25 Jul 2022

Drone-Truck Cooperated Delivery under Time Varying Dynamics

Arindam Khanda

Federico Corò

Sajal K. Das
Missouri University of Science and Technology, sdas@mst.edu

Recommended Citation

A. Khanda et al., "Drone-Truck Cooperated Delivery under Time Varying Dynamics," *ApPLIED 2022 - Proceedings of the 2022 Workshop on Advanced Tools, Programming Languages, and PLatforms for Implementing and Evaluating Algorithms for Distributed Systems*, pp. 24 - 29, Association for Computing Machinery (ACM), Jul 2022.
The definitive version is available at https://doi.org/10.1145/3524053.3542743

Drone-Truck Cooperated Delivery Under Time Varying Dynamics

Arindam Khanda
akkcm@mst.edu
Missouri University of Science and Technology
Rolla, MO, USA

Federico Corò
federico.coro@mst.edu
Missouri University of Science and Technology
Rolla, MO, USA

Sajal K. Das
sdas@mst.edu
Missouri University of Science and Technology
Rolla, MO, USA

ABSTRACT

Rapid technological developments in autonomous unmanned aerial vehicles (or drones) could soon lead to their large-scale implementation in the last-mile delivery of products. However, drones have a number of problems such as limited energy budget, limited carrying capacity, etc. On the other hand, trucks have a larger carrying capacity, but they cannot reach all the places easily. Intriguingly, last-mile delivery cooperation between drones and trucks can synergistically improve delivery efficiency.

In this paper, we present a drone-truck co-operated delivery framework under time-varying dynamics. Our framework minimizes the total delivery time while considering low energy consumption as the secondary objective. The empirical results support our claim and show that our algorithm can help to complete the deliveries time efficiently and saves energy.

CCS CONCEPTS

• **Computing methodologies** → **Distributed algorithms**.

KEYWORDS

Distributed Delivery System, Drone, Dynamic graph

ACM Reference Format:
Arindam Khanda, Federico Corò, and Sajal K. Das. 2022. Drone-Truck Cooperated Delivery Under Time Varying Dynamics. In *Proceedings of the 2022 Workshop on Advanced tools, programming languages, and PLatforms for Implementing and Evaluating algorithms for Distributed systems (ApPLIED '22), July 25, 2022, Salerno, Italy*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3524053.3542743

1 INTRODUCTION

Drones can be used in a plethora of applications including, but not limited to, monitoring [11], localization [1], surveillance service [2], precision agriculture [15], search and rescue [5], package delivery [9, 21]. Compared with the truck-based delivery system, a drone can deliver more efficiently due to various advantages such as its capability to deliver in hard-to-reach places or overcome traffic congestion on roads. However, there are several challenges involved in a drone-based delivery system, summarized as follows.

Since drones are powered by limited capacity batteries, they are forced to recharge the battery again after each trip to a customer. Also, the energy requirement for a fixed route can vary due to various external dynamics like wind [9, 24]. Drones can carry a maximum payload mass to customers, and the consumed energy for delivering also depends on the actual payload. Due to technical constraints, a drone can deliver a single package at a time [18]. Therefore, in a simple drone-based delivery system a drone picks up the package from a warehouse, delivers it to a customer, and then returns to the warehouse again before the next delivery. Drones can serve a limited area due to their limited communication range [19].

Drone-truck cooperation can address some of these challenges directly as a truck can carry several packages and can be used as a mobile charging station. However, a *drone-truck co-operated delivery system* (DTCDS) requires coordination between the drones and the truck. An energy and time-efficient solution to deliver all the items in such a system becomes more complex under time-varying dynamics as the time of traveling a fixed distance becomes variable. More specifically, road traffic can affect the truck's movement whereas the wind speed and direction can significantly impact the flying duration of a drone to traverse a fixed distance. In our paper, we will consider as a time-varying dynamic factor for the drone the wind, in particular, we will use the relative speed and energy consumption model (under varying wind) presented in [21].

The major contributions of this paper are as follows.

- To the best of our knowledge, we are the first to consider time-varying dynamics in a DTCDS scenario. Due to the time-dependent parameters travel time for both truck and drone becomes unpredictable, and efficient route selection becomes complex.
- We devise a distributed delivery setup, where each drone and truck recompute their route independently under time-varying dynamics, and synchronization between any drone and the truck is formulated by message passing.
- We present a greedy algorithm to minimize total delivery time by reducing the waiting time for the truck to gather the drones returning after delivery.
- As a secondary objective our proposed algorithm minimizes the energy consumption of the drones without increasing the total delivery time.

The rest of the paper is structured as follows. Section 2 surveys the related works. In Section 3, we devise a drone-truck co-operated delivery system. We present our drone-truck cooperation framework to minimize the delivery time in Section 4. Section 5 shows the experimental results. Finally, Section 6 concludes the paper and offers future directions.

ApPLIED '22, July 25, 2022, Salerno, Italy

ACM ISBN 978-1-4503-9280-8/22/07…$15.00
https://doi.org/10.1145/3524053.3542743

2 RELATED WORK

This section first reviews the literature related to the problem of delivering packages with the help of trucks and drones. Then it discusses the dynamics that can affect the delivery. As the shortest path algorithm is the core of our delivery route computation, recent parallel approaches for computing it are also reviewed.

Drone-Truck Cooperated Delivery: The DTCDS scenario was first investigated in [13] where the authors considered both the vehicle and one single drone to perform deliveries. A similar approach is studied for multiple drones in [7, 14].

In [6], a greedy heuristic is proposed where the approach first finds a solution for the truck only, and then it greedily builds subpaths for the drones by removing some deliveries from the truck to reduce the overall time. In [17], the authors consider that the truck has less speed than the drones due to traffic congestion and proposed an optimal mixed-integer linear programming formulation based on timely synchronizing. A hybrid approach is presented in [25], where the authors propose to simultaneously employ trucks, truck-carried drones, and independent drones to construct a more efficient delivery system.

In [3], the authors consider a predefined route for the truck, and the problem is to optimize the planning of the drone's flights to/from the truck while serving customers. The goal is to determine the launch and meeting points between drones and trucks. However, they do not assume a battery constraint for drones and their goal is to reduce the total time difference between the start and end of a delivery sequence. A multi-drone scheduling problem in a similar delivery model is introduced in [20]. The authors devise an optimal Integer Linear Programming model and propose multiple greedy heuristic algorithms to solve the problem. In our paper, we also consider that the route for the truck is predefined. However, we do not focus on the scheduling problem. Rather we consider dynamic delivery routes and minimize the total delivery time by reducing wait time for the truck.

Dynamics Affecting Drone-based Delivery: In [21], the authors investigate the Mission-Feasibility Problem for a drone-based delivery system (DBDS) in varying global wind conditions. A drone's relative speed and estimated energy consumption in different wind speeds and directions are computed to find the feasibility of a delivery in this paper. Delivery route computation in a centralized DBDS is proposed in [9]. The authors assume a model where all the computations are done at a centralized server and updated instruction is sent to the drones to complete the deliveries efficiently. The mission planning problems for drones under weather uncertainty are studied in [24]. The mathematical formulation considers the demand of goods at the delivery point, collision avoidance, and customer satisfaction along with factors like wind conditions and ground speed of the drones.

Algorithms for Finding Delivery Paths: If the delivery area is modeled as a graph where edge weights are energy requirements to travel from one endpoint to another, then the Single Source Shortest Path (SSSP) problem provides the most energy-efficient route.

A high-performance parallel graph library, Gunrock [26] provides a data-centric abstraction on a set of vertices or edges and uses a three-step architecture (advance, filter, and compute) to compute SSSP on GPUs. Their algorithm only focuses on static graphs. In [4], a GPU implementation of the Bellman-Ford shortest path algorithm is proposed. This algorithm exploits dynamic parallelism. A detailed study on the performance of various algorithms on graphs including SSSP on temporal graphs, different multi-core architectures, and GPU accelerators, is proposed in [16].

A dynamic incremental and decremental SSSP algorithm is implemented using JavaScript in [8]. However, the results in this paper show that the algorithm performs well only if the number of changed edges is less than 10%. Srinivasan et al. [23] propose the first shared-memory algorithm for updating SSSP on dynamic networks, and implemented it using OpenMP. A parallel algorithm template for updating SSSP in large-scale dynamic networks is proposed in [10]. The authors deal with generic undirected graphs and present a computing-architecture independent solution. We use a similar approach to update delivery routes efficiently under time-varying dynamics.

3 SYSTEM MODEL

Let A be the 2-D *delivery area* representing our last-mile delivery scenario. Let $\psi \in A$ be the *depot* from where the deliveries start. At the depot, a *truck* is in charge of transporting a fleet of m *drones* $Q = q_1, \ldots, q_m$.

The truck *does not* perform deliveries; it only carries the drones within A. Let ρ_j be a road segment connecting two rest areas λ_j and λ_{j+1}, where $j = 0, \ldots, r$ and $\Lambda = \{\lambda_1, \ldots, \lambda_r\} \subset A$. The truck leaves the depot ψ visiting the first rest area λ_1 along the road segment $\rho_0 = \overline{\psi\lambda_1}$, then the next endpoint λ_2 along the segment $\rho_1 = \overline{\lambda_1\lambda_2}$, and so on, up to the last endpoint λ_r, and eventually going back to the depot ψ. These segments make a closed path (*cycle*) C formed by the sequence of endpoints $\lambda_0, \lambda_1, \ldots, \lambda_r, \lambda_{r+1}$ such that $\lambda_0 = \lambda_{r+1} = \psi$

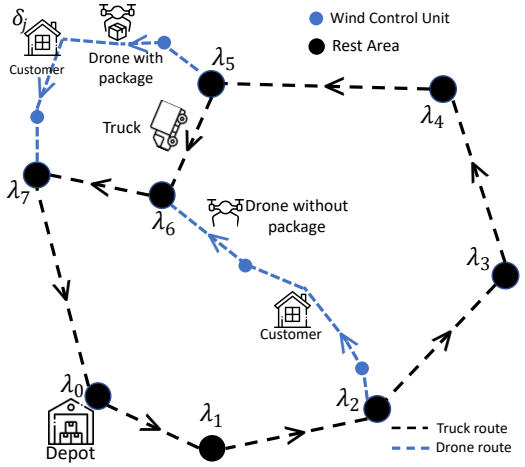


Figure 1: A Drone-Truck cooperated delivery system

Let $D = \{\delta_1, \ldots, \delta_n\} \subset A$ be the customers' locations to be served by the drones. For a set of deliveries, the path for the truck is fixed. However, an efficient delivery path for a drone can be affected at the time of flying due to time-varying dynamics such as wind speed/direction. Therefore, for each drone, we consider having a weighted dynamic graph $G = (V, E, w)$, where the set of nodes V consists of the set of deliveries D, a set of wind control units, and

the set of rest areas on the truck path. Figure 1 shows a DTCDS, where the truck follows the black route and two drones follow their delivery paths marked in blue.

The edge weights can be the time to fly from one endpoint of an edge to another or the energy consumption to fly the same. The relationship between time and energy is not linear due to factors like wind direction, wind speed, and path direction. Therefore, an energy-efficient path may not be the most time-efficient path. Depending on the edge weight we consider G^T and G^E are the graphs where edge weights are the time to fly and energy requirement respectively. Let t_i^T (resp. t_i^E) be the time when a drone starting from a fixed δ_i^L reaches δ_i^R after completing the delivery δ_i, by following the most time-efficient path (resp. energy-efficient path). t_i^T and t_i^E can be initialized by finding the shortest path in G^T and G^E respectively. However, the shortest path can be updated at the flying time due to the time-varying factors changing edge weights in the graph.

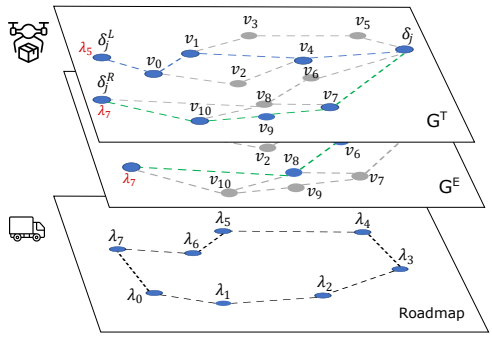


Figure 2: Truck has a fixed path starting from the depot. A drone should have two graphs G^E and G^T for each delivery.

A drone's delivery is performed by planning a sub-flight in G. Specifically, the drones take off (with a package), deliver packages to the customers, and return to the rendezvous locations with the truck again. For each customer's location δ_i, let δ_i^L and δ_i^R be respectively the *launch point* and *rendezvous point* of the drone. Therefore, the sub-flight for the drone will be a path either in G^E or G^T from δ_i^L to δ_i, and a path from δ_i to δ_i^R. Let a drone can take off or return to the truck only when the truck is at a rest area. Therefore, $\delta_i^L, \delta_i^R \in \Lambda$. Figure 2 shows the delivery graphs stored in the truck and the drones. Each drone stores both G^T and G^E, whereas the truck stores only the fixed road-map to complete the deliveries.

Note that in our problem we consider knowing in advance the set of deliveries D and the set of launch points δ_i^L for each delivery i, however, since the graph is dynamic, we are considering that the paths that each drone has to take and the rendezvous points with the truck are not known in advance. When both the truck and the drone arrive at δ_i^R, the truck gathers again the drones and they continue to travel up to point λ_{r+1}. The time spent by the truck at each rest area to gather the drones can be considered as waiting time. Let the tentative waiting time at λ_j be ω_j. Let the truck starts from a rest area λ_j at τ_j tentatively and $\mathcal{T} = \{\tau_0, \tau_1, \ldots, \tau_r\}$. Initially, it can be computed by finding a tentative speed of the truck and a fixed waiting time at each rest area. W.l.o.g., we can consider that the truck reaches the depot at τ_{r+1}. Note that, to save time, the truck does not travel back and forth for picking up the drones.

Problem formulation: In our scenario, the truck leaves the depot with a series of assigned deliveries and returns to the depot after completing all deliveries. We define the total delivery time as the difference between the time the drone reaches the depot after making all the deliveries, and the time the truck leaves the depot for delivery. The primary objective of this paper is to minimize such total delivery time. Formally, we intend to minimize $O_1 = (\tau_{r+1} - \tau_0) = \sum_{j=0}^{r}((\text{travel time of } \rho_j) + \omega_j)$. Note that, as the route for the truck is fixed, and traffic is uncertain, only ω_j can be modified to effectively minimize O_1. Thus, our first objective is to minimize the waiting time.

Our secondary objective is to minimize the energy consumption of the drones without increasing the total delivery time. If a set of drones are gathered at a rest area λ_j, then the truck's waiting time is dependent on the last drone's arrival time. An energy-efficient route takes more or equal time compared to the most time-efficient route. Therefore, a drone can save energy without increasing the truck's waiting time, if it follows the energy-efficient path and still can return to the truck before the arrival of the last drone.

4 DRONE-TRUCK COOPERATION FRAMEWORK

Here we present a three-step drone-truck cooperation framework to minimize the total delivery time. The first step is the preprocessing stage, which estimates the delivery route for each customer assuming a static wind condition. The next step is executed at each drone, where the assigned delivery route is updated when time-varying dynamics affect the traveling time. The first two steps try to minimize the drone's energy consumption also. The last step considers the road traffic and drones' traveling time to adjust the truck's waiting time.

4.1 Pre-Processing

We assume that all the customers to serve are known before starting the delivery. So, before starting the actual delivery it can be estimated how many drones are required at a time and from where the drones can be launched. A drone completes a trip (known as *forward trip*) from a rest area (launching point δ_i^L) to the customer's location δ_i and a trip (known as *backward trip*) from δ_i to another rest area (rendezvous point δ_i^R). Let u_0 be a dummy source vertex added in G^T in such a way that all the rest areas are connected to u_0 with edge weight zero. Let this modified graph be G^{T*}. Now if the SSSP is computed from u_0 to δ_i, the second vertex in this shortest route provides the rest area (Let λ_u) from where a drone can be launched to serve δ_i with the least amount of time. Additionally, the path length or distance (Let d^f) provides the estimated time required for forward trip (See Algorithm 1 Line 2).

Backward trip time (Let d_y^b) to return a rest area λ_y, $(u \leq y \leq r+1)$ can be estimated by finding shortest path from δ_i to all these rest areas. Any classic SSSP algorithm can be used for this purpose. The total delivery time of a DTCDS can be minimized when the total waiting time of the truck to gather the drones is minimized. Therefore, we can choose a rest area λ_v such that the time gap between the drone's reaching time and τ_v is minimized or

$v = argmin_y(|\tau_y - (\tau_u + d^f + d^b_y)|)$. The time t^T_i, when the drone reaches λ_v can be computed by adding the forward and backward time with the launching time instance τ_u. If t^T_i is greater than τ_v, then the truck needs to wait for the drone to return and τ_v should be updated (Algorithm 1 Line 8).

Algorithm 1: PreProcess ($G^{T*}, G^E, D, \mathcal{T}$)

1 **for** $i \in 1 \dots n$ **do**
2 Compute SSSP on G^{T*} to find δ^L_i (Let λ_u) and shortest route distance d^f from λ_u to δ_i.
3 Compute shortest path distances d^b_y from δ_i to all rest area λ_y such that $u \le y \le r+1$.
4 $v \leftarrow argmin_y(|\tau_y - (\tau_u + d^f + d^b_y)|)$
5 $\delta^R_i \leftarrow \lambda_v$
6 $t^T_i \leftarrow \tau_u + d^f + d^b_v$
7 **if** $t^T_i > \tau_v$ **then**
8 $\tau_v \leftarrow t^T_i$
9 Compute t^E_i by finding shortest path from λ_u to δ_i, and shortest path from δ_i to λ_v on G^E.
10 **if** $t^E_i < \tau_v$ **then**
11 $\delta_i.mode \leftarrow E$ `// Energy efficient`
12 **else**
13 $\delta_i.mode \leftarrow T$ `// Time efficient`

Next, using any SSSP algorithm, t^E_i can be computed by fixing the launching point at λ_u and rendezvous point at λ_v in G^E. As t^E_i denotes the time when the drone reaches its rendezvous point by following the most energy-efficient route, a drone is preferred to follow the energy-efficient route unless t^E_i is greater than τ_v. Therefore, at the end of the pre-processing stage, either energy-efficient mode E or time-efficient mode T is assigned for each delivery δ_i. More specifically, the *mode* parameter determines if the energy-efficient route will be followed instead of the time-efficient route while delivering a package.

4.2 Dynamic Route Selection at Drone

It has been observed that in a dynamic network, updating a SSSP takes less time than recomputing it from scratch in case of change in network topology [10]. A dynamic shortest path update algorithm considers a set of change edges $\Delta = Ins, Del$ (Ins be the inserted edges and Del are the deleted edges) as input and finds out the affected subgraph using a tree structure known as SSSP tree. Then the algorithm updates the distance of the affected vertices using an iterative shortest path computation. The SSSP update algorithm can be used to find the updated shortest path length from a source vertex u_s to destination vertex u_d. The overview of parallel SSSP update is shown in Algorithm 2.

In our DTCDS, we consider the edge weights change in both G^T and G^E due to changes in wind characteristics. All such changes can be tracked by the wind control units distributed over the delivery area A, and the changed edge weights can be broadcasted to the drones [9]. Let Δ^T (resp. Δ^E) be the set of changes in G^T (resp. G^E).

Algorithm 2: RouteUpdate(u_s, u_d, G, Δ)

`/* Step1: Find affected vertices */`
1 **for** *each change edge* $(u, v) \in \Delta$ *in parallel* **do**
2 Find affected endpoint $x \in u, v$ such that distance of x is changed due to Δ.
3 Add x in a set of affected vertices Aff.
`/* Step2: Update distance of affected vertices and their neighbors iteratively */`
4 **while** Aff *is not empty* **do**
5 **for** *vertex* $x \in Aff$ *in parallel* **do**
6 Update distance of x by finding a path through a neighbor in updated G, such that the distance of x from u_s decreases.
7 If the distance of x or its any neighbor x_n gets an updated distance, add the vertex to Aff.
8 Return the updated shortest route from u_s to u_d.

So, the drones receive a set of changes $\Delta_{all} = \{\Delta^T, \Delta^E\}$ as input to compute the updated delivery route.

Algorithm 3: DynamicDroneRoute($\Delta_{all}, G^T, G^E, \delta_x$)

`/* Drone's current location` loc`, assigned delivery` δ_x`, and current time` t_{now} `*/`
1 **if** *new* Δ_{all} *received* **then**
2 $route^E \leftarrow$RouteUpdate $(loc, \delta^R_x, G^E, \Delta^E)$
3 $route^T \leftarrow$RouteUpdate $(loc, \delta^R_x, G^T, \Delta^T)$
4 $t^E_x \leftarrow t_{now}$+ time to travel $route^E$
5 $t^T_x \leftarrow t_{now}$+ time to travel $route^T$
`/* Let` $\delta^R_x = \lambda_v$ `*/`
6 **if** $t^E_x > \tau_v$ **then**
7 **if** $t^T_x \le \tau_v$ **then**
8 $\delta_x.mode \leftarrow T$ `// Time efficient`
9 **else**
10 Send (λ_v, t^T_x) to the Truck

A drone serving δ_x can use Algorithm 2 on G^E and G^T to find the updated rendezvous time t^E_x and t^T_x respectively. Let the rendezvous point be λ_v for the drone. We aim to minimize the energy usage of a drone whenever possible without increasing the truck's waiting time. Therefore, an energy-efficient mode is the natural choice unless specified otherwise by the pre-processing step or t^E_x becomes greater than τ_v. In case t^E_x and t^T_x both become greater than τ_v, the drone uses time-efficient route and informs the truck about the updated rendezvous time t^T_x. The drone sends a tuple consisting of the rendezvous point and updated rendezvous time to the truck (See Algorithm 3).

4.3 Computation at Truck

The truck's trip can be affected by (i) the road traffic, and (ii) the wait time to gather the drones. When a truck is on ρ_j, but unable to reach the next rest area λ_{j+1} at the pre-estimated time τ_{j+1}, then the

Algorithm 4: ComputeAtTruck($\mathcal{T}, \Lambda$)

```
/* Truck's current location loc, and current time
   t_now                                          */
if loc lies on ρ_j and t_now > τ_{j+1} then
    Estimate τ_{j+1} using traffic movement and road distance
      to cover.
    Update all τ_x for j < x < r + 1 in T.
    Broadcast T to all drones.
if a message (λ_j, t) received from a drone then
    if τ_j < t then
        τ_j ← t
        Update all τ_x for j < x < r + 1 in T.
        Broadcast T to all drones.
```

truck recomputes the new estimated τ_{j+1} considering the current traffic and distance to cover. An update on τ_{j+1} requires update on all next elements $\tau_x \in \mathcal{T}, (j < x < r + 1)$. This step can be done in parallel threads in constant time and broadcasted to all the drones. When a drone fails to return to its rendezvous point λ_v on or before τ_v, it informs the truck about the new rendezvous time. The information exchange is one-to-one and consists of a tuple only. The truck updates τ_v, all next entries in $\mathcal{T}$ and broadcasts $\mathcal{T}$ (Algorithm 4).

Complexity Analysis: Let G^T or G^E have $\mathcal{V}$ vertices and $\mathcal{E}$ edges. Then, the shortest paths during the pre-processing step can be computed in $O(\mathcal{V}+\mathcal{E}\log(\mathcal{V}))$ time by using a Dijkstra's shortest path algorithm. The *argmin* computation (Line 4 in Algorithm 1) requires $O(r)$ time. Therefore, for n deliveries, the pre-processing step takes total $O(n(\mathcal{V} + \mathcal{E}\log(\mathcal{V}) + r))$ time.

The dynamic route selection performed by the drone uses an existing shortest path update algorithm [10] which takes time $O(|\Delta|/p) + O(\mathcal{D}\chi\varrho/p + \mathcal{D})$. Here, p is the number of processing units (or available threads) in a drone, χ is the number of affected vertices at each iteration in Step 2 of Algorithm 2, ϱ and $\mathcal{D}$ are the average degree and diameter of the graph. Computation at the truck level mainly involves updating $\mathcal{T}$ in case of varying traffic or change in a drone's rendezvous time: thus $O(|\Lambda|)$.

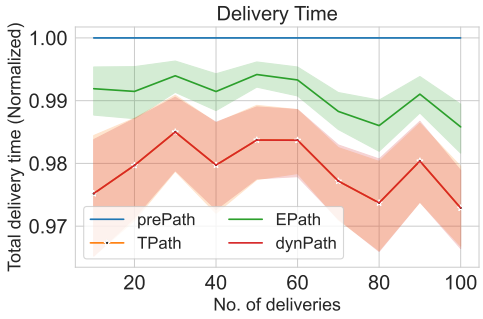
Communication: Algorithm 3 involves only point to point communication between a drone and the truck and the cost for each communication is $O(l + \mu/B)$, where l is latency, μ is the message size and B is the bandwidth. As the drone sends only a tuple with two values μ is very less and the communication time becomes insignificant. The truck may need to broadcast the updated $\mathcal{T}$ to all the drones and it requires $O((l + \mu/B)log(m + 1))$ time.

5 EXPERIMENTAL EVALUATION

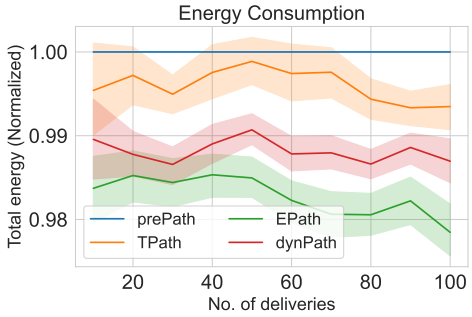
To implement a realistic delivery map, we use TataNld network [12] and generate a bi-directional weighted graph using the provided latitude and longitude of the nodes. Then we scale down the edge weight i.e., the distance such that any delivery trip for a drone becomes 5 Kilometre or less. A few fixed vertices are considered as rest areas and others are considered as potential customers' locations. A drone flying model and varying wind model are implemented as mentioned in [22].

5.1 Delivery Time and Energy Consumption

In our first experiment, we increase the total number of deliveries from 10 to 100 and observe the total delivery time and energy consumption by the drones. We repeat each experiment 50 times and select the customers' locations randomly. We consider four delivery models, which are *1) prePath:* Drone follows the delivery path pre-estimated at pre-processing stage considering a fixed wind condition, *2) TPath:* All drones follow the most time-efficient delivery path under varying wind, *3) EPath:* All drones follow the most energy-efficient delivery path under varying wind, *4) dynPath:* Drones follow our proposed algorithm to change their delivery mode (Energy- and Time-efficient) under varying wind condition.



(a) Total Delivery Time. TPath and dynPath values overlapped



(b) Energy Consumption by Drones.

Figure 3: Performance analysis of DTCDS. (a) Total delivery time, (b) Total energy consumption by the drones. Ratios with respect to prePath values are shown along the Y-axis.

Figure 3a shows the total delivery time for all delivery models. We divide the total delivery time in each model by the *prePath* model delivery time to get normalized results. Overlapping of *TPath* and *dynPath* delivery times in the plot indicates that the *dynPath* model takes the minimum delivery time similar to the *TPath* delivery time.

Figure 3b shows that the energy consumption of *dynPath* model is higher than the most energy efficient model i.e., *EPath*. However, the *dynPath* model uses much less energy than the *TPath* model to complete the deliveries. Therefore, the empirical results show that the *dynPath* model satisfies our objective of minimizing the total delivery time while reducing the energy consumption.

5.2 Experiment on Delivery Modes

As a delivery can be completed by following either time efficient mode (Let T-mode) or energy efficient mode (Let E-mode), in this

experiment we observe the total number of deliveries completed in each of these modes. The *TPath* and *EPath* model accomplish all the deliveries in T-mode and E-mode respectively. However, for a set of deliveries the *prePath* and *dynPath* models may use both T-mode and E-mode. The *prePath* model decides the mode of delivery at preprocessing stage and does not change the mode under varying wind conditions. On the other hand, the *dynPath* model can change a delivery mode at the time of actual delivery.

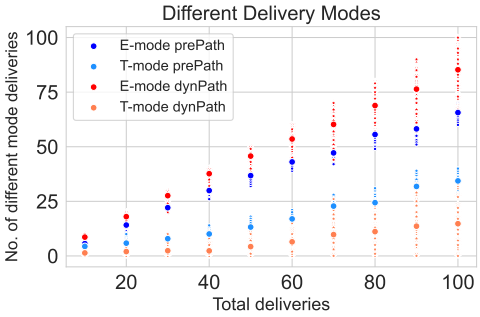


Figure 4: Deliveries performed in different modes.

In Figure 4, along Y-axis we vary the number of total deliveries and along X-axis we present the number of deliveries following each delivery mode. The plot shows that compared to *prePath* model, *dynPath* uses more E-mode deliveries, which means *dynPath* model saves more energy to complete the same number of total deliveries. It can be noticed that in some cases the number of deliveries following T-mode becomes zero in *dynPath* model. It may occur due to two reasons: 1) when there is no alternative route to complete the delivery, 2) when the delivery time by following the energy efficient route and the time efficient route becomes the same.

6 CONCLUSIONS

In this paper, we devise a DTCDS, where each drone and truck update their route independently under time-varying dynamics and each drone synchronizes with the truck by short message passing. We present a greedy three-step drone-truck cooperation framework that minimizes the total delivery time. Our algorithm also reduces the energy consumption of drones as a secondary objective.

Currently, the truck always waits for the drones to return. In future work, it would be interesting to assume the truck waits for a fixed duration at each rest area and the drone meets to some next rest area if it cannot reach the pre-estimated rendezvous point on time. This problem can be solved using an approximation algorithm if a drone's energy budget is fixed.

7 ACKNOWLEDGMENTS

This work was partially supported by the NSF projects SANDY (Award # OAC-1725755) and CANDY (Award # OAC-2104078), and by a grant from the Intelligent Systems Center (ISC) at the Missouri University of Science and Technology, Rolla, USA.

REFERENCES

[1] Francesco Betti Sorbelli, Cristina M. Pinotti, Simone Silvestri, and Sajal K. Das. 2020. Measurement Errors in Range-based Localization Algorithms for UAVs: Analysis and Experimentation. *IEEE Trans. on Mobile Computing* (2020), 1–1.

[2] Igor Bisio, Chiara Garibotto, Fabio Lavagetto, et al. 2018. Blind detection: Advanced techniques for WiFi-based drone surveillance. *IEEE Trans. on Vehicular Technology* 68, 1 (2018), 938–946.

[3] Nils Boysen, Dirk Briskorn, Stefan Fedtke, and Stefan Schwerdfeger. 2018. Drone delivery from trucks: Drone scheduling for given truck routes. *Networks* 72, 4 (2018), 506–527.

[4] Federico Busato and Nicola Bombieri. 2015. An efficient implementation of the Bellman-Ford algorithm for Kepler GPU architectures. *IEEE Trans. Parallel and Distributed Systems* 27, 8 (2015), 2222–2233.

[5] Tiziana Calamoneri, Federico Corò, and Simona Mancini. 2022. A Realistic Model to Support Rescue Operations After an Earthquake via UAVs. *IEEE Access* 10 (2022), 6109–6125.

[6] Gloria Cerasela Crişan and Elena Nechita. 2019. On a cooperative truck-and-drone delivery system. *Procedia Computer Science* 159 (2019), 38–47.

[7] Rami Daknama and Elisabeth Kraus. 2017. Vehicle Routing with Drones. *CoRR* abs/1705.06431 (2017). arXiv:1705.06431 http://arxiv.org/abs/1705.06431

[8] Anurag Ingole and Rupesh Nasre. 2015. Dynamic shortest paths using javascript on *GPUs*. In *IEEE 22nd Int Conf on High-Performance Computing (HiPC)*. 1–5.

[9] Arindam Khanda, Federico Coro, Francesco Betti Sorbelli, Cristina M Pinotti, and Sajal K Das. 2021. Efficient route selection for drone-based delivery under time-varying dynamics. In *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, 437–445.

[10] Arindam Khanda, Sriram Srinivasan, Sanjukta Bhowmick, Boyana Norris, and Sajal K Das. 2021. A Parallel Algorithm Template for Updating Single-Source Shortest Paths in Large-Scale Dynamic Networks. *IEEE Transactions on Parallel and Distributed Systems* (2021).

[11] Aakash Khochare, Yogesh Simmhan, Francesco Betti Sorbelli, and Sajal K. Das. 2021. Heuristic Algorithms for Co-scheduling of Edge Analytics and Routes for UAV Fleet Missions. In *40th IEEE Conference on Computer Communications, INFOCOM 2021, Vancouver, BC, Canada, May 10-13, 2021*. IEEE, 1–10.

[12] S. Knight, H.X. Nguyen, N. Falkner, R. Bowden, and M. Roughan. 2011. The Internet Topology Zoo. *Selected Areas in Communications, IEEE Journal on* 29, 9 (october 2011), 1765 –1775. https://doi.org/10.1109/JSAC.2011.111002

[13] Chase C Murray and Amanda G Chu. 2015. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies* 54 (2015), 86–109.

[14] Chase C Murray and Ritwik Raj. 2020. The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transportation Research Part C: Emerging Technologies* 110 (2020), 368–398.

[15] Deepak Murugan, Akanksha Garg, and Dharmendra Singh. 2017. Development of an adaptive approach for precision agriculture monitoring with drone and satellite data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10, 12 (2017), 5322–5328.

[16] Akif Rehman, Masab Ahmad, and Omer Khan. 2020. Exploring accelerator and parallel graph algorithmic choices for temporal graphs. In *11th Int. Workshop on Programming Models and Applications for Multicores and Manycores*. 1–10.

[17] Roberto Roberti and Mario Ruthmair. 2021. Exact methods for the traveling salesman problem with drone. *Transportation Science* 55, 2 (2021), 315–335.

[18] Suttinee Sawadsitang, Dusit Niyato, Puay Siew Tan, Ping Wang, and Sarana Nutanong. 2019. Multi-Objective Optimization for Drone Delivery. In *IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. 1–5.

[19] Weisen Shi, Haibo Zhou, Junling Li, Wenchao Xu, Ning Zhang, and Xuemin Shen. 2018. Drone assisted vehicular networks: Architecture, challenges and opportunities. *IEEE Network* 32, 3 (2018), 130–137.

[20] Francesco Betti Sorbelli, Federico Corò, Sajal K. Das, Lorenzo Palazzetti, and Cristina M. Pinotti. 2022. Greedy Algorithms for Scheduling Package Delivery with Multiple Drones. In *23rd International Conference on Distributed Computing and Networking (ICDCN)*. ACM, 31–39.

[21] Francesco Betti Sorbelli, Federico Corò, Sajal K. Das, and Cristina M. Pinotti. 2021. Energy-Constrained Delivery of Goods With Drones Under Varying Wind Conditions. *IEEE Trans. Intell. Transp. Syst.* 22, 9 (2021), 6048–6060.

[22] Francesco Betti Sorbelli, Federico Corò, Sajal K. Das, and Cristina M. Pinotti. 2021. Energy-Constrained Delivery of Goods With Drones Under Varying Wind Conditions. *IEEE Trans. Intell. Transp. Syst.* 22, 9 (2021), 6048–6060.

[23] Sriram Srinivasan, Sara Riazi, Boyana Norris, Sajal K Das, and Sanjukta Bhowmick. 2018. A Shared-Memory Parallel Algorithm for Updating Single-Source Shortest Paths in Large Dynamic Networks. In *25th IEEE Int. Conf. on High Performance Computing (HiPC)*. 245–254.

[24] Amila Thibbotuwawa, Grzegorz Bocewicz, Peter Nielsen, and Banaszak Zbigniew. 2019. Planning deliveries with UAV routing under weather forecast and energy consumption constraints. *IFAC-PapersOnLine* 52, 13 (2019), 820–825.

[25] Desheng Wang, Peng Hu, Jingxuan Du, Pan Zhou, Tianping Deng, and Menglan Hu. 2019. Routing and scheduling for hybrid truck-drone collaborative parcel delivery with independent and truck-carried drones. *IEEE Internet of Things Journal* 6, 6 (2019), 10483–10495.

[26] Yangzihao Wang, Andrew Davidson, Yuechao Pan, Yuduo Wu, Andy Riffel, and John D Owens. 2016. Gunrock: A high-performance graph processing library on the GPU. In *21st Symp. on Principles and Practice of Parallel Programming*. 1–12.