ENERGY-EFFICIENT ROUTE PLANNING FOR AUTONOMOUS AERIAL VEHICLES BASED ON GRAPH SIGNAL RECOVERY

Tianxi Ji[1], *Siheng Chen*[1], *Rohan Varma*[1], *Jelena Kovačević*[1,2]

[1]Dept. of ECE, [2]Dept. of BME ,
Carnegie Mellon University,
Pittsburgh, PA, USA

ABSTRACT

We use graph signal sampling and recovery techniques to plan routes for autonomous aerial vehicles. We propose a novel method that plans an energy-efficient flight trajectory by considering the influence of wind. We model the weather stations as nodes on a graph and model wind velocity at each station as a graph signal. We observe that the wind velocities at two close stations are similar, that is, the graph signal of wind velocities is smooth. By taking advantages of the smoothness, we only query a small fraction of it and recover the rest by using a novel graph signal recovery algorithm, which solves an optimization problem. To validate the effectiveness of the proposed method, we first demonstrate the necessity to take wind into account when planning route for autonomous aerial vehicles, and then show that the proposed method produces a reliable and energy-efficient route.

Index Terms— Graph signal processing, sampling and recovery, route planning, autonomous vehicle

1. INTRODUCTION

As a generalization of classical discrete signal processing, signal processing on graphs is an effective tool to analyze arbitrary signals residing on irregular, complex structures. The framework of signal processing on graphs models the underlying structures as graphs and the supported signals as graph signals, and the basic concepts such as filters, convolution, z-transform, Fourier transform, frequency components are also generalized to have their corresponding counterparts in this framework [1, 2].

One of the most fundamental problems in signal processing is sampling and recovery. Some recent works study graph signal recovery based on either smoothness [3] or a bandlimited assumptions [4, 5]. For example, in [3], the authors formulate the recovery task as an optimization problem; in [6], the authors show that perfect recovery is possible for bandlimited graph signals; in [7], the authors relax the constraint of bandlimited graph signal, propose approximately bandlimited graph signal, and evaluate the performance of recovery strategies based on random and experimentally designed sampling on two types of graph. In this paper, we study an application of sampling and recovery of graph signals.

With the rise of autonomous vehicles, a lot of approaches have been proposed for the task of route planning. Previous applications for route planning involve planning optimal route for electric vehicles with limited battery storage capacity [8], real-time route planning for autonomous aerial vehicles to handle unforeseeable changes of environments [9], and planning path to avoid obstacles for underwater vehicles [10]. A widely used route planning algorithm is Dijkstra's algorithm, which uses the distance between each pair of accessible nodes to plan a shortest path that minimize the travel cost. Dijkstra's algorithm requires the accessibility of data recorded at all nodes, which makes it vulnerable to the inaccessibility or loss of data. This is common in many problems when dealing with data of huge volume and complex structure.

In this paper, we propose a framework to plan energy-efficient route for autonomous aerial vehicles, such as the Amazon octocopter drone. As far as we know, this is the first work to plan routes for aerial vehicles based on the wind information. Instead of using the data recorded at all positions, we only query a small fraction of data at certain positions. The data is then interpolated using graph signal recovery techniques. In the the proposed framework, we also propose a series of procedure to compute the energy consumption between each pair of accessible nodes as the input of Dijkstra's algorithm. We show the necessity of taking wind into consideration for route planning, validate the framework on real wind dataset and show that it produces a reliable and energy-efficient route with high effectiveness.

2. BACKGROUND

In this section we briefly review the basic concept of signal processing on graphs. For more details, see [2, 1].

The authors gratefully acknowledge support from the NSF through awards 1130616,1421919, the University Transportation Center grant (DTRT12-GUTC11) from the US Department of Transportation, and the CMU Carnegie Institute of Technology Infrastructure Award. This paper is a project extension from course 18-790 at Carnegie Mellon University, the authors gratefully acknowledge the contribution from teammates George Lederman and Khoi Nguyen.

Signal processing on graphs studies signals with complex, irregular structure represented by a graph $G = (\mathcal{V}, \mathrm{A})$, where $\mathcal{V} = \{v_0, \ldots, v_{N-1}\}$ is the set of nodes and $\mathrm{A} \in \mathbb{R}^{N \times N}$ is a weighted adjacency matrix. It represents the connections of the graph G. In this paper, we restrict ourselves to undirected graphs and consequently symmetric graph shifts.

Graph Signal. A *graph signal* is defined as the map on the graph nodes that assigns the signal coefficient $x_n \in \mathbb{R}$ to the node v_n. Once the node order is fixed, the graph signal can be written as a vector

$$\mathbf{x} = \left[x_0, x_1, \ldots, x_{N-1}\right]^T \in \mathbb{R}^N.$$

Graph Fourier Transform. In general, a Fourier transform corresponds to the expansion of a signal using basis functions that are invariant to filtering; the basis here is the eigenbasis of a graph representation matrix B. A graph representation matrix can be either the weighted adjacency matrix, or other versions normalized by degrees. Some common normalized versions include $\mathrm{D} - \mathrm{A}$, $\mathrm{D}^{-\frac{1}{2}}\,\mathrm{A}\,\mathrm{D}^{-\frac{1}{2}}$ and $\mathrm{A}\,\mathrm{D}^{-1}$, where D is a degree matrix with $\mathrm{D}_{i,i} = \sum_j \mathrm{A}_{i,j}$. The spectral decomposition of B is [11]

$$\mathrm{B} = \mathrm{V}\,\Lambda\,\mathrm{V}^{-1},$$

where the eigenvectors of B form the graph Fourier basis V (the norm of each column is normalized to one), the graph Fourier transform matrix $\mathrm{U} = \mathrm{V}^{-1}$, and $\Lambda \in \mathbb{R}^{N \times N}$ is the diagonal matrix of corresponding eigenvalues $\lambda_0, \ldots, \lambda_{N-1}$ of B. These eigenvalues represent frequencies on the graph [12]. The *graph Fourier transform* of $\mathbf{x} \in \mathbb{R}^N$ is $\widehat{\mathbf{x}} = \mathrm{U}\,\mathbf{x}$. The *inverse graph Fourier transform* is $\mathbf{x} = \mathrm{V}\,\widehat{\mathbf{x}}$. The vector $\widehat{\mathbf{x}}$ represents the signal's expansion in the eigenvector basis and describes the frequency content of the graph signal $\mathbf{x}$. Various graph representation matrices lead to different graph Fourier bases, which model different types of graph signals. The choice depends on specific applications.

3. PROPOSED FRAMEWORK FOR ENERGY-EFFICIENT ROUTE PLANNING

Previous methods for energy-efficient route planning are vulnerable to the inaccessibility or loss of data on which they dependent. To alleviate this, we propose a framework that only queries a small part of the entire dataset, and recovers the rest with a novel graph signal recovery algorithm.

3.1. Proposed Framework

We consider a graph representing the network of road. The intersections are represented as nodes on the graph, where edge exists between adjacent nodes if the corresponding intersections are connected by a road. We assume there is a wind monitor station at each intersection that records wind velocity, since each station measures the local wind velocity, the wind velocities of all the stations form a graph signal.

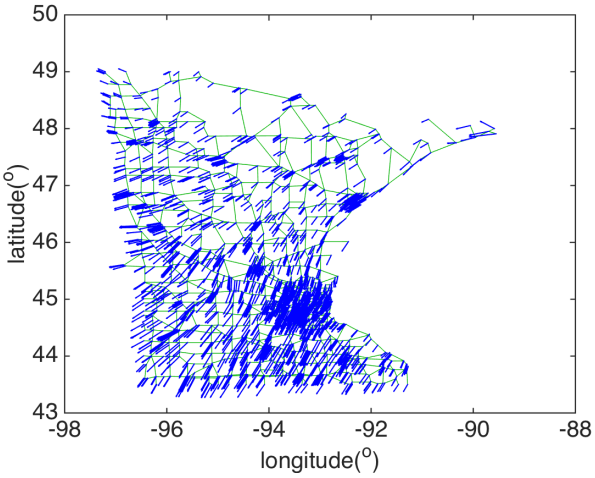


Fig. 1: A snapshot of wind velocities at 2642 intersections of Minnesota road network. The green edges show the outline of the road network, and the blue arrow denotes the wind velocity residing on each intersection.

Figure 1 shows a snapshot of wind velocities we used in experiment. We see that the wind velocities at two close stations are similar, that is, the graph signal of wind velocities is smooth. By taking advantages of the smoothness, we use graph sampling techniques to query wind velocities at a small fraction of positions, instead of using the wind velocities recorded at all the positions. The other wind velocities are then interpolated using graph signal recovery techniques. We then use the recovered wind velocities to plan a route. Since we only query the wind velocities from a few positions, our proposed method is efficient.

The proposed method is shown in Figure 2. In the sampling block, we consider three sampling strategies to query the wind velocities at certain positions; in the recovery block, we recover the wind velocities at the rest of the positions by using the proposed recovery algorithm; and in the wind velocity based route planning block, we perform the Dijkstra's algorithm on the recovered wind velocities to plan an energy-efficient route.

3.2. Signal Sampling and Recovery on Graph

We query the wind velocities from M stations (or intersections), that is, we sample M coefficients from a graph signal $\mathbf{x} \in \mathbb{R}^N$ to produce a sampled graph signal $\mathbf{x}_{\mathcal{M}} \in \mathbb{R}^M$ ($M < N$), where $\mathcal{M} = (\mathcal{M}_0, \cdots, \mathcal{M}_{M-1})$ denotes the sequence of *sampled* weather stations, and $\mathcal{M}_i \in \{0, 1, \cdots, N-1\}$. We then recover the wind velocities for the rest stations, that is, we recover $\mathbf{x}_{\mathcal{M}}$ to get $\mathbf{x}' \in \mathbb{R}^N$. The sampling operator $\Psi \in \mathbb{R}^{M \times N}$ is a linear mapping from $\mathbb{R}^N$ to $\mathbb{R}^M$, defined as

$$\Psi_{i,j} = \begin{cases} 1, & j = \mathcal{M}_i; \\ 0, & \text{otherwise}, \end{cases}$$

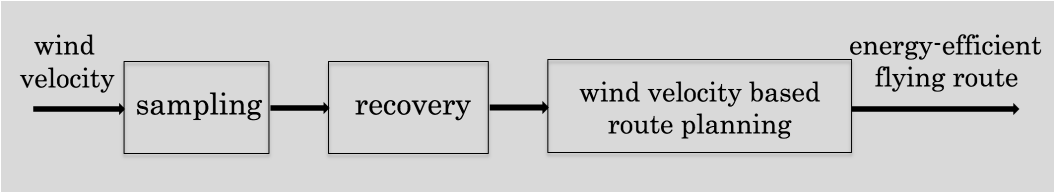


Fig. 2: Diagram for the proposed framework.

and the recovery operator $\Phi \in \mathbb{R}^{N\times M}$ is a mapping from $\mathbb{R}^M$ to $\mathbb{R}^N$. $\mathbf{y} = \Psi\mathbf{x} + \mathbf{e} \in \mathbb{R}^M$ are the sampled wind velocities, where $\mathbf{e}$ is noise, and $\mathbf{x}' = \Phi\mathbf{y} \in \mathbb{R}^N$ recovers $\mathbf{x}$ either exactly or approximately.

In this paper, we consider two sampling strategies: random sampling means that sample indices are chosen from the node set independently and randomly; experimentally design sampling means that sample indices can be chosen beforehand. For experimentally design sampling, we also consider two approaches, including a stochastic one and a deterministic one. The main idea of the design is to take advantages of the smoothness of a graph signal. When two weather stations are close to each other, we expect the corresponding wind velocities to be similar, and hence the graph signal of wind velocities is smooth. Further, we assume that the smooth graph signals are bandlimited, that is, we approximate a graph signal by the first $K \ll N$ graph Fourier basis vectors [6].

For the stochastic approach, we choose a node according to some sampling distribution $\{\pi_i\}_{i=1}^N$, that is, the ith node has its own probability π_i to be chosen depending on the graph structure. As suggested by [7], we consider the sampling distribution be the square root of leverage scores, defined as $\pi_i = \|\mathbf{u}_i\|_2 / \sum_{j=0}^{N-1} \|\mathbf{u}_j\|_2$, where $\mathbf{u}_i$ is the ith column of the first K rows of graph Fourier transform matrix, $\mathrm{U}_{(K)}$. This sampling distribution with a simple recovery technique provides an unbiased estimator of the first K frequency components of a graph signal and optimizes the recovery error [7]. We call this approach experimentally designed sampling based on the sampling distribution. For the deterministic approach, we consider an optimal sampling operator from sampling theory of graph signals, which optimizes the minmax recovery error and is implemented in a greedy manner [6]. We call this approach experimentally designed sampling based on the optimal sampling operator. The advantages of the sampling distribution approach are that it is computationally cheap and works for a larger class of graph signals; the advantages of the optimal sampling operator approach are that its sample set is designed only once and it usually has a better empirical performance especially when the sample size is small.

We then recover the wind velocities at the rest of the positions by using the least square:

$$\begin{aligned}\mathbf{x}' &= \Phi\mathbf{y} = \mathrm{V}_{(K)}\,\widehat{\mathbf{x}}^*_{(K)} \\ &= \mathrm{V}_{(K)} \arg\min_{\widehat{\mathbf{x}}_{(K)}\in\mathbb{R}^K} \left\|\mathrm{D}\,\Psi\,\mathrm{V}_{(K)}\,\widehat{\mathbf{x}}_{(K)} - \mathrm{D}\,\mathbf{y}\right\|_2^2 \\ &= \mathrm{V}_{(K)}(\mathrm{D}\,\Psi\,\mathrm{V}_{(K)})^\dagger\,\mathrm{D}\,\mathbf{y}, \qquad (1)\end{aligned}$$

where $\mathrm{D} \in \mathbb{R}^{M\times M}$ is a diagonal weight matrix, $\mathrm{V}_{(K)} \in \mathbb{R}^{N\times K}$ denotes the first K columns of V, and $\widehat{\mathbf{x}}_{(K)}$ is the estimated bandlimited signal with bandwidth K. For random sampling and experimentally designed sampling based on the sampling distribution, $\mathrm{D}_{i,i} = 1/\sqrt{|\mathcal{M}|\pi_j}$ when $\Psi_{i,j} = 1$; for experimentally designed sampling based on the optimal sampling operator, $\mathrm{D}_{i,i} = 1$. The weight matrix D is introduced to compensate for the non uniformity of sampling. When it is an identity matrix, (1) is exactly the recovery operator proposed in [6]. Note that (1) assumes that the graph signal is a bandlimited signal and only recovers the low frequency content. This bandlimited assumption is not completely accurate in general, however, our goal is to plan a route, instead of exactly recovering the graph signal. In the experiments, we generate accurate routes based on this bandlimited approximation.

3.3. Wind Velocity based Route Planning

To use the Dijkstra's algorithm, we propose a series of procedures to compute the energy consumption. First, we assume that the vehicle only flies along the road between the intersections. Since a $200ft$ slab of air, located between $200ft$ and $400ft$ from the ground was proposed by Amazon aeronautics experts to be a high speed transit airspace[1], which is beneath the height of many skyscrapers in urban area, thus, to diminish the influence of drone to civilian life and to make easy recycle after accidental falling, the assumption is reasonable.

Based on this assumption, the flying route of the drone is modeled as a trajectory consisting of different segments that connect the nodes as it passes by. We further assume the drone flies along each segment at a constant velocity v_0. To maintain that velocity, its engine should provide speed less than v_0 when it flies following the wind, and greater than v_0 to compensate when it has to fly through a head-wind area. Since the wind differs at each interval, the drone will adjust its speed

[1] Refer to Amazon's letter to the U.S. Federal Aviation Administration on July 9, 2014, via `http://z-ecx.images-amazon.com/images/G/01/rowland/AmazonPetitionforExemption_July92014.pdf`

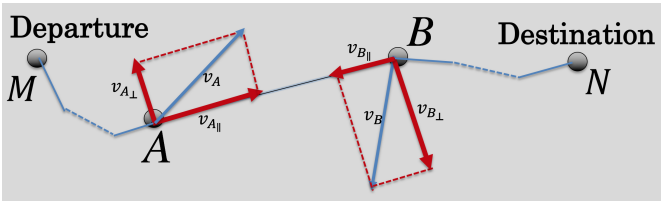


Fig. 3: Illustration of an interval A, B with wind velocity v_A and v_B.

to adapt to the wind. For simplicity, we consider the energy consumption as the power to provide adaptive speed during each interval to maintain the constant velocity. As the drone flies from the departure node to the destination node, for all pairs of accessible nodes, we compute its energy consumption at each interval as the input for the Dijkstra's algorithm.

Consider a scenario where the drone departs from node M to node N, one possible interval of the trip is the segment between node A and B, as in Figure 3. To satisfy the assumption that the vehicle flies along the segment at a constant velocity v_0, the resultant velocity that the engine should provide during interval A, B denoting as $v_{A,B}$ is obtained as following:

$$\begin{aligned} v_{A,B_\parallel} &= v_0 - (v_{A_\parallel} + v_{B_\parallel}), \\ v_{A,B_\perp} &= -(v_{A_\perp} + v_{B_\perp}), \\ v_{A,B} &= \sqrt{{v_{A,B_\parallel}}^2 + {v_{A,B_\perp}}^2}, \end{aligned}$$

where $v_{A_\parallel}$ and $v_{A_\perp}$ are the tangential and normal velocity at node A, $v_{A,B_\parallel}$ and $v_{A,B_\perp}$ are the tangential and normal component of $v_{A,B}$.

Then the *interval energy consumption* from node A to B, denoted as $E_{A,B}$ is obtained as:

$$E_{A,B} = C\frac{|d_{A,B}|}{v_0}\,|v_{A,B}|,$$

where C is a scalar that parameterizes the energy consumption for the engine to provide an unit amount velocity at that certain interval. Mathematically, $E_{A,B}$ is determined by the adaptive speed $v_{A,B}$ during the interval A, B and the flying time to cover that distance with the constant flying velocity v_0.

Given the energy consumption at each accessible interval, the energy-efficient route from M to N is obtained as:

$$\{M, L^*, \cdots, K^*, N\} = \arg\min_{L,\cdots,K} E_{M,L} + \cdots + E_{K,N}. \quad (2)$$

Equation (2) is solved by the Dijkstra's algorithm and the overall energy consumption for the trajectory is the sum of all the *interval energy consumption* obtained from (2).

4. EXPERIMENTS

In this section, we validate the proposed framework on a real wind velocity dataset.

We collect a dataset recording the wind velocities at 2642 intersections of the Minnesota road. In this paper, we present the data on January 1st, 2015[2]. Figure 1 shows a snapshot of the wind distribution on the entire Minnesota road. We model the Minnesota road network as a graph whose vertices represent intersections and edges represent the roads [2]. We use the metric as shown in [1] to assign weight to each edge:

$$\mathrm{A}_{n,m} = \frac{e^{-d_{nm}^2}}{\sqrt{\sum_{k\in\mathcal{N}_n} e^{-d_{nk}^2} \sum_{l\in\mathcal{N}_m} e^{-d_{ml}^2}}},$$

where $\mathrm{A}_{n,m}$ is the weight for an edge between the nth node and the mth node, d_{nm} is the distance between the nth and the mth intersection, and $\mathcal{N}_n$ is the set of neighborhood of the nth intersection. We report a series of experiments to answer the following question:

Q1. **Necessity**: Should we consider wind when planning the flying route? Here we consider two factors:
a. Various wind velocities;
b. Various flying distances.

Q2. **Effectiveness**: When wind do influence route planning of the drone, can we query the wind velocities at a small fraction of positions and still plan a route that is comparable with the route planned based on full knowledge of the wind velocities?

4.1. Q1 - Necessity

To elaborate the experiment results, we first show the list of key notations used in the experiments in Table 1. Note that the route planned based on the full knowledge of wind data, denoted as R_{truth}, is the optimal route in the spirit of minimizing energy consumption, which is regarded as the ground truth, and the route planned without wind, denoted as R_{dist}, is the baseline during experiments. The constant velocity for the Amazon octocopter drone is chosen to be 100km/h in the experiments[3].

a. Wind Velocities. To study the influence of different wind velocities on flying route planning, we define the concept of Percentage of Disagreement to be the percentage of occurrence where R_{dist} and R_{truth} does not agree among a certain number of trials. Mathematically, it follows:

$$\text{Percentage of Disagreement} = \frac{\#\{R_{\text{dist}} \neq R_{\text{truth}}\}}{\#\text{trials}}.$$

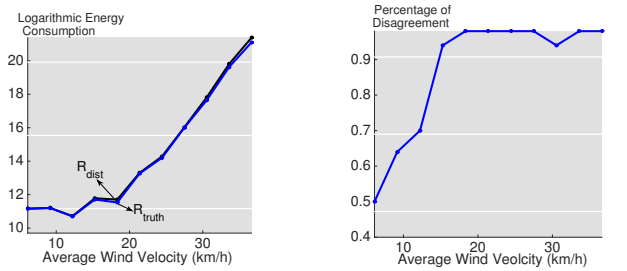
We compare the energy consumption between R_{dist} and R_{truth} among randomly picked pair of start location and destination with varying wind velocity. For each certain wind

[2]Data is collected from "Powered by Forecast", via `http://forecast.io` and is available on `http://jelena.ece.cmu.edu/research/graphs/index.html`.

[3]About 60 knots, a common velocity of an operating drone for commercial purpose, see, for example `http://www.theguardian.com/technology/2015/jul/28/amazon-autonomous-drones-only-airspace-package-delivery`

Symbol	Description
Ψ_{rnd}	sample randomly
Ψ_{stoc}	experimentally design sampling based on sampling distribution
Ψ_{opt}	experimentally design sampling based on optimal sampling operator
R_{dist}	route planned without wind
R_{truth}	route planned with full knowledge of wind
R_{rnd}	route planned based on Ψ_{rnd}
R_{stoc}	route planned based on Ψ_{stoc}
R_{opt}	route planned based on Ψ_{opt}

Table 1: Key notations used in experiments.



(a) Energy consumption (b) Percentage of disagreement

Fig. 4: Wind influences flying route planning, especially when wind velocity is high. In (a), R_{dist} (black dotted line) compared with R_{truth} (blue dotted line).This particular comparison is chosen to have more energy consumption with increasing wind velocity, and R_{truth} consumes less energy under any wind velocity; in (b) with increasing wind velocity, the route planned based with and without wind wind will disagree more.

velocity, the comparison is averaged over 50 chosen pairs. We also record the Percentage of Disagreement with various wind velocities.

Figure 4(a) shows the logarithmic energy consumption of R_{dist} (black dotted line) and R_{truth} (blue dotted line) in y axis versus different wind velocities in x axis. We scale the wind velocity to fit in the range from light air (close to 5km/h) to fresh breeze (close to 35km/h) that a flying drone may encounter. Although the energy consumption for R_{dist} and R_{truth} are close in Figure 4(a), the energy based on R_{truth} always consumes less. Figure 4(b) shows that when the wind velocity is above gentle breeze (about 15km/h), R_{truth} differs with R_{dist} with high probability, which means we should take wind into account for most wind velocities.

b. Flying Distance. We fix the wind velocity to be moderate breeze (about 25km/h), and vary the average distance between the start position and the destination from 200km to 600km to study the influence of wind across large area. For each average distance, we randomly pick several pairs of positions and compute the average difference of energy consumption between R_{dist} and R_{truth}, Figure 5 shows that the logarithmic energy consumption difference increases as the average distance increases. It means that the longer distance between two positions, the more significant influence wind will impose on energy consumption. We should take wind into consideration for a long distance route. In summary, wind velocity influences the flying route planning, especially when the wind velocity is fast and the flying distance is long.

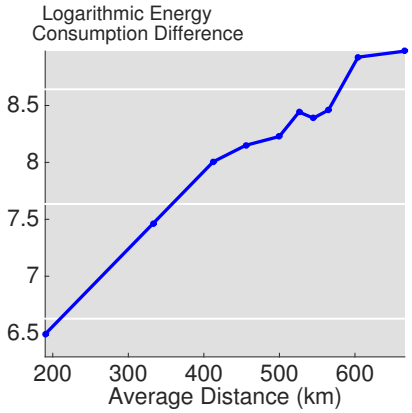


Fig. 5: Logarithmic of energy consumption difference between R_{truth} and R_{dist} versus increasing average distance. The performance is averaged over 20 trials. Wind influences flying route planning, especially when flying distance is long.

4.2. Q2 - Effectiveness

The two scenarios demonstrated in Section 4.1 indicate that it is necessary to consider wind for drone route planning. Since the task of the proposed framework is to plan a energy-efficient route, not to recover the wind velocity with high reliability, then we only need to query a very small fraction of the entire wind data such that it is sufficient to generate reliable energy-efficient route.

Experimental Setup. As shown in Figure 2, we first sample the wind using the three sampling strategies stated in Section 3.2, and then recover the data via solving an optimization problem in (1). After computing the energy consumption at each pair of accessible nodes from Ψ_{rnd}, Ψ_{stoc} and Ψ_{opt}, we use the Dijkstra's algorithm to plan the corresponding R_{rnd}, R_{stoc} and R_{opt}.

Results. Prior to the demonstration of planned routes, we show how the energy consumption of the planned routes for a certain pair of start position and destination (from St. Vincent to Owatonna in this case) change versus increasing sample size in Figure 6. For each sampling strategies, the wind velocity is recovered only based on the first 20 frequency components, $\widehat{\mathbf{x}}_{(20)}$. We observe that with just 20 queries for Ψ_{opt}, R_{opt} outperforms R_{dist} in terms of energy consumption and is very close to R_{truth}, and with slightly more queries, about 40 sam-

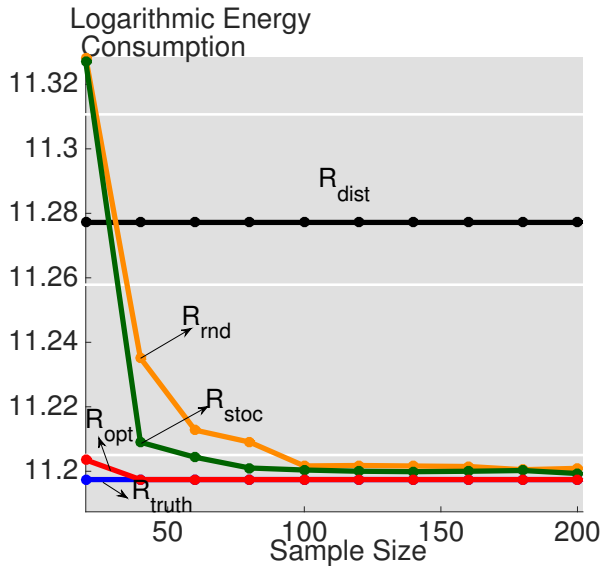


Fig. 6: Energy consumption of R_{rnd}, R_{stoc} and R_{opt} versus increasing sampling size. The performance of R_{rnd} and R_{stoc} is averaged over 100 trials. With only 40 samples R_{dist} and R_{stoc} produce route having less energy consumption then R_{dist}, and R_{opt} performs the best, since it produces the same route with the ground truth with more than 40 samples.

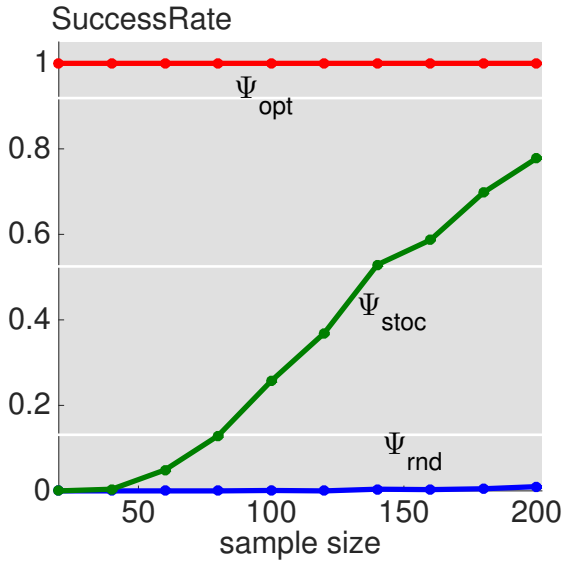


Fig. 7: Performance for the three sampling-recovery strategies. Even though the recovery error for Ψ_{rnd} may be beyond the thresholding level, it has good performance in route planning, and since Ψ_{stoc} samples according to the sampling distribution, which treat each vertex with different importance, with increasing sampling size the probability of small recovery error grows. Ψ_{opt} is the best among the three sampling strategies.

ples, R_{rnd} and R_{stoc} beat R_{dist} and R_{opt} performs exactly the same with R_{truth}.

As shown in Figure 6, R_{stoc} converges to R_{truth} faster than R_{dist}, we compare the performance of Ψ_{rnd} and Ψ_{stoc} in Figure 7. Since the query set for Ψ_{rnd} and Ψ_{stoc} is non-deterministic, the performance is evaluated as the frequently of the algorithm to recover the wind velocity with the energy of recovery error less than 0.3, which is defined as

$$\text{Success Rate} = \frac{1}{M}\sum_{i=1}^{M} \text{I}\left(\frac{\|v_{\text{recovery}} - v_{\text{original}}\|_2^2}{\|v_{\text{original}}\|_2^2} \leq 0.3\right),$$

where i denotes the incidence, and M is the total number of trials. For each sampling size, we perform M trials, and count the number of trials where the energy of recovery error is less than 30% of the energy of the original speed.

The success rate is computed based on 1000 trials for Ψ_{rnd} and Ψ_{stoc}. We can see that for the Minnesota road graph, Ψ_{stoc} outperforms Ψ_{rnd} because its success rate grows with an increasing sampling size. Ψ_{rnd} has the best performance in terms of deterministic query set and low recovery error.

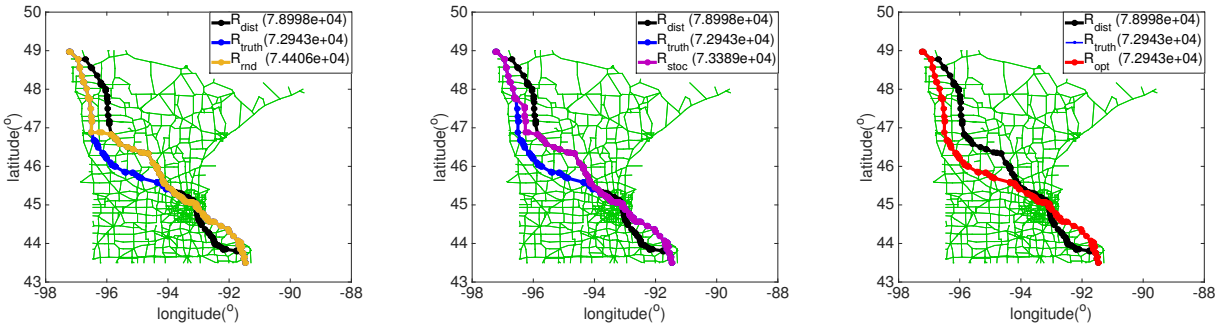
In Figure 6, we see that with 50 queries, Ψ_{opt} produces the same route with R_{truth}, and Ψ_{rnd} and Ψ_{stoc} has a high probability to produce a route that is close to R_{truth} and consume less energy than R_{dist}. In Figure 8, we compare the energy consumption of R_{rnd}, R_{stoc} and R_{opt} with R_{dist} and R_{truth}. We see that with only 50 samples both of R_{rnd} and R_{stoc} outperform R_{dist}, and R_{opt} gives exactly the same route with R_{truth}.

4.3. Future Work

As illustrated in Figure 5 wind velocity has significant influence on long distance flying route. In real life, it takes the drone hours to cover a large area, and it is of highly possibility that the original planned route may fail to be the most energy-efficient route due to the change of wind velocity with time. Besides, in reality, the wind velocity also varies across altitude, in this case the wind velocity can be further molded as a product graph signal [13]. Consequently, choosing the appropriate vertical route and plan 3D flying trajectory are also important. It is more practical to consider time evolving wind velocities.

5. CONCLUSIONS

In this paper we propose a novel method to generate an energy-efficient flight route. By taking advantages of the smoothness property of the wind velocities, we only query the wind velocities from a small number of positions and recover the rest by using a novel graph signal recovery algorithm, which solves an optimization problem. In the experiments we show the necessity to take wind into account when planning flying route



(a) Random sampling

(b) Experimentally designed sampling based on sampling distribution

(c) Experimentally designed sampling based on optimal sampling operator

Fig. 8: Flying route and energy consumption comparison. R_{rnd} and R_{stoc} perform very well in terms of saving energy consumption, and R_{opt} is the best. In (a), R_{rnd} (orange dotted line) compared with R_{dist} (black dotted line) and R_{truth} (blue dotted line). With only 50 samples, R_{rnd} covers most part of R_{truth}, and it consumes less energy than R_{dist}; In (b), R_{stoc} (pink dotted line) compared with R_{dist} (black dotted line) and R_{truth} (blue dotted line).With only 50 samples, R_{stoc} covers most part of R_{truth}, and it consumes less energy than R_{dist}; In (c), R_{opt} (red dotted line) compared with R_{dist} (black dotted line) and R_{truth} (blue dotted line). With only 50 samples, R_{opt} covers R_{truth} exactly, and it consumes the same amount as the ground truth.

and the effectiveness of our framework. We found that approximating the wind velocity as a bandlimited graph signal is a good assumption, and all three strategies give good performance for generating reliable and energy-efficient route. Since for the autonomous aerial vehicles, the computational power and storage space are quite limited, our work shows potential for a wider scope of applications. With the increasing complexity of data structure and growing volume of data size, aims such as decreasing computation complexity and expediting data access are increasingly important. We believe that signal processing on graphs, efficient data compression and robust decompression as a result have many promising applications.

6. REFERENCES

[1] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, 2013.

[2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, pp. 83–98, 2013.

[3] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4609–4624, Sept. 2015.

[4] I. Z. Pesenson, "Sampling in paley-wiener spaces on combinatorial graphs," *Trans. Amer. Math. Soc.*, vol. 360, no. 10, pp. 5603–5627, 2008.

[5] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Trans. Signal Process*, vol. 63, no. 9, pp. 2432–2444, 2015.

[6] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, pp. 6510 – 6523, Aug. 2015.

[7] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Signal recovery on graphs: Random versus experimentally designed sampling," in *SampTA*, Washington, DC, May 2015, pp. 337–341.

[8] E. Jochen, F. Stefan, and S. Sabine, "Optimal route planning for electric vehicles in large networks.," in *AAAI*, 2011.

[9] C. Zheng, L. Li, F. Xu, F. Sun, and M. Ding, "Evolutionary route planner for unmanned air vehicles," *Robotics, IEEE Transactions on*, vol. 21, no. 4, pp. 609–620, 2005.

[10] C. W. Warren, "A technique for autonomous underwater vehicle route planning," *Oceanic Engineering, IEEE Journal of*, vol. 15, no. 3, pp. 199–204, 1990.

[11] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of Signal Processing*, Cambridge University Press, 2014, http://www.fourierandwavelets.org/.

[12] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, 2014.

[13] A. Sandryhaila and J. M. F. Moura, "Big data analysis with signal processing on graphs: Representation and

processing of massive data sets with irregular structure," *Signal Processing Magazine, IEEE*, vol. 31, no. 5, pp. 80–90, 2014.