computation time of the FO model as opposed to the more stable

Table 10
Scheduling results of 1000 UAV flights with 30-min simulation period

Model	Congestion cost /$	Delay reduction from baseline %	# flights delayed	# alternative paths
BO	29	92%	243	353
SDR	77	85%	262	283
SD	811	0%	443	n/a

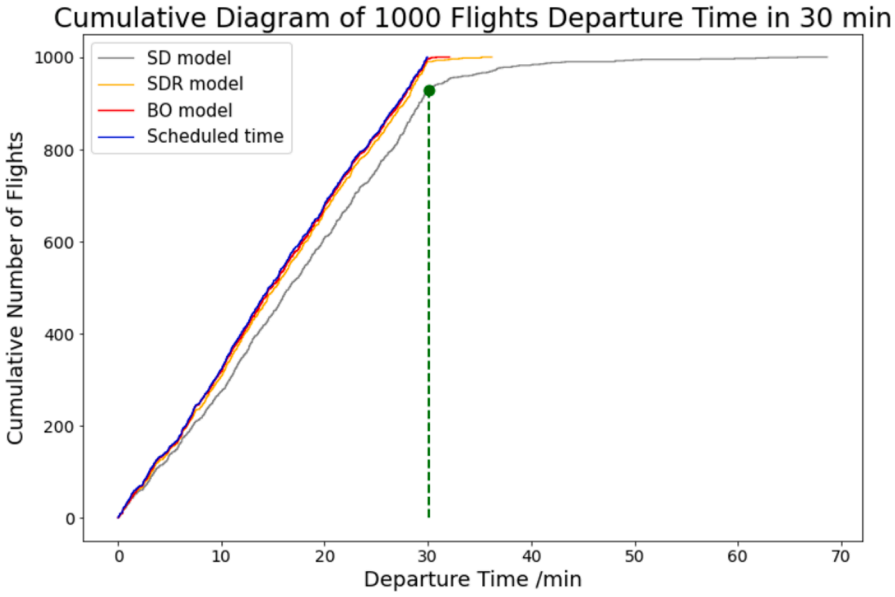

Fig. 12. Cumulative diagram of 1000 flights departing in 30 min by SD, SDR and BO models

computation time of the BO model motivates us to focus on the latter for large-scale numerical analysis, as presented below. The BO model is thus a good heuristic for solving large-scale problems for which the FO model is computationally intractable.

4.5. BO model results with 1000 flights

Given the promise of the BO model in generating good quality results with a reasonable amount of solution time, this subsection focuses on using the BO model for tackling larger-size problems. For the case of package delivery, we consider 1000 flights for a period of 30 min. In December 2019, around 100,000 packages are delivered in San Francisco daily (Lim, 2019). Assuming a 10-hour delivery window, this amounts 5000 packages delivered every 30 min. Thus, considering 1,000 flights corresponds to up to 20% of packages delivered by UAV. We use the same method as in subsection 4.2 to generate UAV flight ODs and scheduled departure times for 1000 flights as inputs for the BO model. 90,959 spatial conflict pairs are detected over the 2000 paths (recall each flight has two alternative paths).

For solving the BO model with 1000 flights, we first find that one flight is not spatially conflict with any of the remaining 999 flights. This flight is assigned its scheduled departure time and optimal path. The spatial conflict network is based on the remaining 999 flights. The spatial conflict network that is needed for solving the BO model thus has 999 nodes. Corresponding to the 90,959 spatial conflicts identified, 71,315 edges are constructed for the network. The number of edges is less than the number of spatial conflicts because two flights, each of which as two paths, can have multiple spatial conflicts. We use the sum of travel time through conflict regions between any two flights as the edge weight.

Next, the Louvain method is applied to detect communities in the spatial conflict network. The 999 flight nodes are clustered into 9 communities. Modularity is the degree to which a system's components may be separated and recombined, with a value between 0 and 1. The modularity of our network is 0.48. A community partition of a real-world network is generally good if modularity is between 0.3 and 0.7 (Yin et al, 2015).

With the clustering of flights, the BO model is then applied. As a reference for comparison, we also apply the SDR model and the SD model. Results in Table 10 show that the BO model generates schedules with significantly lower congestion cost. Compared to the SDR model and the SD model, the BO model assigns less delay to flights on average. Though fewer UAV flights take optimal paths using the BO model, the total cost is smaller than the other two models.

The cumulative diagrams of the 1000 UAV flights based on the SD, SDR, and BO models are shown in Fig. 12. We observe that the diagrams of the BO and the SDR models are much closer to the plot of the scheduled departure time than the SD model. The BO model

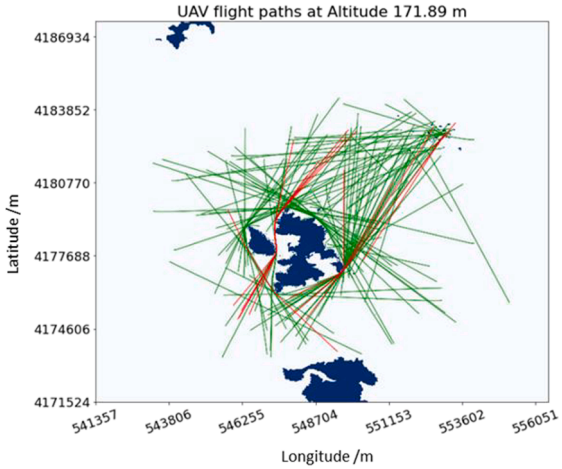

Fig. 13. UAV flight paths at altitude 172 m with flights on the 'tail' colored by red

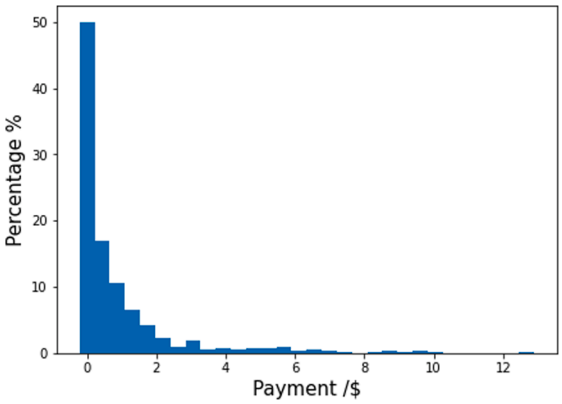

Fig. 14. Payment distribution of 1000 UAV flights

performs better than the SDR model, which is better than the SD model. Given the long 'tail', UAV traffic is saturated under the SD model. Many flights are delayed departing after the end of the 30-minute period, which is indicated by the vertical green dashed line. Fig. 13 visualizes delayed flights at the altitude of 172 m. In the figure, all flight paths assigned to this altitude under the SD model are plotted. Paths colored in red are flights on the 'tail' after the green dashed line, while all other flights are colored green. UAV flights that are assigned large delay by the SD model tend to share a long corridor with several other flights. Those flights need to wait longer time to avoid conflicts with multiple other flights passing through longer conflict regions. In contrast, the BO model and SDR model have much shorter 'tails' as shown in Fig. 12.

4.6. Payment analysis under mechanism design

Under the assumption that a VCG mechanism is in place, the resulting payment for each of the 1000 UAV flights is calculated for a 30-minute period. Again, the BO model as an approximation for system cost minimization is used. Fig. 14 shows the payment distribution of 1000 UAV flights. We find that about half of the flights have zero or very small payments. Fewer flights have higher payments. Flights with higher payments have larger influence on the delay of other flights in the system. Fig. 15 uses different colors to denote the payment of 1000 flights along with their paths assigned by the BO model at seven altitudes: 264, 215, 172, 139, 108, 77 and 46 m, which have 35, 114, 136, 243, 299, 128, 45 UAV flights respectively. No flight is scheduled to altitudes at 542, 310 and 12 m. Conceptually, a flight with more conflicts tend to impose more "externalities" on other flights, thereby incurring more payment. In line with the expectation, altitudes with fewer UAV flights (e.g. 46 and 264 m) tend to have lower payment ranges. At the altitude of 77 m, three flight paths on the left part of the graph do not conflict with any other flight at this altitude and have very low payment. However, the payment is non-zero as some other flights may be assigned their second-best paths to avoid conflict with these paths. At the altitude of 108 m, some flights have paths conflict with multiple other flight paths resulting in relatively high payments. The numbers of UAV flight paths are almost doubled at altitude 108 and 139 m compared to 77, 172 and 215 m, which also contribute to higher payments.

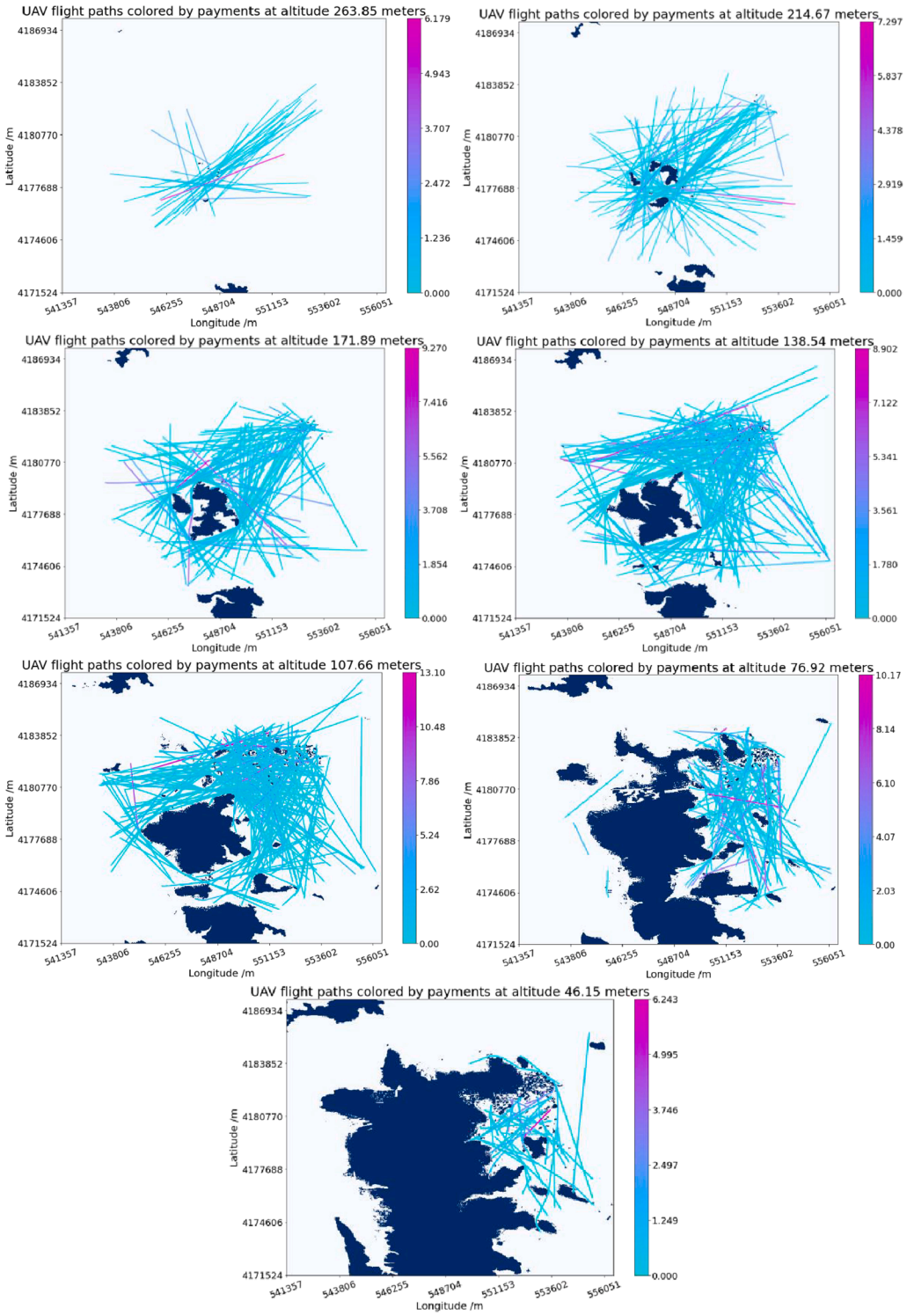

Fig. 15. Visualization of 1000 UAV paths colored by payments from BO model results

5. Conclusions

This research proposes a framework of UAS Traffic Management (UTM) including clustering-based UAV path planning, systematic UTM models with conflict resolution, and mechanism design for airspace resource allocation. The UAV path planning algorithm takes

advantage of clustering obstacles to generate representative altitude candidates, which is more efficient than the evenly spaced altitudes assumed in most research of UAV system planning and analysis. In determining the optimal travel path, the tradeoff between horizontal shortest path length and vertical travel distance at different altitudes is recognized. In addition, four traffic management models are proposed to systematically assigns schedules to UAV flights with conflict resolution. These traffic management models allocate spatial and temporal airspace resource to each UAV mission, while determining the departure time and which path to take for each UAV flight. We find that the Batch Optimization (BO) model, which can solve large-scale systems in short time while making a slight compromise on solution optimality, is more attractive than the Full Optimization (FO) and Sequential Delay/ Reroute (SDR) models to solve large-size problems. Finally, as traffic management requires private information from UAV operators, it is important that the private information received is truthful. To this end, a VCG-style mechanism is adapted, in which the payment made by an UAV flight is the "externalities" caused by the flight to the rest of the system. We find that the payment made by a UAV flight increases with traffic intensity and the extent of interactions a flight has with other flights.

This paper presents a beginning of integrating UTM with UAV delivery that deals with allocating spatial–temporal airspace resources to UAV missions. Multiple directions can be further explored in future research. Here we suggest a few. First, in this paper we propose the Batch Optimization (BO) model as an approximation of the Full Optimization (FO) model. Further research could be directed to investigating other heuristic approaches to solve the Full Optimization (FO) model, and compare the solution quality and computation efficiency. Second, stochastic factors could be introduced, for example, by accounting for uncertainties in weather and UAV performance (obstacle detection, flying stability, etc.) while determining the optimal and second-optimal paths of each UAV mission. Third, the performance of different models could be tested in more areas with varying population density, tall building concentration, and terrain types, especially given the particular landscape and topology of San Francisco. Doing so will help glean more generic insights about UTM. Lastly, as last-mile urban delivery is shifting toward nonmotorized vehicles and drones, it would be interesting to look into traffic management in a multimodal context involving UAVs and ground modes to enhance the overall efficiency of the delivery system while meeting demand.

CRediT authorship contribution statement

Ang Li: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Funding acquisition. **Mark Hansen:** Conceptualization, Methodology, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Bo Zou:** Conceptualization, Writing – review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was supported in part by the National Institute for Congestion Reduction (NICR) and by the National Science Foundation (NSF) under grant number CMMI-1663411. The financial support of NICR and NSF is gratefully acknowledged. Earlier versions of the paper were presented at 2020 and 2021INFORMS Annual Meetings, the third Symposium on Aviation Research, and the NICR webinar. We thank the attendees of our presentations and also the anonymous reviewers for their constructive feedback which helped us improve the paper.

References

Boysen, N., Briskorn, D., Fedtke, S., Schwerdfeger, S., 2018. (2018) Drone delivery fromtrucks: Drone scheduling for given truck routes. Networks. 72 (4), 506–527.
Botea, A., Mueller, M., Schaeffer, J., 2004. Near optimal hierarchical path-finding. J. Game Dev. 1, 1–30.
Ball, M.O., Estes, A.S., Hansen, M., Liu, Y., 2020. Quantity-contingent auctions and allocation of airport slots. Transport. Sci. 54 (4), 858–881.
Bagherian, M., Alos, A., October 2015. 3D UAV trajectory planning using evolutionary algorithms: a comparison study. Aeronaut. J. 119 (1220), 1271–1285.
Blondel, V.D., Guillaume, J., Lambiotte, R., Lefebvre, E. (2008) Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment. 2008 (10): P1,0008, arXiv: 0803.0476, doi: 10.1088/1742-5468/2008/10/P1,0008, S2CID 334423.DataSF (2020) Building Footprints https://data.sfgov.org/Geographic-Locations-and-Boundaries/Building-Footprints/ynuv-fyni (Accessed: 8 January 2021).
Bard, J.F., Jarrah, A.I., 2013. Integrating commercial and residential pickup and delivery networks: a case study. Omega, Elsevier 41 (4), 706–720.
Clarke, E.H., 1971. Multipart pricing of public goods. Public Choice 11 (1), 17–33.
Dorling, K., Heinrichs, J., Messier, G.G., Magierowski, S., 2017. Vehicle routing problems for drone delivery. IEEE Trans. Syst. Man Cybernet.: Syst. 47 (1), 70–85. https://doi.org/10.1109/TSMC.2016.2582745.
Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T., Jurišica, L., 2014. Path planning with modified a star algorithm for a mobile robot. Procedia Eng. 96, 59–69.
Fadhil, D.N., 2018. A GIS-Based Analysis for Selecting Ground Infrastructure Locations for Urban Air Mobility. Technical University of Munich, Munich, Germany. Master's thesis.
Ferrandez, S., Harbison, T., Weber, T., Sturges, R., Rich, R., 2016. Optimization of a truck-drone in tandem delivery network using K-means and genetic algorithm. J. Ind. Eng. Manage. JIEM 2016–9 (2), 374–388.
Foina, A.G., Krainer, C., Sengupta, R., 2015. An unmanned aerial traffic management solution for cities using an air parcel model. Proc. Int. Conf. Unmanned Aircraft Syst. (ICUAS) 1295–1300.
González-Varona, J.M., Villafáñez, F., Acebes, F., Redondo, A., Poza, D., 2020. Reusing newspaper kiosks for last-mile delivery in urban areas. Sustainability 12 (22), 9770.

Groves, T., 1973. Incentives in teams. Econometrica 41 (4), 617. https://doi.org/10.2307/1914085.
Haag, M., Hu, W. (2019) 1.5 Million Packages a Day: The Internet Brings Chaos to N.Y. Streets [Online]. Available at: https://www.nytimes.com/2019/10/27/nyregion/nyc-amazon-delivery.html (Accessed: 22 November 2020).
Ho, F., Salta, A., Geraldes, R., Goncalves, A., Cavazza, M., Prendinger, H., 2019a. Multi-Agent Path Finding for UAS traffic management. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '19). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp.131-139.
Ho, F., Geraldes, R., Goncalves, A., Rigault, B., Oosedo, A., Cavazza, M., Prendinger, H., 2019b. Pre-flight conflict detection and resolution for UAV integration in shared airspace: Sendai 2030 Model Case. IEEE Access 7, 170226–170237. https://doi.org/10.1109/ACCESS.2019.2954987.
Ho, F., Geraldes, R., Goncalves, A., Cavazza, M., Prendinger, H., 2019c. Improved conflict detection and resolution for service UAVs in shared airspace. IEEE Trans. Veh. Technol. 68 (2), 1231–1242.
Harabor, D., Botea, A., Kilby, P., 2011. Path Symmetries in Undirected Uniform-Cost Grids. SARA 2011 - Proceedings of the 9th Symposium on Abstraction, Reformulation, and Approximation.
Harabor, D., Grastien, A., 2011. Online Graph Pruning for Pathfinding on Grid Maps. National Conference on Artificial Intelligence (AAAI).
Hoffman, R., Burke, J., Lewis, T., Futer, A., Ball, M.O., 2012. Resource Allocation Principles for Airspace Flow Control, AIAA Guidance, Navigation, and Control Conference and Exhibit.
Josephs, L., 2019. UPS wins first broad FAA approval for drone delivery [Online]. Available at: https://www.cnbc.com/2019/10/01/ups-wins-faa-approval-for-drone-delivery-airline.html (Accessed: 22 November 2020).
Janjoš, F., Reichart, R., Niermeyer, P., 2017. Smooth Path-Generation Around Obstacles Using Quartic Splines and RRTs, IFAC-PapersOnLine, Volume 50, Issue 1, 2017, pp.9108-9113, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2017.08.1708.
Kafle, N., Zou, B., Lin, J., 2017. Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery. Transport. Res. B: Methodol. 99, 62–82. https://doi.org/10.1016/j.trb.2016.12.022.
Kopardekar,P., Rios, J., Prevot, T., Johnson, M., Jung J., Robinson III, J.E., 2016. Unmanned Aircraft System Traffic Management (UTM) Concept of Operations, in 16th AIAA Aviation Technology, Integration, and Operations Conference, AIAA Aviation, Washington, D.C.
Kim, J., Moon, H., Jung, H., 2020. Drone-based parcel delivery using the rooftops of city buildings: model and solution. Appl. Sci. 10 (12), 4362. https://doi.org/10.3390/app10124362.
Li, A., Hansen, M., 2020. Obstacle Clustering and Path Optimization for Drone Routing. 9th International Conference for Research in Air Transportation.
Lee, H.L., Chen, Y., Gillai, B., Rammohan, S., 2016. Technological Disruption and Innovation in Last-Mile Delivery. Retrieved from Stanford Graduate School of Business [Online]. Available at: https://www.gsb.stanford.edu/faculty-research/publications/technological-disruption-innovation-last-mile-delivery (Accessed: 22 November 2020).
Lim, J., Jung, H., 2017. Drone delivery scheduling simulations focusing on charging speed, weight, and battery capacity: Case of remote islands in South Korea. In Proceedings of the 2017 Winter Simulation Conference (WSC), Las Vegas, NV, USA, 3–6 December 2017; pp. 4550–4551.
Labib, N.S., Danoy, G., Musial, J., Brust, M.R., Bouvry, P., 2019. Internet of unmanned aerial vehicles-a multilayer low-altitude airspace model for distributed UAS traffic management. Sensors. 19 (21), 4779. https://doi.org/10.3390/s19214779.
Liu, Z., Sengupta R., Kurzhanskiy, A., 2017. A power consumption model for multi-rotor small unmanned aircraft systems, 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 2017, pp. 310-315, doi: 10.1109/ICUAS.2017.7991310.
Liu, Y., Yin, M., Hansen, M., 2019. Economic costs of air cargo flight delays related to late package deliveries. Transport. Res. E Logist. Transport. Rev. 125, 388–401. https://doi.org/10.1016/j.tre.2019.03.017.
Lim D., 2019. San Francisco shipping companies prepare for busiest mailing week of the year [Online]. Available at: https://abc7news.com/mail-post-office-will-my-package-ship-in-time-delivery/5760532/ (Accessed: 22 November 2020).
Le, T.V., Stathopoulos, A., Van Woensel, T., Ukkusuri, S.V., 2019. Supply, demand, operations, and management of crowd-shipping services: a review and empirical evidence. Transport. Res. C: Emerg. Technol. 103, 83–103.
Murray, C.C., Chu, A.G., 2015. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transport. Res. C: Emerg. Technol. 54, 86–109. https://doi.org/10.1016/j.trc.2015.03.005.
Murray, C.C., Raj, R., 2020. The multiple flying sidekicks traveling salesman problem: parcel delivery with multiple drones. Transportat. Res. C: Emerg. Technol. 110, 368–398.
Norman, H., 2019. Drone delivery launched by FedEx and Wing in first-of-its-kind trial in US [Online]. Available at: https://www.parcelandpostaltechnologyinternational.com/news/automation/drone-delivery-launched-by-fedex-and-wing-in-first-of-its-kind-trial-in-us.html (Accessed: 22 November 2020).
Ng, H.K., Sridhar, B., Grabbe, S., 2014. Optimizing aircraft trajectories with multiple cruise altitudes in the presence of winds. J. Aerospace Inform. Syst. 11 (1), 35–47. https://doi.org/10.2514/1.I010084.
Neufville, R.D., Odoni, A., 2003. Airport Systems: planning, design and management, first edition, 2003.
Palmer, A., 2020. Amazon wins FAA approval for Prime Air drone delivery fleet [Online]. Available at: (Accessed: 22 November 2020).
Perez, S., 2020. COVID-19 pandemic accelerated shift to e-commerce by 5 years, new report says [Online]. Available at: https://techcrunch.com/2020/08/24/covid-19-pandemic-accelerated-shift-to-e-commerce-by-5-years-new-report-says/ (Accessed: 22 November 2020).
Peinecke N., Kuenz, A., 2017. Deconflicting the urban drone airspace, 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC), St. Petersburg, FL, 2017, pp. 1-6. doi: 10.1109/DASC.2017.8102048.
Rath, S., Chow, J.Y.J. (2019) Air Taxi Skyport Location Problem for Airport Access. [Online]. Available at: http://arxiv.org/pdf/1904.01497v2 (Accessed: 22 November 2020).
Ren, L., Castillo-Effen, M., Yu, H., Johnson, E., Yoon, Y., Takuma, N., Ippolito, C.A., 2017. Small Unmanned Aircraft System (Suas) Categorization Framework for Low Altitude Traffic Services. In 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC), St. Petersburg, FL, USA, September. doi: 10.1109/DASC.2017.8101996.
Ranieri, L., Digiesi, S., Silvestri, B., Roccotelli, M., 2018. A review of last mile logistics innovations in an externalities cost reduction vision. Sustainability 10 (3), 782.
Scherer, S., Ferguson, D., Singh, S. (2009) Efficient C-space and cost function updates in 3D for unmanned aerial vehicles. in IEEE international conference on robotics and automation, ICRA'09 (pp.2049–2054). New York: IEEE Press, 2009.
SupplyChainBrain (2020) Who's Winning in the Small-Parcel Delivery Market? [Online]. Available at: https://www.supplychainbrain.com/articles/31160-qa-whos-winning-in-the-small-parcel-delivery-market (Accessed: 22 November 2020).
Sudbury, A.W., Hutchinson, E.B., 2016a. A Cost Analysis of Amazon Prime Air (Drone Delivery). J. Econ. Educat. Middle Tennessee State University, Business and Economic Research Center 16 (1), 1–12. https://ideas.repec.org/a/mts/jrnlee/v16y2016i1p1-12.html.
Sathyaraj, B.M., Jain, L.C., Finn, A., Drake, S., 2008. Multiple UAVs path planning algorithms: a comparative study. Fuzzy Optim Decis Making 7 (3), 257–267. https://doi.org/10.1007/s10700-008-9035-0.
Sethian, J.A., 1999. (1999) Fast-Marching methods. SIAM Rev. 41 (2), 199–235.
Shavarani, S.M., Nejad, M.G., Rismanchian, F., Izbirak, G., 2018. Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of Amazon prime air in the city of San Francisco. Int J Adv Manuf Technol 95 (9-12), 3141–3153. https://doi.org/10.1007/s00170-017-1363-1.
Samà, M., D'Ariano, A., Pacciarelli, D., 2013. Rolling Horizon Approach for Aircraft Scheduling in the Terminal Control Area of Busy Airports, Procedia - Social and Behavioral Sciences, Volume 80, 2013, pp.531-552, ISSN 1877-0428, https://doi.org/10.1016/j.sbspro.2013.05.029.
Sudbury, A.W., Hutchinson, E.B., 2016. A Cost Analysis of Amazon Prime Air (Drone Delivery), J. Econ. Educat. Middle Tennessee State University, Business and Economic Research Center, vol. 16(1), pp.1-12, Fall.
Stolaroff, J.K., Samaras, C., O'Neill, E.R., Lubers, A., Mitchell, A.S., Ceperley, D., 2018. Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. Nat. Commun. 9 (1) https://doi.org/10.1038/s41467-017-02411-5.

Torabbeigi, M., Lim, G.J., Kim, S.J., 2020. Drone delivery scheduling optimization considering payload-induced battery consumption rates. J. Intell. Robot. Syst. 97 (3-4), 471–487. https://doi.org/10.1007/s10846-019-01034-w.
Unmanned Airspace, 2019. NLR and everis ADS to collaborate on UTM concepts [Online]. Available at: https://www.unmannedairspace.info/uncategorized/nlr-and-everis-ads-to-collaborate-on-utm-concepts/ (Accessed: 22 November 2020).
Vascik, P.D., Hansman, R.J., 2019. Development of vertiport capacity envelopes and analysis of their sensitivity to topological and operational factors. AIAA Scitech 2019 Forum. San Diego, California.
van Duin, J.H.R., Wiegmans, B.W., van Arem, B., van Amstel, Y., 2020. From home delivery to parcel lockers: a case study in Amsterdam. Transp. Res. Procedia 46, 37–44.
Valero-Gomez, A., Gómez, J., Garrido, S., Moreno, L., 2013. Fast marching methods in path planning. IEEE Rob. Autom. Mag. 20, 111–120.
Vickrey, W., 1961. Counterspeculation, auctions and competitive sealed tenders. J. Finance 16 (1), 8–37.
Weinert, A., Campbell, S., Vela, A., Schuldt, D., Kurucar, J., 2018. Well-clear recommendation for small unmanned aircraft systems based on unmitigated collision risk. J. Air Transport. 26 (3), 113–122. https://doi.org/10.2514/1.D0091.
Weibel, R., Edwards, M., Fernandes, C., 2012. Establishing a Risk-Based Separation Standard for Unmanned Aircraft Self Separation, 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference.
Wikipedia, Storey [Online]. Available at: https://en.wikipedia.org/wiki/Storey (Accessed: 22 November 2020).
Yang, J., Yin, D., Niu, Y., Zhu, L., 2016. Cooperative conflict detection and resolution of civil unmanned aerial vehicles in metropolis, Adv. Mech. Eng., vol. 8, no. 6, 2016, Art. no. 1687814016651195.
Young, L., 2020a. Amazon signs lease for big South Bronx warehouse [Online]. Available at: https://www.bizjournals.com/newyork/news/2020/06/10/amazon-leases-south-bronx-warehouse.html (Accessed: 22 November 2020).
Young, L., 2020b. Amazon signs another warehouse lease in New York City [Online]. Available at: https://www.bizjournals.com/newyork/news/2020/06/24/amazon-signs-warehouse-lease-in-nyc.html (Accessed: 22 November 2020).
Yin, C., Zhu, S., Chen, H., Zhang, B., David, B., 2015. A method for community detection of complex networks based on hierarchical clustering. Int. J. Distrib. Sens. Netw. 11 (6), 849140. https://doi.org/10.1155/2015/849140.
Zou, B.o., Kafle, N., Wolfson, O., Lin, J.J., 2015. A mechanism design based approach to solving parking slot assignment in the information era. Transport. Res. B: Methodol. 81, 631–653.

Further reading

Cho, J., Yoon, Y., July 2018. How to assess the capacity of urban airspace: a topological approach using keep-in and keep-out geofence. Transport. Res., Part C: Emerg. Technol. 92, 137–149. https://doi.org/10.1016/j.trc.2018.05.001.
Haddad, J., Mirkin, B., Assor, K., 2021. Traffic flow modeling and feedback control for future Low-Altitude Air city Transport: An MFD-based approach, Transportation Research Part C: Emerging Technologies, Volume 133, 103380, ISSN 0968-090X, https://doi.org/10.1016/j.trc.2021.103380.
Ribeiro, M.J., Ellerbroek, J., Hoekstra, J.M., 2019. Analysis of Conflict Resolution Methods for Manned and Unmanned Aviation Using Fast-Time Simulations. 9th SESAR Innovation Days 2nd – 5th December 2019. http://resolver.tudelft.nl/uuid:73d8d050-9755-4dde-bf36-c6c0cc305f37.
Sedov, L., Polishchuk, V., 2018. Centralized and Distributed UTM in Layered Airspace. In: 8th International Conference on Research in Air Transportation, June 2018.
Sunil, E., Ellerbroek, J., Hoekstra, J., Vidosavljevic, A., Arntzen, M., Bussink, F., Nieuwenhuisen, D., 2016. Analysis of airspace structure and capacity for decentralized separation using fast-time simulations. J. Guid., Control Dyn. 40 (1), 38–51.
Vascik, P.D., Cho, J., Bulusu, V., Polishchuk, V., 2020. Geometric approach towards airspace assessment for emerging operations. J. Air Transport. 28 (3), 124–133.