2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Risk-aware Recharging Rendezvous for a Collaborative Team of UAVs and UGVs

Ahmad Bilal Asghar,[1*] Guangyao Shi,[1*] Nare Karapetyan,[1] James Humann,[2]
Jean-Paul Reddinger,[2] James Dotterweich,[2] Pratap Tokekar[1]

***Abstract*— We introduce and investigate the recharging rendezvous problem for a collaborative team of Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs), in which UAVs with limited battery capacity and UGVS persistently monitor an area. The UGVs also act as mobile recharging stations for the UAVs. In contrast to prior work on such problems, we consider the challenge of dealing with stochastic energy consumption in a risk-aware fashion. Specifically, we consider a bi-criteria optimization problem of minimizing the time taken by the UAVs on recharging detours while ensuring that the probability that no UAV runs out of charge is greater than a user-defined risk tolerance. This problem (termed Risk-aware Recharging Rendezvous Problem (RRRP)) is a combinatorial problem with a matching constraint — to ensure UAVs are assigned to the limited UGV recharging slots, and a knapsack constraint — to capture the risk tolerance. We propose a novel bicriteria approximation algorithm to solve RRRP and demonstrate its effectiveness in the context of a persistent monitoring mission compared to baseline methods.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) are increasingly being used in applications such as surveillance [1], [2], package delivery [3], [4], environmental monitoring [5], [6], and precision agriculture [7] due to their ability to monitor large areas in a short period of time. One bottleneck that hinders long-term deployments of UAVs is their limited battery capacity. Recently, there have been efforts in overcoming this bottleneck by using Unmanned Ground Vehicles (UGVs) as mobile recharging stations [8]–[19]. However, these works make the simplifying but restrictive assumption that the energy consumption of the UAV is deterministic [10]–[12]. In practice, the energy consumption is stochastic and the planning algorithms must be able to deal with the risk of running out of charge. In our recent work [9], we presented a risk-aware planning algorithm for planning for a single UAV and UGV. However, that algorithm scales exponentially with the planning horizon and the number of robots, and will require decomposition of the task for multiple robots [20], [21]. In this paper, we present an efficient risk-aware coordination algorithm for scalable, long-term UAV-UGV missions.

We consider a scenario where the UAVs and the UGVs are executing a persistent monitoring mission by visiting a sequence of *task* nodes in a given order (Figure 1). The UAVs can take a detour from the planned mission to rendezvous with some UGV and recharge. The UGV can recharge the UAV while moving [9], [10], [17]. We present several algorithms that decide *when* and *where* the UAVs should recharge and *which* UGV they should recharge on.

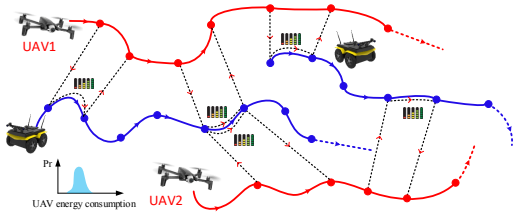


Fig. 1. An illustrative example of the recharging rendezvous problem considered in this paper. A team of two UAVs and two UGVs is executing their tasks in a predefined order. The UAVs need to decide when and where to land on the UGVs in order to recharge minimizing the total detour time. The energy consumption of the UAVs is stochastic. The black dashed lines represent the recharging detours.

To take into account the stochastic nature of UAV's energy consumption, one could take recharging detours more frequently, thereby reducing the risk of running out of charge.[1] However, if a UAV takes a detour, then the task nodes visited after the detour will experience a delay, thereby worsening the task performance. Since we are planning over a longer horizon with multiple UAVs and UGVs, there is a complex trade-off between task performance and risk tolerance which requires careful planning and coordination.

To study this trade-off, we introduce the Risk-aware Recharging Rendezvous Problem (RRRP), which is a combinatorial optimization problem with matching and knapsack constraints. While there are sophisticated Integer Linear Programming (ILP) solvers, we show how to exploit the structural properties of the problem to yield more efficient algorithms with theoretical performance guarantees. Other combinatorial problems with knapsack or budget constraints have been studied in the literature. Approximation algorithms for the budgeted version of the minimum spanning tree problem and maximum weight matching problem are presented in [22] and [23] respectively. They both use Lagrangian relaxation which resembles our approach, however, due to different problem structures, we propose several algorithms, including a bicriteria approximation algorithm, to solve RRRP. We demonstrate the effectiveness of our formulation

This work is supported in part by National Science Foundation Grant No. 1943368 and Army Grant No. W911NF2120076.

* indicates equal contribution and authors are listed alphabetically

[1]University of Maryland, College Park, MD 20742 USA [gyshi, knare, abasghar, tokekar]@umd.edu

[2]DEVCOM Army Research Laboratory, MD USA jean-paul.f.reddinger.civ, james.d.humann.civ, james.m.dotterweich.civ]@army.mil

[1]In practice, when a UAV is about to run out of charge, it can land and then be retrieved by a human. By reducing the risk of running out of charge, we aim to reduce the overhead of such costly human interventions.

and algorithm in the context of a persistent monitoring mission.

The main contributions of this paper are:

- We introduce RRRP as the first risk-aware version of the cooperative UAV-UGV recharging problem. We show how to formulate the RRRP as a matching problem on a bipartite graph with an additional knapsack constraint.
- We present several algorithms to solve RRRP, that build on each other.
- We demonstrate the effectiveness of our formulation and the proposed algorithm in the context of a persistent monitoring mission compared to baseline greedy methods and ILP solvers.

II. Finite Horizon Risk-Aware Recharging Rendezvous Problem (RRRP)

We first define the RRRP for a finite horizon, and then show how to formulate it as a matching problem and model the risk-tolerance as a knapsack constraint.

A. Problem Statement

Consider a team of N_a UAVs and N_g UGVs persistently monitoring a set of locations in an environment. The UAVs and UGVs move on the graphs $G_a = (U_a, E_a)$ and $G_g = (U_g, E_g)$ with their deterministic speeds v_a and v_g respectively. The vertex sets U_a and U_g represent the locations to be monitored by the aerial and ground vehicles respectively. The edge set E_a may be complete since the UAVs can move between any of the tasks, whereas the edge set E_g represents the road network on which the ground vehicles can move. We assume that the ordering of the task nodes for the UAVs and the UGVs is given, i.e., we have pre-defined persistent monitoring tours for the UAVs and UGVs. These tours can be generated by planners that either do not consider recharging [24]–[26] or assume deterministic discharge [12], [14]. The tours for i^{th} UAV and k^{th} UGV are denoted by $\mathcal{T}_i^a$ and $\mathcal{T}_k^g$ respectively. Similar to our recent work for the single robot case [9], we show how to refine these tours in a risk-aware fashion.

A UAV i can take a *detour* from its monitoring tour $\mathcal{T}_i^a$ at any point along the tour to rendezvous with a UGV k and land on it, for recharging. The UGV keeps moving along its tour $\mathcal{T}_k^g$. We only model the case where only UAVs take detours, and not UGVs, since UAVs are typically much faster and not restricted to the road network. The UAV can also wait at a rendezvous location for the UGV if it reaches there before the UGV. The number of UAVs that can simultaneously charge on a given UGV is d. Once recharged, the UAV leaves the UGV and goes to the next task node along its monitoring tour.

We assume that the UGVs do not run out of charge since typically they have much larger battery capacity than UAVs or can be easily refueled. We consider a stochastic energy discharge model for the UAVs that is given. We assume that the probability of a UAV running out of charge within the next t time units given the current charge level can be calculated, as shown in [9]. The stochastic battery discharge is monotonic in the time traveled by the UAV.

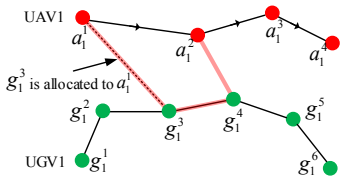


Fig. 2. Recharging detour example. The UAV 1 leaves its route at the node a_1^1 to rendezvous with UGV 1 at the node g_1^3. The recharging occurs when they reach the node g_1^4 and the UAV moves to its next task node a_1^2. Note that if $d = 1$, no other UAV can recharge on this UGV between g_1^3 and g_1^4.

Since we consider a persistent monitoring mission, we solve RRRP in a receding-horizon manner. Given time horizon T, we seek a policy for the UAVs to recharge at most once in the horizon. Moreover, as the battery discharge rate of UAVs is stochastic, there may be a non-zero probability of some UAVs running out of charge. Hence, we also need to have a notion of risk-aversion for the UAVs. On the other hand, to avoid frequent recharging, we also need to reduce the detour time spent by the UAVs for recharging.

At a high level, we consider the following problem: *Given a time horizon T, a risk-tolerance probability $\rho \in (0, 1)$, task routes for the N_a UAVs and N_g UGVs along with their current locations and state of charge, find a recharging schedule for each UAV such that:*

1) *each UAV recharges at most once during T,*
2) *no UGV can charge more than d UAVs at a time,*
3) *the probability that no UAV runs out of charge during T is at least ρ, and*
4) *the total detour time of the UAVs is minimized.*

Next, we will show how to formally model this problem.

B. Modeling RRRP as a Matching Problem with Knapsack Constraint

Given the tour $\mathcal{T}_k^g$ for UGV k, we discretize it by introducing vertices every f units of time starting from the UGV's current position where f is the maximum time the UGV needs to travel for the UAV to recharge in the worst-case. These vertices representing possible rendezvous locations are denoted by $V(\mathcal{T}_k^g)$ and are shown as green nodes in Figure 2. Similarly, the set $V(\mathcal{T}_i^a)$ represents the set of locations from which UAV i can leave its monitoring tour $\mathcal{T}_i^a$ for a recharging detour (Figure 2). The set $V(\mathcal{T}_i^a)$ contains the current position of UAV i and its task nodes that can be visited within the next T time by UAV i. We do not need to further discretize UAVs tours as it can be shown [27, Lemma 6] that there exists an optimal solution where the UAVs will leave their tours for recharging from either their current position or a task node.

We define the RRRP formally on a bipartite graph $G = (V_a \cup V_g, E)$ as follows.

UAV vertices: The vertex set V_a consists of N_a disjoint node sets, i.e., $V_a = \cup_{i=1}^{N_a} \mathcal{V}_i$ where $\mathcal{V}_i = V(\mathcal{T}_i^a) \cup a_i^{\emptyset}$. The vertex $a_i^{\emptyset}$ represents the scenario where the UAV i chooses not to rendezvous in the current horizon.

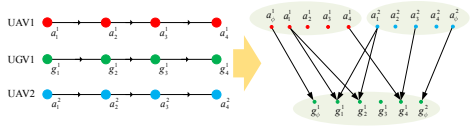


Fig. 3. An example to show how to construct a bipartite graph for one planning horizon from the UAV and UGV route segments.

UGV vertices: The vertex set V_g consists of $\{g_1^\emptyset, \ldots, g_i^\emptyset, \ldots, g_{N_a}^\emptyset\}$ and d copies of the set $\cup_{k=1}^{N_g} V(\mathcal{T}_k^g)$. The vertex $g_i^\emptyset$ for UAV i represents the scenario where UAV i chooses not to rendezvous for the current horizon. The d copies of each vertex in $V(\mathcal{T}_k^g)$ are to represent up to d different UAVs recharging at a time on a UGV.[2]

Edges: The edge set E denotes the set of all feasible recharging detour options. If the UAV i, starting from a node $j \in V(\mathcal{T}_i^a)$, is able to rendezvous with the UGV k at $l \in V(\mathcal{T}_k^g)$, an edge exist between the corresponding two nodes in V_a and V_g. The vertex $a_i^\emptyset$ is connected to $g_i^\emptyset$ for all $i \in [N_a]$.

Edge cost: For an edge $(i,j) \in E$ where $i \in V_a$ and $j \in V_g$, the edge cost c_{ij} represents the time needed to complete the recharging detour along the edge (i,j). The time needed for the recharging detour consists of three parts: the time to reach the rendezvous node, the waiting time and the recharging time at the rendezvous node, and the time to go to the next node on the tour. The edge cost along edge $(a_i^\emptyset, g_i^\emptyset)$ is zero.

Edge success probability: For an edge $(i,j) \in E$ where $i \in V_a$ and $j \in V_g$, the edge success probability p_{ij} is defined as the overall probability to finish the task route for the current horizon given that the recharging detour along edge (i,j) is taken. It is the product of the two success probabilities: the probability of successfully completing the recharging detour, and the probability of finishing the rest of the route after recharging. The probability along edge $(a_i^\emptyset, g_i^\emptyset)$ is the probability of reaching the end of the current horizon without recharging.

The finite horizon recharge rendezvous problem can now be defined as the following combinatorial optimization problem.

Problem 1 (Risk-aware Recharging Rendezvous Problem (RRRP))**.** *Given a bipartite graph $G = (V_a \cup V_g, E)$ where $V_a = \cup_{i=1}^{N_a} \mathcal{V}_i$, with edge costs c_{ij} and probabilities p_{ij} for edge $(i,j) \in E$, solve:*

$$\min \sum_{i,j} c_{ij} x_{ij}, \qquad (1)$$

$$\text{s.t.} \sum_{i,j} \log \frac{1}{p_{ij}} x_{ij} \leq \log \frac{1}{\rho}, \qquad (2) \qquad \sum_i x_{ij} \leq 1 \ \forall j, \qquad (3)$$

$$\sum_{j,\ i \in \mathcal{V}_r} x_{ij} = 1, \forall r, \qquad (4) \qquad x_{ij} \in \{0,1\}. \qquad (5)$$

The variable x_{ij} indicates whether edge (i,j) is in the solution or not. The objective is to minimize the total time incurred by the recharging detours of all UAVs. Constraint (2) enforces that the probability of no UAV running out of charge during the current horizon is at least ρ. We can write this as a linear constraint since the stochastic discharging processes of the UAVs are independent. For ease of notation, we will use the following inequality instead of Constraint (2).

$$\sum_{i,j} a_{ij} x_{ij} \leq B \qquad (6)$$

where $B = \log 1/\rho$ and $a_{ij} = \log 1/p_{ij}$. Constraint (3) enforces that each UGV can recharge at most d UAVs at a time (by making sure that at most one UAV recharges at one of the d copies of a UGV vertex). Constraint (4) enforces that each UAV should be recharged at most once. Given a solution x_l, let M_l represent the corresponding edge set on graph G. Let us define $c(M) = c(x) = \sum c_{ij} x_{ij}$, $a(M) = a(x) = \sum a_{ij} x_{ij}$ for ease of notation.

A reduction from the problem EVENODDPARTITION [28] shows that Problem 1 is NP-hard. The proof is omitted due to space constraints. See the extended version [27] for all proofs.

Lemma 1. *Problem 1 is NP-hard.*

In the next section, we present our approach for solving this problem.

III. ALGORITHMS AND ANALYSIS

Although Problem 1 can be solved using ILP solvers, as the number of variables in the problem increases, the runtime of the solver becomes intractable. Since we need to solve this problem repeatedly in a receding horizon approach, an efficient solver is desirable. In this section, we present a series of algorithms to solve Problem 1, that build on top of each other: Algorithm 1 finds a heuristic solution, Algorithm 2 improves on it, and Algorithm 3 uses Algorithm 2 to provide a solution with bicriteria approximation guarantees.

A. *Algorithm 1:* BINARYSEARCH

We start with the following relaxation of Problem 1.

Problem 2.

$$\min \sum_{i,j} c_{ij} x_{ij} + \lambda \left(\sum_{i,j} a_{ij} x_{ij} - B \right), \qquad (7)$$

such that Constraints (3)*,* (4) *and* (5) *hold.*

Let $w_\lambda(M) = w_\lambda(x) = c(x) + \lambda a(x)$. The key insight behind the algorithms is that Problem 2 can be solved in polynomial time using a minimum cost network flow problem with edge costs w_λ.

Lemma 2. *Problem 2 is solvable in $O(|E| \log |V| (|E| + |V| \log(V))$ time on graph $G(V_a \cup V_b, E)$ where $V = V_a \cup V_b$.*

Algorithm 1 is a heuristic algorithm that uses binary search on the Lagrangian multiplier λ to find a feasible and an infeasible solution (with respect to Constraint (6)), i.e., M_1 and M_2 such that $a(M_1) \leq B \leq a(M_2)$.[3] The following observation enables us to use binary search in Algorithm 1.

[2]We use d copies of each vertex in $V(\mathcal{T}_k^g)$ to keep the analysis of the algorithm simple. The problem can be defined without having d copies for each $v \in V(\mathcal{T}_k^g)$ by changing Constraint (3) in Problem 1 to $\sum_i x_{ij} \leq d$.

[3]Note that if no solution M_2 exists such that $a(M_2) > B$, then solving Problem 2 with $\lambda = 0$ solves Problem 1.

Observation 1. *Let x_1 and x_2 be the optimal solutions to Problem 2 with Lagrangian multipliers λ_1 and λ_2, where $\lambda_1 > \lambda_2$. Then $a(x_1) \leq a(x_2)$, and $c(x_1) \geq c(x_2)$.*

Although the run time of Algorithm 1 depends on the stopping threshold value $\Delta\lambda_{\min}$, an optimal Lagrangian multiplier λ and two solutions x_1 and x_2 to Problem 2, with $w_\lambda(x_1) = w_\lambda(x_2)$ and satisfying $a(x_1) \leq B \leq a(x_2)$ and $c(x_1) \geq c(x_2)$ can be found in polynomial time [23], [29, Theorem 24.3]. We use Algorithm 1 in experiments to find M_1 and M_2 as it is simple to implement and works well in practice as seen in Section IV.

B. Algorithm 2: LocalSearch

In order to improve the solution returned by Algorithm 1, we consider the symmetric difference of the solutions M_1 and M_2. The symmetric difference $M_1 \oplus M_2$ is a disjoint union of connected components where a connected component is defined below.

Definition 1. Given the bipartite graph $G = (V_a \cup V_g, E)$ where $V_a = \cup_{i=1}^{N_a} \mathcal{V}_i$, consider the bipartite graph G' with the vertices for each robot merged together, i.e., $G' = (V_c \cup V_g, E')$ where $V_c = \{\mathcal{V}_1, \ldots, \mathcal{V}_{N_a}\}$, and E' has an edge between $\mathcal{V}_r$ and $j \in V_g$ if E has an edge between some $i \in \mathcal{V}_r$ and j. Then with abuse of notation, we define a subgraph H of G as a *connected component* if the edges of H form a connected path or cycle in G'.

Algorithm 2 is a local search that takes the two solutions returned by Algorithm 1, and repeatedly improves them by removing connected components from the symmetric difference until $M_1 \oplus M_2$ contains exactly one connected component. This procedure is similar to that of finding two adjacent extreme point solutions in the solution polytope for a maximum weight matching problem [23]. The solution returned by Algorithm 2 has the following properties.

Lemma 3. *Algorithm 2 returns λ and two solutions M_1, M_2, such that their symmetric difference contains exactly one connected component, and*
1) $w_\lambda(M_1) = w_\lambda(M_2)$, 2) $a(M_1) \leq B \leq a(M_2)$, 3) $c(M_1) \geq c(M_2)$, 4) $c(M_1) \leq \texttt{opt} + \lambda B$.

C. Algorithm 3: RendezvousSearch

Algorithm 3 is a bicriteria approximation algorithm, which uses Algortithm 2 as a subroutine. Given M_1, M_2 and λ from Algorithm 2, in Lines 8 through 15 of Algorithm 3, we find a sequence Z'' of edges from the connected component $M_1 \oplus M_2$ such that $M_1 \oplus Z''$ has a better cost than M_1 and $a(M_1 \oplus Z'') \leq B$. The following lemma is the main result of this section and shows that Algorithm 3 finds a solution M that may violate at most one constraint and the cost of M is within $c_{\max}$ of the optimal, where $c_{\max}$ is the largest edge cost in G.

Lemma 4. *Let M^* be the optimal solution to Problem 1 with cost $\texttt{opt}$. There is a polynomial time algorithm that finds a scheduling M such that $c(M) \leq \texttt{opt} + c_{max}$, and one of the UGVs may have to recharge at most $d+1$ UAVs at a time.*

Algorithm 1 BinarySearch

Input: Graph $G = (V_a \cup V_g, E)$ with edge costs c and weights a, bound B

1: Set a threshold $\Delta\lambda_{\min}$
2: $\lambda_l \leftarrow 0$, $\lambda_u \leftarrow \infty$
3: Start from a positive value of λ
4: **while** $\lambda_u - \lambda_l \geq \Delta\lambda_{\min}$ **do**
5: Solve Problem 2 using λ to get solution x
6: **if** $a(x) \leq B$ **then**
7: $M_1 \leftarrow x,\ \lambda_u \leftarrow \lambda,\ \lambda \leftarrow (\lambda_u + \lambda_l)/2$
8: **else**
9: $M_2 \leftarrow x,\ \lambda_l \leftarrow \lambda,\ \lambda \leftarrow \min\{2\lambda, \lambda_u\}$
10: **return** M_1, M_2, λ

Algorithm 2 LocalSearch

Input: Graph $G = (V_a \cup V_g, E)$ with edge costs c and weights a, bound B

1: Get λ, M_1 and M_2 such that $a(M_1) \leq B \leq a(M_2)$ using Algorithm 1 or [23]
2: $M' \leftarrow M_1 \oplus M_2$
3: **for** connected components Y in M' **do**
4: **if** M' has one connected component **then**
5: **return** M_1, M_2, λ
6: **else**
7: **if** $a(M_1 \oplus Y) \leq B$ **then** $M_1 \leftarrow M_1 \oplus Y$
8: **else** $M_2 \leftarrow M_1 \oplus Y$
9: Remove Y from M'

Corollary 1. *If each UAV can recharge at at least N_a different UGV recharging locations, there is a polynomial time algorithm that finds a feasible scheduling M satisfying Constraints* (4) *and* (3) *such that $a(M) \leq B + 2a_{max}$ and $c(M) \leq \texttt{opt} + 3c_{max}$.*

Given an $\epsilon \in (0, 1]$, Algorithm 3 guesses $\lceil 1/\epsilon \rceil$ edges of highest cost in the solution and removes these edges and corresponding UAV and UGV vertices from the problem instance. It then finds a solution on the resulting problem instance that satisfies Lemma 4. This procedure results in the following bicriteria approximation.

Theorem 1. *For a given $\epsilon \in (0, 1]$, there is a $(1+\epsilon, 2)$-bicriteria approximation algorithm for Problem 1, i.e., we get a solution x such that $c(x) \leq (1+\epsilon)\texttt{opt}$, and $\sum_i x_{ij} \leq 2$, $\forall\ j \in V_g$.*

Since the bicriteria approximation algorithm *may* violate one of the constraints of the problem (one of the UGVs may need to recharge up to $d+1$ UAVs at a time), in practice, we can use Algorithm 3 to solve the problem and if the solution violates the constraint, we can use the feasible solution returned by Algorithm 2.

Algorithm 3 RENDEZVOUSSEARCH

Input: Graph $G = (V_a \cup V_g, E)$ with edge costs c and weights a, bound B, $\epsilon \in (0, 1]$

$M_H^* \leftarrow$ Guess $p = \lceil \frac{1}{\epsilon} \rceil$ edges of highest cost in M^*
$E \leftarrow E \setminus \{e : c(e) \geq \min_{e' \in M_H^*} c(e')\}$
for $(i, j) \in M_H^*$ **do**
    $V_a \leftarrow V_a \setminus \{\mathcal{V}_r : i \in \mathcal{V}_r\}$
    $V_g \leftarrow V_g \setminus \{j\}$
$B \leftarrow B - a(M_H^*)$
Find M_1, M_2 and λ from Algorithm 2
$Z \leftarrow M_1 \oplus M_2$
for edge z_i in $Z = \{z_0, z_1, \ldots, z_{k-1}\}$ **do**
    if $z_i \in M_1$ **then** $\alpha_i = w_\lambda(z_i)$
    if $z_i \in M_2$ **then** $\alpha_i = -w_\lambda(z_i)$
Find z_i such that $\sum_{j=i}^{i+h} \alpha_{i \pmod k} \leq 0$ for all h
Find the longest sequence Z'' starting from z_i such that $a(M_1 \oplus Z'') \leq B$
if Last edge of Z'' is not in M_2 **then**
    Remove the last edge of Z''
$M_L \leftarrow M_1 \oplus Z''$
if M_L has two edges connected to $\mathcal{V}_r$ for some r **then**
    Remove one of those edges from M_L
if M_L has two edges (i_1, j) and (i_2, j) for a $j \in V_g$ **then**
    if $\exists$ free j' such that $a(i_\ell, j') \leq a(i_\ell, j)$ **then**
        $M_L \leftarrow M_L \setminus \{(i_\ell, j)\} \cup \{(i_\ell, j')\}$
return $M_L \cup M_H^*$

IV. EXPERIMENTAL RESULTS

In this section, we first present a qualitative example of the persistent monitoring mission. Next, we study how the value of risk tolerance influences the recharging behaviors and task performances of the UAVs. Then, we compare the performance of our scheduling strategy with a baseline (greedy strategy). Moreover, we empirically evaluate the performance of the proposed heuristic algorithm. All experiments are conducted on a PC with the i9-8950HK processor unless specified otherwise. The baseline solver is Gurobi 9.5.0.

A. Experimental Setup

We consider a team consisting of two UAVs and two UGVs. The task routes $\mathcal{T}_a$ and $\mathcal{T}_g$ used in the problem can be either generated jointly by some task planners similar to those in [10], [30] or can be generated separately by different task planners. The UAV and UGV move at $v_a = 9.8$ m/s and $v_g = 4.5$ m/s respectively based on the field test data collected and used in our previous work [9]. The recharging process (swapping battery) takes 100s. The UAV and UGV need to persistently monitor the task nodes on the route. We apply our recharging strategy in a receding horizon fashion: every two minutes, the UAVs-UGVs team solves the RRRP problem to decide the UAVs' recharging schedule for the next $T = 2500$ seconds. For each UAV, the current position will be the first node when we construct the bipartite graph. If some UAV is on a detour, we do not replan until the UAV has finished its detour.

We consider two sources of stochasticity in the energy consumption model of UAVs: weight and wind velocity contribution to longitudinal steady airspeed. The energy consumption model of the UAV is the same as that in [9]

TABLE I
STATISTICAL RESULTS FOR UAVS

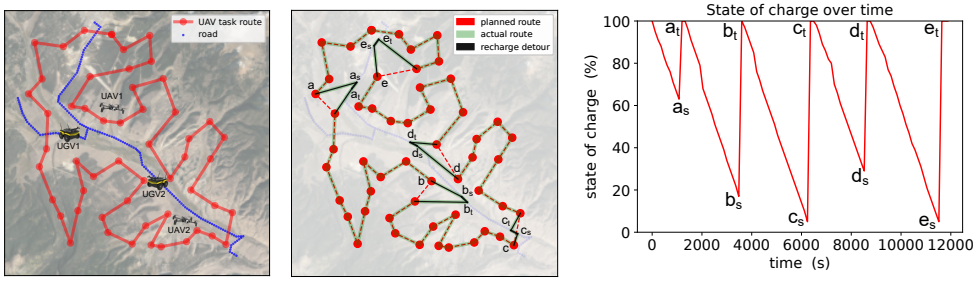
UAV data	$\rho = 0.01$	$\rho = 0.1$	$\rho = 0.3$
Mean time before failure (s)	39660	27600	24360
Avg. travel time overhead	19.7 %	18.5 %	17.8 %
Avg. # of task nodes visited	158	110	105
Avg. # of rendezvous per T	1.4	1.3	1.3

B. Simulation Results

a) Qualitative Example: The input of the problem consists of UAV task nodes, and nodes of the road network (Figure 4a). Figure 4b shows one tour route of one UAV when the system executes the proposed strategy in a receding horizon fashion. The UAV monitors the task route persistently. When the UAV reaches node a, it doesn't move forward to its next task node (connected through a dashed red line). Instead, the new schedule is to rendezvous with UGV at a_s and takes off from the UGV at a_t, and then go to its next task node. Similarly, the UAV will rendezvous with the UGV when it is close to nodes b, c, d and e. Subscriptions s and t denote the start and the terminal of the recharging. A sample of the history of the state of charge (SOC) is shown in Figure 4c. We can observe in Figure 4c that the UAV's recharging strategy is more than a simple rule, such as for example get recharged when the SOC is below 50 % and may get recharged at various values of SOC.

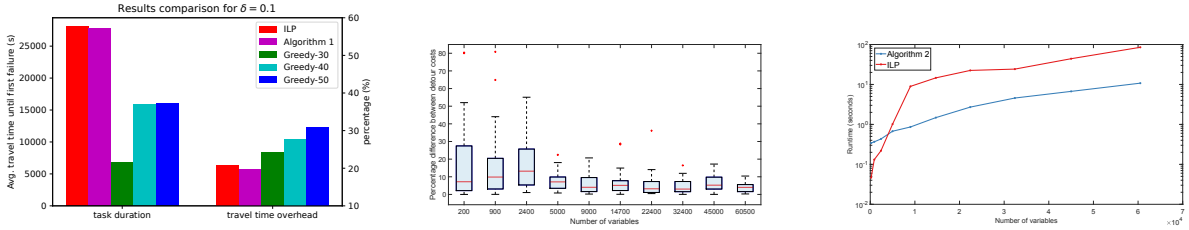
b) Effect of Risk Tolerance: We study how various risk tolerances influence the strategy (see Table I). We set the risk tolerance ρ to be 0.01, 0.1, and 0.3 and use four metrics to quantify the performance of the strategy:
(1) Mean time before the first failure: The time before the first UAV runs out of charge and needs human intervention. *(2) Travel time overhead*: (`actual travel time`$-$`task time`)/`task time`, where `task time` is the travel time of the route without any recharging. A lower overhead is desired since it accounts for the delay in visiting task nodes. *(3) Average number of task nodes visited:* by the UAV before its first failure. Similar to (1), a higher number reflects a better strategy. *(4) Average number of rendezvous per planning horizon T*. If this number is too large, it suggests that the UAV takes too many recharging detours, which should be avoided.

In general, we observe that when the risk tolerance is set to be smaller, the mean time before the first failure will be longer. Similarly, the travel time overhead and the average number of rendezvous per planning horizon will be greater, which implies the UAV spends more portion of flight time in the recharging detours.



(a) (b) (c)

Fig. 4. A qualitative example to illustrate how UAV and UGV rendezvous with each other solving the RRRP. The risk tolerance is set to be $\rho = 0.1$ in this case study. Subscripts s and t denote the start and the terminal of the recharging process. (a) The input of the RRRP problem includes the UAV and UGV tasks and the road network. (b) One sample tour of UAV 1 when it persistently monitors the route. (c) One sample history of SOC for UAV 1.



(a) (b) (c)

Fig. 5. Quantitative results. (a) Comparisons of the RRRP scheduling and the greedy strategies with $\rho = 0.1$. ILP and Algorithm 1 refer to the solution returned by the ILP solver and Algorithm 1 by solving RRRP. (b) % increase in the objective function of Algorithm 2 as compared to an ILP solver. The boxplot shows the result of 20 experiments for each problem size. (c) Comparison of run times of Algorithm 2 and ILP solver. Note, y-axis is logarithmic.

c) Comparison of Algorithm 1 with Baseline: We compare our strategy with a greedy baseline. The greedy policy chooses to rendezvous whenever the state-of-charge drops below a set value. We consider three set values 30%, 40%, and 50%, and the corresponding strategies are denoted as *Greedy-30*, *Greedy-40*, and *Greedy-50*. We set $\rho = 0.1$. The first observation is that Algorithm 1 achieves close performances in both metrics compared to that of ILP. Second, as shown in Figure 5a, our strategy (obtained by both ILP and Algorithm 1) can achieve a longer travel time before the first failure on average (left group) and a relatively lower travel time overhead (right group), which implies that our strategy will avoid unnecessary rendezvous. Moreover, Algorithm 1 has marginally better travel time overhead than ILP. We conjecture that this is due to horizon effect: since we are solving the problem in a receding horizon fashion, an optimal solution within a horizon may still be suboptimal over the entire duration.

d) Scalability of Algorithm 2: We also compare the performance of Algorithm 2 with an ILP solver empirically. We used Algorithm 2 for comparison instead of Algorithm 3 as Algorithm 2 and the ILP always returns a feasible solution, making the objective value comparison fair. The ILP solver used for this set of experiments is `intlinprog` function from MATLAB, and Algorithm 2 was also implemented in MATLAB for a fair comparison. Since ILP solves Problem 1 optimally, the cost returned by Algorithm 2 is at least that of the ILP solver. The percentage difference in the objective function values for different problem sizes is shown in Figure 5b. For each problem size, represented by the number of edges or variables, twenty random problem instances were created and the boxplot of resulting objective value difference is shown in the plot. On average, among all the instances, Algorithm 2 was within 15% of the optimal solution. Note that the performance of Algorithm 2 improves as the number of variables increases.

Figure 5c shows the average runtime of Algorithm 2 and the ILP solver. Note that the y-scale is logarithmic. For smaller problem instances, both the algorithms solved the problem within a second, with ILP being faster, however, as the number of variables increases, ILP becomes much slower, with the runtime for ILP being up to seven times more than that of Algorithm 2 for 60500 variables. Note that there may be other solvers for ILP that have better run time, but since ILP is NP-complete, the exponential gap between run times is likely to continue as the number of variables increases.

V. Conclusion

We introduced a multi-robot version of the risk-aware UAV-UGV recharging problem. We showed how to model this as a matching problem with knapsack constraints. We presented three algorithms that build on each other. In particular, Algorithms 1 and 2 solve the original problem and our empirical analyses show that yield efficient solutions compared to ILP and other strategies. Algorithm 3, on the other hand, yields a theoretical bicriteria approximation guarantee. We validate our formulation and the proposed algorithm in a persistent monitoring application. In our current formulation, we assume that the task routes of vehicles are given. In future work, one direction that we will explore is to propose efficient and scalable routing algorithms to generate task routes for vehicles.

REFERENCES

[1] D. Kingston, R. W. Beard, and R. S. Holt, "Decentralized perimeter surveillance using a team of uavs," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1394–1404, 2008.

[2] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 16–25, 2006.

[3] C. Lin, "A vehicle routing problem with pickup and delivery time windows, and coordination of transportable resources," *Computers & Operations Research*, vol. 38, no. 11, pp. 1596–1609, 2011.

[4] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.

[5] J. Diaz, E. Corrales, Y. Madrigal, D. Pieri, G. Bland, T. Miles, and M. Fladeland, "Volcano monitoring with small unmanned aerial systems," in *Infotech@ Aerospace 2012*, 2012, p. 2522.

[6] Y. Sung, D. Dixit, and P. Tokekar, "Environmental hotspot identification in limited time with a uav equipped with a downward-facing camera," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 264–13 270.

[7] H. Dhami, K. Yu, T. Xu, Q. Zhu, K. Dhakal, J. Friel, S. Li, and P. Tokekar, "Crop height and plot estimation for phenotyping from unmanned aerial vehicles using 3d lidar," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2643–2649.

[8] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic uav and ugv system for precision agriculture," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1498–1511, 2016.

[9] G. Shi, N. Karapetyan, A. B. Asghar, J.-P. Reddinger, J. Dotterweich, J. Humann, and P. Tokekar, "Risk-aware uav-ugv rendezvous with chance-constrained markov decision process," *2022 IEEE 61th Annual Conference on Decision and Control (CDC)*, 2022.

[10] K. Yu, A. K. Budhiraja, S. Buebel, and P. Tokekar, "Algorithms and experiments on routing of unmanned aerial vehicles with mobile recharging stations," *Journal of Field Robotics*, vol. 36, no. 3, pp. 602–616, dec 2018.

[11] N. Mathew, S. L. Smith, and S. L. Waslander, "Multirobot rendezvous planning for recharging in persistent tasks," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 128–142, 2015.

[12] P. Maini, K. Sundar, M. Singh, S. Rathinam, and P. Sujit, "Cooperative aerial–ground vehicle route planning with fuel constraints for coverage applications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 3016–3028, 2019.

[13] P. Maini and P. Sujit, "On cooperation between a fuel constrained uav and a refueling ugv for large scale mapping applications," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2015, pp. 1370–1377.

[14] K. Sundar and S. Rathinam, "Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 1, pp. 287–294, 2013.

[15] L. Liu and N. Michael, "Energy-aware aerial vehicle deployment via bipartite graph matching," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2014, pp. 189–194.

[16] H. Y. Jeong, B. D. Song, and S. Lee, "Truck-drone hybrid delivery routing: Payload-energy dependency and no-fly zones," *International Journal of Production Economics*, vol. 214, pp. 220–233, 2019.

[17] K. Yu, J. M. O'Kane, and P. Tokekar, "Coverage of an environment using energy-constrained unmanned aerial vehicles," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[18] A. Karak and K. Abdelghany, "The hybrid vehicle-drone routing problem for pick-up and delivery services," *Transportation Research Part C: Emerging Technologies*, vol. 102, pp. 427–449, 2019.

[19] H. Li, J. Chen, F. Wang, and M. Bai, "Ground-vehicle and unmanned-aerial-vehicle routing problems from two-echelon scheme perspective: A review," *European Journal of Operational Research*, vol. 294, no. 3, pp. 1078–1095, 2021.

[20] N. Karapetyan, K. Benson, C. McKinney, P. Taslakian, and I. Rekleitis, "Efficient multi-robot coverage of a known environment," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1846–1852.

[21] N. Karapetyan, J. Moulton, J. S. Lewis, A. Q. Li, J. M. O'Kane, and I. Rekleitis, "Multi-robot dubins coverage with autonomous surface vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2373–2379.

[22] R. Ravi and M. X. Goemans, "The constrained minimum spanning tree problem," in *Scandinavian Workshop on Algorithm Theory*. Springer, 1996, pp. 66–75.

[23] A. Berger, V. Bonifaci, F. Grandoni, and G. Schäfer, "Budgeted matching and budgeted matroid intersection via the gasoline puzzle," *Mathematical Programming*, vol. 128, no. 1, pp. 355–372, 2011.

[24] N. Nigam and I. Kroo, "Persistent surveillance using multiple unmanned air vehicles," in *2008 IEEE Aerospace Conference*. IEEE, 2008, pp. 1–14.

[25] A. B. Asghar, S. L. Smith, and S. Sundaram, "Multi-robot routing for persistent monitoring with latency constraints," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 2620–2625.

[26] S. Hari, S. Rathinam, S. Darbha, K. Kalyanam, S. Manyam, and D. Casbeer, "Efficient computation of optimal uav routes for persistent monitoring of targets," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2019, pp. 605–614.

[27] A. B. Asghar, G. Shi, N. Karapetyan, J. Humann, J.-P. Reddinger, J. Dotterweich, and P. Tokekar, "Risk-aware resource allocation for multiple uavs-ugvs recharging rendezvous," *http://arxiv.org/abs/2209.06308*, 2022.

[28] D. S. Johnson and M. R. Garey, *Computers and intractability: A guide to the theory of NP-completeness*. WH Freeman, 1979.

[29] A. Schrijver, *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[30] S. G. Manyam, K. Sundar, and D. W. Casbeer, "Cooperative routing for an air–ground vehicle team—exact algorithm, transformation method, and heuristics," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 1, pp. 537–547, 2019.