Remote Sensing with UAV and Mobile Recharging Vehicle Rendezvous

Michael H. Ostertag, Jason Ma, and Tajana Rosing

Abstract—Small unmanned aerial vehicles (UAVs) equipped with sensors offer an effective way to perform high-resolution environmental monitoring in remote areas but suffer from limited battery life. In order to perform large-scale remote sensing, a UAV must cover the area using multiple discharge cycles. A practical and efficient method to achieve full coverage is for the sensing UAV to rendezvous with a mobile recharge vehicle (MRV) for a battery exchange, which is an NP-hard problem. Existing works tackle this problem using slow genetic algorithms or greedy heuristics. We propose an alternative approach: a two-stage algorithm that iterates between dividing a region into independent subregions aligned to MRV travel and a new diffusion heuristic that performs a local exchange of points of interest between neighboring subregions. The algorithm outperforms existing state-of-the-art planners for remote sensing applications, creating more fuel efficient paths that better align with MRV travel.

I. INTRODUCTION

As costs continue to drop for sensors and sensing products, remote sensing is transforming a range of industries, including energy transfer, border security, climate monitoring, and environmental surveying and management [1]. Remote sensing missions face a trade off between access constraints and resolution requirements. High-altitude aerial or satellite surveys permit coverage of remote areas but lack the required resolution while higher resolution measurements from deploying stationary sensors or low-altitude craft are limited by road access and difficult terrain [2], [3].

An emerging application for high-resolution remote monitoring is reducing the economic costs of forest fires, which were estimated at more than \$140B in California in 2018 [4]. Costs can be reduced and damage mitigated through increased remote sensing, including powerline monitoring [5], [6], surveying fire breaks [7], [8], and early detection of forest fires [9]–[11].

Multi-rotor unmanned aerial vehicles (UAVs) equipped with cameras and relevant sensors offer a cost-effective solution, enabling close inspection of points of interest (POIs) with flexible launch and recovery due to vertical take-off and landing capabilities. Despite their high resolution sensing, UAVs are limited in aerial endurance and range by their battery life. For large-scale remote monitoring, trajectories must be planned over multiple cycles, in between which the UAV must replenish its energy by either recharging or replacing its battery. Recharging is a slow process (hours) but can be performed reliably by autonomous stations. Conversely, replacing a battery is fast (minutes) but requires complex mechanical solutions. Either

The authors are with the Computer Science Engineering Department, University of California, San Diego, La Jolla, CA 92093, USA {mosterta, jam043, tajana}@ucsd.edu.
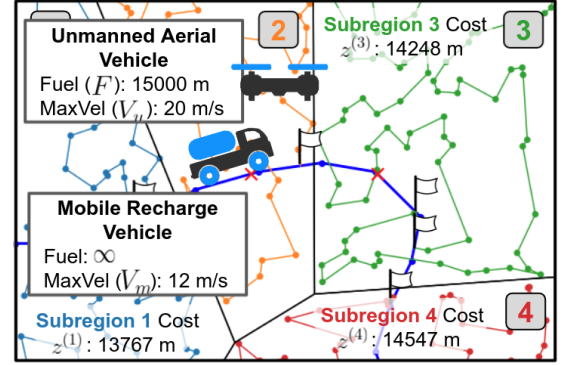

Fig. 1. Overview of remote sensing paths for a UAV and MRV team, including rendezvous locations, subregion divisions, and subregion fuel costs, after being generated by CAR-Diff, the proposed algorithm.

recharging or replacement can be effective depending on the desired operational tempo.

One option is to rendezvous with static recharge stations that either recharge or replace the battery [12]–[14]. Static recharge stations are most useful when a fixed area is persistently monitored, such as power plants [15] or regional borders [16], but would be expensive and inefficient across large regions, requiring many recharging stations, or at long intervals, such as monitoring firebreaks [7], [8] or powerline corridor surveying [5], [6] that are performed on monthly to yearly schedules.

We take an alternate approach of refuel rendezvous with mobile recharge vehicles (MRVs). A single MRV can replace an entire network of static recharge stations by traveling along existing road networks to provide refuelling capabilities as needed. The MRV-UAV refuelling rendezvous problem is a form of the Mobile Depot Vehicle Routing problem, an extension of the canonical NP-hard traveling salesman problem. Exact solutions can be computed for small problems [17], [18] but heuristics are necessary for real-world problem sizes of hundreds to thousands of POIs [17]–[24].

Current state-of-the-art algorithms are insufficient for land-based refuelling rendezvous, developing MRV paths without regard to potential travel constraints [19]–[21] or harboring an inherent assumption that the MRV can get close to the POIs [17]–[19], which is invalid for remote monitoring missions where the UAV may travel far from the rendezvous locations.

We propose an algorithm that generates trajectories for minimum-time remote sensing of a set of POIs by a UAV with UAV-MRV refuelling rendezvous called Clustering Aligned to Roadways with Diffusion Heuristic (CAR-Diff). CAR-Diff is an iterative two-stage algorithm that first transforms the mobile-depot vehicle routing problem (VRP) problem into independent TSPs by clustering the POIs based their aligned distance to the

roadway. The paths through each subregion can then be solved independently and unconstrained for which efficient methods exist. Second, we apply our proposed Diffusion Heuristic to perform local search and exchange between subregions in order to meet fuel constraints. The Diffusion Heuristic can be easily tuned between greedy and random behavior through the adjustment of several parameters. We compare CAR-Diff against the existing state-of-the-art [18], [24] and report an improvement of 7.8% avg. (25.6% max.) in monitoring time and more efficient rendezvous for real-world examples.

II. RELATED WORK

Our specific problem of remote sensing with a fuel-constrained UAV initially stems from two related research areas: trajectory planning for an MRV for an already prescribed set of UAV trajectories [25] and trajectory planning for a fuel-constrained UAV traveling between static refuelling stations [13], [14]. The combined problem forms the mobile-depot VRP (MoD-VRP), a known NP-hard problem [18].

A common approach is to constrain rendezvous to a set of discretized locations along the path, converting the problem from a Mobile-Depot VRP to a Multi-Depot VRP. Early work solved for a single path and a single MRV-UAV pair using a genetic algorithm to maximizing POIs visited [26] or iterative heuristics to improve upon an initial approximation [22]. Others solved with a split heuristic problem, which creates a single UAV tour and splits it into feasible ones given the UAV's range [19]. Maini et al. [18] follow a similar approach with their cut-and-repair heuristic.

The addition of more autonomous agents beyond a single UAV-MRV team dramatically increases the solution space. For situations that do not have constraints on MRV travel, heuristics based on genetic algorithms have shown promise [20], [21], [24], but while a solution can be generated, the solution quality is typically worse, requiring additional heuristics to fix pathing, and the solution is very dependent on hyperparameters.

The closest works to our problem of remote sensing are by Maini et al. [18] and Li et al. [24]. Maini [18] proposes two exact formulations to solve a VRP with MRV-UAV rendezvous, which can be solved exactly for small problem sizes. For refuelling rendezvous, the MRV is constrained to a road network while UAV is only constrained by a inter-rendezvous travel distance. To solve larger problems, the authors propose a cut-and-repair heuristic (Repair 2019) that solves for an approximate path with no constraints and applies constraints afterwards in an iterative greedy process. Li [24] utilizes a genetic algorithm termed the memetic algorithm (Memetic 2021). Due to the high complexity of the problem, the algorithm first attempts to cluster the POIs into subregions, enabling the memetic algorithm to better explore the solutions space.

The Repair algorithm suffers from an inherent assumption that the cost of traveling from any POI to a valid roadway will be minimal due to the greedy nature of selecting return paths. CAR-Diff addresses the issue by representing rendezvous locations by a supernode when generating paths with costs equivalent to the minimum required to travel from a POI to any rendezvous location. The Memetic algorithm clusters the POIs but each cluster can be far away from a valid rendezvous location. CAR-Diff remedies this by creating subregions that are aligned with the MRV path.

III. PROBLEM FORMULATION

Consider a problem where a set of N POIs $\mathcal{G}_q := \{q_1, \ldots, q_N\}$ where $q_i \in \mathbb{R}^2$ must all be visited in minimal time by a UAV equipped with appropriate sensors. The UAV is limited by maximum velocity V_u and fuel F, and we are interested in the scenario where the region to be covered exceeds the capacity of a UAV for a single flight cycle and the UAV must rendezvous with a MRV for refuelling.

The MRV is limited by a maximum velocity V_g but has unlimited fuel, and during each rendezvous, the UAV energy is replenished through a battery swap, which requires a constant time of τ_r. MRV travel is constrained to a path represented by the set of N_p points and linear interpolation between successive points $\mathcal{G}_p := \{\lambda p_i + (1 - \lambda)p_j \mid 0 \le \lambda < 1, j = i + 1 \ \forall \ i = \{1, \ldots N_p - 1\}\}$ where $p_i \in \mathbb{R}^2$. We model the MRV path as a supernode, which has full connectivity to all POIs with the edge cost equal to the fuel cost between the POI and the closest point in $\mathcal{G}_p$.

The POIs and rendezvous supernode form a graph $\{\mathcal{G}, \mathcal{E}\}$ with vertices representing sensing and refuelling locations $\mathcal{G} := \mathcal{G}_q \cup \mathcal{G}_p$ and edges $\mathcal{E}$ representing fuel cost for traveling between those locations. Note: to simplify notation, we use a generic index i when referencing a vertex of the graph, such that $i \in \mathcal{G}$ can represent either q_i or the closest point within the supernode $\mathcal{G}_p$.

The path between two recharge rendezvous locations is termed a cycle, and the set of all cycles is denoted as $\mathcal{C} := \{1, \ldots, K_c\}$. In order to achieve complete coverage, a single drone must travel multiple cycles or multiple drones can travel one or more cycles, depending on the drone availability. The continuity constraints ensure that each POI is visited once:

$$\sum_{i \in \mathcal{G} \setminus \{j\}} x_{ij}^{(c)} = \sum_{i \in \mathcal{G} \setminus \{j\}} x_{ji}^{(c)} \quad \forall j \in \mathcal{G}, \ c \in \mathcal{C}$$
$$\sum_{c \in \mathcal{C}} \sum_{i \in \mathcal{G}} x_{ij}^{(c)} = 1 \ \forall \ j \in \mathcal{G}_p \tag{1}$$

Fuel costs for a given cycle are tracked using the variable $z_i^{(c)}$, which is the cost of reaching POI i during cycle c.

$$z_i^{(c)} = \sum_{j \in \mathcal{G}} (z_j^{(c)} + f_{j \to i}) x_{ji}^{(c)} \quad 0 \le z_i^{(c)} \le F \ \forall \ i \in \mathcal{G}$$
$$z_k^{(c)} = 0 \ \forall \ k \in \mathcal{G}_p \tag{2}$$

For this work, we assume a linear relationship between fuel consumption and flight time:

$$f_{j \to i} = k_f t_{j \to i} \tag{3}$$

where $f_{j \to i}$ is the fuel consumed, k_f is a constant relating fuel consumption, and $t_{j \to i}$ is travel time between POIs j and i. As discussed in Sec. IV, our algorithm solves an unconstrained TSP and then checks for fuel constraint violations during the Diffusion Heuristic, allowing other more complex models of energy consumption [27] to be applied without issue.

To ensure a continuous path for each route, a subtour elimination constraint should be applied each time the optimization is run and a subtour is found. For a subset of vertices $v \subset \mathcal{G}$, we can define the set of edges leaving the subgroup as follows:

$$\sum_{(i,j)\in\mathcal{N}(v)} x_{ij}^{(d)} \geq 1 \quad , \quad \mathcal{N}(v) = \{(i,j) : i \in v, j \notin v\} \quad (4)$$

Problem 1: The objective is to visit every POI while minimizing the mission total time, which includes the time required to travel between all POIs and the time for each refuel rendezvous, and being subject to a fuel constraint.

$$\min_{x_{ij}} \sum_{c\in\mathcal{C}, i,j\in\mathcal{G}} t_{i\rightarrow j} x_{ij}^{(c)} + |\mathcal{C}|\tau_r \quad (5)$$
$$\text{s.t.} \quad \text{Eqs. (1) - (4)}$$

where $x_{ij}^{(c)} \in [0,1]^{N\times N}$ indicate if an edge was travelled on a given cycle c, $t_{i\rightarrow j}$ is the time cost to travel between q_i and q_j, and τ_r is the time required for a battery swap during rendezvous.

IV. CAR-DIFF ALGORITHM

To generate a solution for the Problem 1, we propose the Clustering Aligned to Roadways with Diffusion Heuristic (CAR-Diff) algorithm with an overview presented in Alg. 1. CAR-Diff is an iterative two-stage algorithm that first clusters the POIs based on distance to a set of subregion generator points that balance road travel with the density of nearby points (Sec. IV-A). Each cluster represents a single cycle of travel for the UAV and a shortest path is found using existing TSP solvers without fuel constraints (Sec. IV-B). Then, our Diffusion Heuristic is applied to balance the fuel costs of each subregion (Sec. IV-C). If fuel constraints are met for all subregions, the set of paths is returned. Otherwise, the number of generator points is incremented and the process repeats.

A. Path-Aligned Clustering

Voronoi tessellation (VT) is an efficient method to subdivide a region based on a set of generator points $\mathcal{V} := \{v_c \mid c \in \mathcal{C}\}$ where $v_c \in \mathcal{G}_p$ where the union of subregions cover the entire space. Each generator point v_c defines one subregion $\mathcal{R}_c$ that consists of the area closest to the generator point as defined by the L2-norm, such that $\mathcal{R}_c = \{p \mid \|p - v_m\|_2 \leq \|p - v_n\|_2 \forall m \neq n\}$. All POIs are assigned to exactly one unique subregion corresponding to a generator point and all points in the subregion must be visited in a single cycle.

When clustering, special attention must be given to the distance of the cluster to the nearest rendezvous location. Unnecessary distance between clusters and rendezvous locations can have a significant impact on fuel consumption. To enforce that each cycle can form an efficient path, CAR-Diff selects generator points along the MRV path distributed evenly across the values of an objective function J, which balances the number of nearby POIs and the distance traveled by the MRV.

Prior to calculating the objective function, the minimum-fuel rendezvous location $\tilde{p}_i \in \mathcal{G}_p$ and required fuel d_i to reach $\tilde{p}_i$

Algorithm 1 CAR-Diff Algorithm Overview

Input: POIs, Fuel Constraint, MRV Path Constraint, Initial Subregion Count
Output: UAV Paths, MRV Path
1: Compute clustering objective function (Sec. IV-A, Eq. (6))
2: **for** K_c in K_{c0} to N **do**
3:     Create K_c subregions (Sec. IV-A)
4:     Compute TSP path for each subregion (Sec. IV-B)
5:     **while** Subregion cost not stabilized **and** $H > 1$ **do**
6:         Perform Diffusion Heuristic (Sec. IV-C, Alg. 2)
7:         Recompute TSP path for each subregion (Sec. IV-B)
8:         **if** Subregion cost stabilized **then**
9:             Reduce H by decay factor
10:         **end if**
11:     **end while**
12:     **if** All subregions meet fuel constraint (Eq. (2)) **then**
13:         break
14:     **end if**
15: **end for**
16: **return** UAV Paths, MRV Path

---

are calculated for each POI (Alg. 1, Line 1). The values are used to compute J as defined per unit MRV path length l as:

$$J(l) = \frac{w_j}{L}l + \frac{(1-w_j)}{\sum_i d_i}\sum_i x_i(l)d_i \quad (6)$$

where w_j balances placement of generator points along the road with placement dependent on nearby point density, L is the total length of the road, l is the travel distance from the the first MRV point p_1 to the current linearized position on the path, $x_i(l)$ is a binary indicator if the nearest road intersection point $\tilde{p}_i$ has been passed, and d_i is the distance from POI at q_i to the nearest rendezvous point. Setting $w_j = 1$ places generator points evenly along the MRV path while $w_j = 0$ places generator points in proportion to nearby POIs. In our experimentation, setting $0 < w_j < 1$ is appropriate as including both terms balances the increased cycles for servicing dense regions with the added distance required visit sparse points.

From the objective function J, K_c generator points are selected to evenly divide the range of J, resulting in subregions that are aligned with the road. Note that the computation to form the objective function is only performed once and J does not change for successive iterations. An example of the process is provided in Figure 2, which illustrates the initial POI distribution, the assignment of each POI to a nearest rendezvous location, the formation of the objective function J, and the final subregion divisions derived from K_c generator points. The K_c subregions are passed to a TSP solver, which generates a minimum-cost path for all POIs within each subregion.

B. Pathing

For each subregion, a path is generated that services all points and begins and ends along the MRV path using OR-Tools [28]. The path starts and ends at the road supernode with actual rendezvous locations $\tilde{p}_i$ generated as the closest road locations to the first and last POI in the sequence. CAR-Diff solves for the optimal path without applying fuel constraints, enabling the
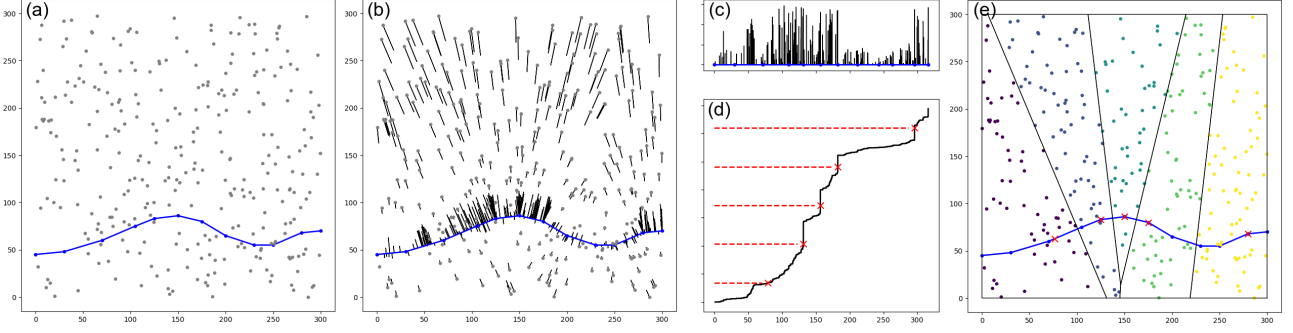
Fig. 2. Illustrated example of path-aligned clustering using 300 POIs, $w_j = 0.1$, and $K_c = 5$. (a) Initial distribution of POIs (gray) and road (blue). (b) Direction and scaled distance to nearest road point from Alg. 1. (c) Flattened road with associated distances. (d) Objective function J (shown in black) and K_c evenly selected generator points where the the red dotted lines illustrate evenly spaced values. (e) Initial segmentation from generator points.

formation of infeasible paths. The fuel cost for each subregion is used as input for the next stage, the Diffusion Heuristic.

C. Diffusion Heuristic

With a path cost $z^{(c)}$ for each subregion $\mathcal{R}_c$, CAR-Diff applies the Diffusion Heuristic, a local search technique that exchanges POIs between neighboring subregions to: first, reduce maximum subregion cost below the fuel constraint, and second, reduce the total time cost for the current cycle count. The Diffusion Heuristic is run for multiple rounds of updates, iterating between exchanging POIs between subregions and resolving the visitation order by re-pathing (Sec. IV-B).

Algorithm 2 Diffusion Heuristic

Input: $\mathcal{R}_c$, H, w_p, w_b
Output: $\mathcal{R}_c$, $z^{(c)}$
1: $p_{m,n} \leftarrow$ Eq. (7)
2: **for** h in 1 to H **do**
3:     $m, n \overset{\mathcal{U}}{\leftarrow} \{p_{m,n}\}$
4:     $p_i^{(m,n)} \leftarrow$ Eq. (8)
5:     $i \overset{\mathcal{U}}{\leftarrow} \{p_i^{(m,n)}\}$
6:     $\mathcal{R}_m \leftarrow \mathcal{R}_m \setminus q_i$, $\mathcal{R}_n \leftarrow \mathcal{R}_n + q_i$
7:     $z^{(m)} \leftarrow$ remove$(\mathcal{R}_m, q_i)$, $z^{(n)} \leftarrow$ insert$(\mathcal{R}_n, q_i)$
8: **end for**
9: **return** $\mathcal{R}_c$, $z^{(c)}$

---

where $i \overset{\mathcal{U}}{\leftarrow} \{p_i\}$ indicates randomly selecting index i in proportion to probability p_i from the set of probabilities $\{p_i\}$, remove reduces the cost of subregion m, and insert increase the cost of subregion n by a minimum cost insertion. Note that $\mathcal{R}_c$ denotes all C subregions and that $\mathcal{R}_c$ is updated by the diffusion heuristic, resulting in new subregion membership due to insertion and removal in Line 7.

POIs are selected for exchange stochastically with two coefficients w_b and w_p that can be tuned to tradeoff between greedy and random search, the classic tradeoff between exploitation and exploration. Each round consists of H exchanges of a POI between neighboring subregions. First, the exchange border and direction is selected stochastically. Then, a POI is selected for exchange based on its proximity to the exchange border.

The border and exchange direction are selected stochastically in proportion to the probability p_{ij} defined by the difference of fuel costs for each region proportional to the border length and scaled by a softmax function as follows:

$$p_{ij} \sim \frac{\exp\left(w_b C_{ij}\right)}{\sum_{m,n \in \{1,\ldots,K_c\}} \exp\left(w_b C_{mn}\right)}$$
$$C_{ij} = b_{ij}\left(z^{(i)} - z^{(j)}\right) \quad w_b = \frac{w_b'}{F \, b_{avg}} \quad (7)$$

where p_{ij} is the exchange probability of a POI from subregion i to j using scaling factor w_b, C_{ij} is the scaled cost difference between neighboring subregions, $b_{ij} = b_{ji}$ is the border length between subregion i and j, F is the maximum fuel constraint, and b_{avg} is the average border length between subregions. For subregions that do not share a border, the possibility of selecting the border for exchange is removed by setting $C_{ij} = -\infty$. Similar to diffusion, the exchange rate scales with the difference in fuel cost across and the size of the border where higher difference in neighboring subregion costs and larger borders result in a higher probability of POI exchange.

The scaling factor w_b controls the greediness of the border selection. As $w_b \to 0$, the exchange approaches uniformly random behavior. Conversely, as $w_b \to \infty$, the exchange selection becomes greedy and $p_{ij} \to 1$ for the border between subregion i and j that has the highest scaled cost difference.

After the border and direction of exchange are selected, a POI is selected based on its proximity to the exchange border. For a transfer from subregion i to subregion j, let $d_i^\perp$ be the distance to the border for all points in i. The POI is selected randomly using probabilities generated from a softmax function on the distance to the border weighted by w_p:

$$p_m^{(i,j)} = \frac{\exp(w_p d_m^\perp)}{\sum_n \exp(w_p d_n^\perp)} \quad (8)$$

where $d_m^\perp$ is the distance from POI m to the border between subregions i and j. Unlike the probabilities in Eq. (7), which must be normalized, $p_m^{(i,j)}$ from Eq. (8) does not require normalization since only the points from a single region are considered at a time for exchange.

The POI selected for exchange is inserted into the path of the receiving subregion to minimize the increase in total length

from a simple insertion. The direct insertion may result in a suboptimal path, so after H updates, the path through each subregion is recomputed using the previously solved path with insertions as the initial solution to improve convergence.

Once the cost of each subregion stabilizes as defined by the best known solution not improving for K_i iterations, the number of updates H contracts by a decay factor λ, such that $H \leftarrow \lambda H$. If the cost of any region is above the fuel restriction F and $H = 1$, then the constraints are deemed to not have been met and the number of subregions K_c is incremented. A new set of generator points are selected according to the method outlined in Sec. IV-A and the algorithm proceeds until the minimum K_c that meets the fuel constraints. The end result is a set of K_c trajectories, one for each subregion, and an ordered set of $2K_c$ points along the road that signify the rendezvous points between the UAV and MRV.

V. RESULTS

The effectiveness of CAR-Diff for remote sensing was tested by simulation over one synthetic (circular path with POIs internal) and one realistic scenario (72 km loop near ignition point of 2003 Cedar Fire in San Diego). For each, we performed 5 runs of uniformly random distributed 100, 300, 600, and 1000 POIs in the sensing region that were within 2 km to 5 km of the MRV path. We assume the UAV has a constant velocity of 10 m/s for a 15 km fuel capacity in line with commercially available UAV systems [29] and a recovery, battery exchange, and launch time of $\tau_r = 600$ s.

We configure CAR-Diff in Random ($w_b = 0.2$), Balanced ($w_b = 2.0$), and Greedy ($w_b = 20$) exchange configurations and compare against two state-of-the-art methods: a Repair Heuristic [18] (Repair 2019) and a memetic algorithm [24] (Memetic 2021). For all tests, each algorithm was run sequentially on the same PC with an Intel i5-8600k CPU at 3.6 GHz and computation time was limited to 25 min, the maximum time for a single cycle. The best trajectory found over that time frame or the final trajectory if the algorithm completed is reported.

A. Comparison Algorithms

The Repair Heuristic [18] solves for an initial tour with no constraints using the Lin-Kernighan-Helsgaun heuristic. Progressing along the tour, when the fuel constraint is violated, a direct path is found by attempting to connect the current tour point to a rendezvous location, iterating backwards until the constraint is satisfied. The Memetic Algorithm [24] performs initial K-means clustering of the regions before application of a genetic algorithm and local search, iteratively optimizing at a cluster and local level. We use the recommended parameters from the original publication ([24], Table II) with an incrementing number of clusters until the fuel constraint is met. All reported times are for the last run of the algorithm, assuming the correct number of clusters was initially selected.

B. Experimental Results

Improvement in total path cost was normalized to the best performing state-of-the-art algorithm (Repair 2019). CAR-Diff outperformed Repair 2019 by 7.8% avg. (25.6% max.) in
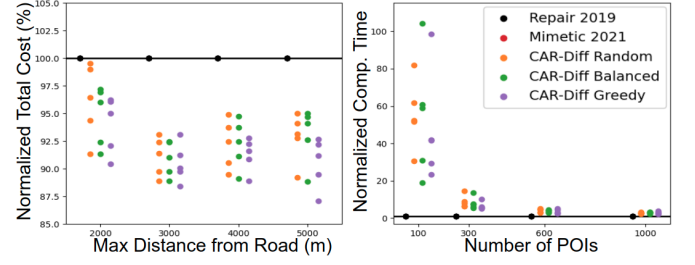

Fig. 3. Summary of Repair 2019 [18], Memetic 2021 [24], and CAR-Diff algorithms for path planning of 1000 POIs near the origin of Cedar Creek Fire. (left) Total cost reduction normalized to Repair 2019. (right) Relative time compared to Repair 2019 as the number of POIs increases from 100 to 1000.
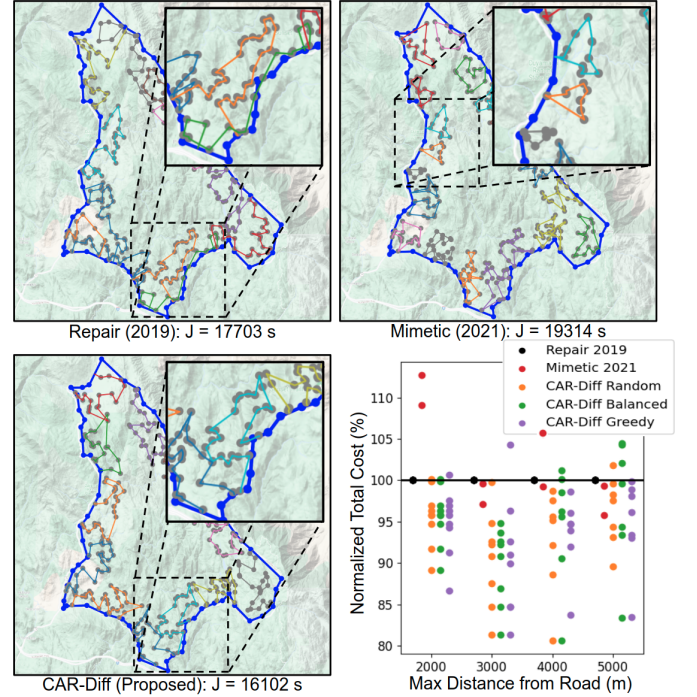

Fig. 4. Comparison of Repair 2019 [18] (top-left), Memetic 2021 [24] (top-right), and CAR-Diff Balanced (bot-left) methods for $N = 300$ POIs near the origin of Cedar Creek Fire. Note the inefficient pathing for rendezvous with the Maini algorithm and the inefficiency of clustering using Euclidean distance when selecting subsets. (bot-right) Performance summary for 100-1000 POIs.

monitoring time across both the synthetic and real-world scenarios. For the synthetic example of a circular path, the highest improvement was achieved for POIs that were distributed up to 5 km from the MRV path (6.9% average improvement) as opposed to when POIs were distributed closer to the MRV path (5.2% average improvement). CAR-Diff was designed for remote sensing applications where POIs are far or difficult to access from paths, which these results support. For small problem sizes of 100 POIs, Repair 2019 and CAR-Diff produce similar solutions as seen in Fig. 4 (bot-right). The plot was included to show the limited results that Memetic 2021 could produce within the time limits of the simulation.

Due to inefficient clustering of POIs that was not aligned to MRV paths and an inability to perform local swaps between subregions, the Memetic 2021 algorithm planned more cycles on average compared to Repair 2019 and CAR-Diff. For 100

POIs in the Cedar Fire loop, Memetic 2021 required an average of 12.2 refuel rendezvous compared to 7.8 and 8.4 for Repair 2019 and CAR-Diff, respectively. The effect of the design decisions of Repair 2019, Memetic 2021, and CAR-Diff can be viewed in Fig. 4, which shows a set of example paths for the Cedar Fire monitoring example. Note the small clusters generated by Mimemtic 2021 and the difference in path quality formed without (Repair 2019) and with (CAR-Diff) clustering.

Computation time for the algorithms is not critical for the operational tempo of typical surveying missions as this can be done offline, but it does offer insight into how the clustering of CAR-Diff results in competitive computation times for larger problem sizes. For 1000 POIs, the average computation time for Repair 2019 was 46 s, nearly all on computing an initial path for repair and average time for CAR-Diff was 117 s (random), 114 s (balanced), and 111 s (greedy). The time to compute CAR-Diff approaches Repair 2019 for larger problem sizes due to the exponential complexity of solving the TSP in Fig. 3(bot). CAR-Diff solves the TSP many more times than Repair 2019, but each problem instance is smaller and can be computed in parallel due to the road-aligned clustering. Memetic 2021 was unable to generate a solution within the allotted time for any problem size greater than 100 POIs, which required 5000-7500x the time required by Repair 2019.

VI. Conclusion

We proposed CAR-Diff, a two-stage algorithm to plan trajectories for UAV remote sensing with refuelling rendezvous. By aligning clustering with the mobile recharging vehicle path, CAR-Diff can generate subregions with low-cost paths to rendezvous locations. The proposed algorithm can be configured for a tradeoff between computation time and solution quality with the adjustment of two parameters. CAR-Diff was compared against existing works [18], [24] and showed up to 25.6% improvement in monitoring time, enabling more efficient surveying of the environment.

References

[1] Ismael Colomina and Pere Molina. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS Journal of photogrammetry and remote sensing*, 92:79–97, 2014.

[2] Grégory Duveiller and Pierre Defourny. A conceptual framework to define the spatial resolution requirements for agricultural monitoring using remote sensing. *Remote Sensing of Environment*, 114(11):2637–2650, 2010.

[3] Chi Yuan, Youmin Zhang, and Zhixiang Liu. A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. *Canadian journal of forest research*, 45(7):783–792, 2015.

[4] Daoping Wang, Dabo Guan, Shupeng Zhu, Michael Mac Kinnon, Guannan Geng, Qiang Zhang, Heran Zheng, Tianyang Lei, Shuai Shao, Peng Gong, et al. Economic footprint of california wildfires in 2018. *Nature Sustainability*, 4(3):252–260, 2021.

[5] Junaid Ahmad, Aamir Saeed Malik, Likun Xia, and Nadia Ashikin. Vegetation encroachment monitoring for transmission lines right-of-ways: A survey. *Electric Power Systems Research*, 95:339–352, 2013.

[6] Yong Zhang, Xiuxiao Yuan, Wenzhuo Li, and Shiyu Chen. Automatic power line inspection using uav images. *Remote Sensing*, 9(8):824, 2017.

[7] João E Pereira-Pires, Valentine Aubard, Rita A Ribeiro, José M Fonseca, João Silva, and André Mora. Semi-automatic methodology for fire break maintenance operations detection with sentinel-2 imagery and artificial neural network. *Remote Sensing*, 12(6):909, 2020.

[8] Davide Ascoli, Lucia Russo, Francesco Giannino, Constantinos Siettos, Francisco Moreira, et al. Firebreak and fuelbreak. *Encyclopedia of Wildfires and Wildland-Urban Interface Fires; Manzello, S., Ed*, 2018.

[9] Yi Zhao, Jiale Ma, Xiaohui Li, and Jie Zhang. Saliency detection and deep learning-based wildfire identification in uav imagery. *Sensors*, 18(3):712, 2018.

[10] Jie Shi, Wei Wang, Yuanqi Gao, and Nanpeng Yu. Optimal placement and intelligent smoke detection algorithm for wildfire-monitoring cameras. *IEEE Access*, 8:72326–72339, 2020.

[11] Osama M Bushnaq, Anas Chaaban, and Tareq Y Al-Naffouri. The role of uav-iot networks in future wildfire detection. *IEEE Internet of Things Journal*, 2021.

[12] F Paulo Kemper, Koji AO Suzuki, and James R Morrison. Uav consumable replenishment: design concepts for automated service stations. *Journal of Intelligent & Robotic Systems*, 61(1):369–397, 2011.

[13] Jonghoe Kim, Byung Duk Song, and James R Morrison. On the scheduling of systems of uavs and fuel service stations for long-term mission fulfillment. *Journal of Intelligent & Robotic Systems*, 70(1):347–359, 2013.

[14] Rafael Santin, Luciana Assis, Alessandro Vivas, and Luciano CA Pimenta. Matheuristics for multi-uav routing and recharge station location for complete area coverage. *Sensors*, 21(5):1705, 2021.

[15] Herman Fesenko, Ihor Kliushnikov, Vyacheslav Kharchenko, Serhii Rudakov, and Elena Odarushchenko. Routing an unmanned aerial vehicle during npp monitoring in the presence of an automatic battery replacement aerial system. In *2020 IEEE 11th international conference on dependable systems, services and technologies (DESSERT)*, pages 34–39. IEEE, 2020.

[16] Seon Jin Kim and Gino J Lim. Drone-aided border surveillance with an electrification line battery charging system. *Journal of Intelligent & Robotic Systems*, 92(3):657–670, 2018.

[17] Kevin Yu, Ashish Kumar Budhiraja, Spencer Buebel, and Pratap Tokekar. Algorithms and experiments on routing of unmanned aerial vehicles with mobile recharging stations. *Journal of Field Robotics*, 36(3):602–616, 2019.

[18] Parikshit Maini, Kaarthik Sundar, Mandeep Singh, Sivakumar Rathinam, and PB Sujit. Cooperative aerial–ground vehicle route planning with fuel constraints for coverage applications. *IEEE Transactions on Aerospace and Electronic Systems*, 55(6):3016–3028, 2019.

[19] Yao Liu, Zhihao Luo, Zhong Liu, Jianmai Shi, and Guangquan Cheng. Cooperative routing problem for ground vehicle and unmanned aerial vehicle: The application on intelligence, surveillance, and reconnaissance missions. *IEEE Access*, 7:63504–63518, 2019.

[20] Bingxi Li, Brian R Page, John Hoffman, Barzin Moridian, and Nina Mahmoudian. Rendezvous planning for multiple auvs with mobile charging stations in dynamic currents. *IEEE Robotics and Automation Letters*, 4(2):1653–1660, 2019.

[21] Bingxi Li, Brian R Page, Barzin Moridian, and Nina Mahmoudian. Collaborative mission planning for long-term operation considering energy limitations. *IEEE Robotics and Automation Letters*, 5(3):4751–4758, 2020.

[22] Wei Gao, Junren Luo, Wanpeng Zhang, Weilin Yuan, and Zhiyong Liao. Commanding cooperative ugv-uav with nested vehicle routing for emergency resource delivery. *IEEE Access*, 8:215691–215704, 2020.

[23] Satyanarayana G Manyam, Kaarthik Sundar, and David W Casbeer. Cooperative routing for an air–ground vehicle team—exact algorithm, transformation method, and heuristics. *IEEE Transactions on Automation Science and Engineering*, 17(1):537–547, 2019.

[24] Jianqiang Li, Tao Sun, Xiaopeng Huang, Lijia Ma, Qiuzhen Lin, Jie Chen, and Victor CM Leung. A memetic path planning algorithm for unmanned air/ground vehicle cooperative detection systems. *IEEE Transactions on Automation Science and Engineering*, 2021.

[25] Neil Mathew, Stephen L Smith, and Steven L Waslander. Multirobot rendezvous planning for recharging in persistent tasks. *IEEE Transactions on Robotics*, 31(1):128–142, 2015.

[26] Halil Savuran and Murat Karakaya. Efficient route planning for an unmanned air vehicle deployed on a moving carrier. *Soft Computing*, 20(7):2905–2920, 2016.

[27] Shimaa Ahmed, Amr Mohamed, Khaled Harras, Mohamed Kholief, and Saleh Mesbah. Energy efficient path planning techniques for uav-based systems with space discretization. In *2016 IEEE Wireless Communications and Networking Conference*, pages 1–6. IEEE, 2016.

[28] Laurent Perron and Vincent Furnon. Or-tools 7.2. https://developers.google.com/optimization/.

[29] Phantom 4 pro v2.0. https://www.dji.com/phantom-4-pro-v2/specs.