

Path Planning for a Jumping Rover Team with a Charging Station in Multi-waypoints Visiting Missions

Myungjin Jung¹, Kai Chuen Tan¹, and Ran Dai^{2,*}

¹ Mechanical and Aerospace Engineering Department, The Ohio State University, Columbus, OH, 43210, United States

^{2,*}School of Aeronautics and Astronautics, Purdue University, 701 W. Stadium Ave., West Lafayette, IN, 47907, United States

Correspondence*:

Ran Dai

randai@purdue.edu

2 ABSTRACT

This paper demonstrates an innovative group of robots, consisting of jumping rovers and a charging station, improved traversability and extended energy endurance when traveling to multiple target locations. By employing different jumping rovers with distinct energy consumption characteristics and jumping capabilities, we focus on searching for the most energy-efficient path of each jumping rover in a multi-waypoints visiting mission with obstacles. As jumping rovers can jump onto or over some obstacles without navigating around them, they have the potential to save energy by generating alternative paths to overcome obstacles. Moreover, due to the energy demands for the multi-waypoints mission and the limited battery capacity, a charging station is considered to provide extra energy for enhanced endurance during the mission. We first apply a refined rapidly-exploring random tree star (RRT*) algorithm to find energy-efficient paths between any two target locations. Then, the genetic algorithm (GA) is applied to select the most profitable combination of paths to visit all targets with energy constraints. Finally, we verify the improved mobility and energy efficiency in both virtual simulation and experimental tests using a group of customized jumping rovers with a charging station and the proposed path planning and task allocation method.

Keywords: Motion Planning, Task Allocation, Jumping Rover, Multi-Robot Systems, Rapidly-Exploring Random Tree, Charging Station

19

1 INTRODUCTION

Unmanned ground vehicles (UGVs) have been used extensively for exploration in unknown or dangerous environments where humanity is not able to access. The terrain in the exploration mission may consist of diversified features, such as flats, cliffs, and slopes. For most of the exploration missions, a UGV is selected according to the specific feature of the overall terrain. For example, wheeled robots have been used to travel to the target location in environments with flat terrain. However, for terrain with complicated geometry, the exploration mission is performed by particular types of locomotive, such as caterpillar tracked robots or unmanned aerial vehicles (UAVs). Although general wheeled UGVs show extended operational time in most cases, they cannot overcome obstacles, such as areas with large gaps or terrain with high elevations.

28 UAVs, on the other hand, are subject to atmospheric effects and are governed by more stringent safety or
29 operational requirements. Considering the limitations of traditional UAVs and UGVs, we propose using a
30 team of wheeled robots with jumping capabilities for multi-waypoints visiting missions with obstacles.
31 The focus is to develop an optimal path planning algorithm for the particular robot team to search for the
32 energy-efficient path in the assigned mission.

33 Compared to a traditional wheeled vehicle that has to avoid obstacles, the jumping capability provides
34 flexibility on path planning, as a jumping rover can decide whether to avoid or jump over/onto the obstacles.
35 The jumping motion is achieved by deforming the robot's parts, e.g., wheels (Ye et al., 2018), or activating
36 their jumping apparatus (Mizumura et al., 2017). In the space industry, hopping robots have been applied
37 in planetary exploration missions (Morad et al., 2018; Hockman and Pavone, 2020). Existing studies on
38 the control of a jumping robot focus on generating precise jumping motion, e.g., legged motion with
39 specific speed and torque (Ding and Park, 2017) and motion control of a jumping robot that uses a tail as its
40 jumping mechanism (Iwamoto and Yamamoto, 2015). The motion planning approach for a jumping robot
41 has been developed that prioritizes safety and minimizes the cost of jumping by finding an optimal landing
42 position (Ushijima et al., 2017). And obstacles are treated as a point that cannot be used as a suitable
43 landing surface in the literature. In this paper, we consider specific dimensions of an obstacle that can be
44 landed by a jumping rover.

45 Another concern that restricts the robot motion in traveling missions is the limited battery capacity,
46 especially in long-term operations. Therefore, to extend the operational time in traveling missions, charging
47 stations have been considered to provide extra power within the mission area. In ideal scenarios, a charging
48 station will allow rovers to work persistently (Mathew et al., 2015; Kingry et al., 2017). When a charging
49 station is considered in the multi-waypoints visiting mission, the path planning problem is more challenging
50 as we need to consider paths from/to a charging station in addition to the paths between the target locations.
51 Furthermore, charging decisions, e.g., which rover should be charged and when to charge them (Michaud
52 and Robichaud, 2002), need to be determined based on the assigned mission and energy consumption rate.
53 These charging stations are required to automatically connect to the inlet of an autonomous robot. Work in
54 (Behl et al., 2019) utilized a camera to detect the relative position between a charging plug and a robot.
55 Another work in (Barzegaran et al., 2017) utilized wireless power transmission for electric vehicles. Due to
56 the size and weight limitation of the jumping rover, we cannot apply those charging methods to connect
57 with the jumping rover. Instead, magnetic connectors are introduced to connect two end effectors of a
58 station and a rover.

59 Many types of route optimization algorithms have been developed to solve path planning problems
60 involving multiple robots, e.g., dynamic programming (Kok et al., 2010; Ou and Sun, 2010), minimum
61 spanning tree algorithms (Pettie and Ramachandran, 2002), Tabu search (Archetti et al., 2006), ant
62 colony optimization(Abousleiman et al., 2017), and particle swarm optimization (Belmecheri et al., 2013).
63 A general multi-waypoints traveling mission performed by a team of robots can be formulated as the
64 well-known multi traveling salesperson problem (mTSP) and then solved via the mixed-integer linear
65 programming (MILP) algorithm, which is not directly applicable to the path planning problem considered
66 in this paper. One reason is the involvement of a charging station, which requires determining visiting
67 sequences and time to a charging station. The other reason is the inclusion of obstacles, which complicates
68 the traditional mTSP. The development of motion planning methods has led to various techniques being
69 used, including sampling-based algorithms such as probabilistic road map (PRM) (Kavraki et al., 1996)
70 and rapidly exploring random tree (RRT) (LaValle, 1998; LaValle and Kuffner Jr, 2001). A variant of RRT,
71 named RRT*, iteratively searches for an optimized solution, e.g., shortest path (Karaman et al., 2011). In

72 this paper, after obtaining a path using the basic RRT* algorithm, path refinement techniques are proposed
73 to search for an optimal path that meets the jumping rover's motion constraint.

74 Based on our prior work of jumping rovers (Tan et al., 2020), this work extends the jumping rover team
75 with a charging station and proposes new algorithms for path planning and task allocation. We first find
76 energy-efficient path segments between any two target locations by a refined RRT* algorithm, where an
77 obstacle can be treated as a possible pathway if a jumping rover is able to jump over it. If there are obstacles
78 between two targets, the refined RRT* determines whether avoiding obstacles or jumping over obstacles
79 is more energy efficient. We then utilize the genetic algorithm (GA) to search for the most profitable
80 combination of path segments to visit all targets, as well as path segments to/from the charging station to
81 satisfy energy constraints. When considering the charging station and energy constraints of each jumping
82 rover, it makes the path planning of the multi-waypoints traveling mission much more challenging, which
83 requires a new formulation and path planning method incorporating the charging function and energy
84 constraints. Compared to our prior work in (Tan et al., 2020), the contribution of this paper includes the
85 following points: (1) a new formulation of the multi-waypoints traveling mission of a robot team integrating
86 the charging function and energy constraints. (2) a path planning algorithm with refined paths that consider
87 more complicated geometries of an obstacle in both rolling and jumping motion, (3) introducing GA to
88 determine both visiting and charging sequences, and (4) design and construction of a charging system that
89 automatically docks with the jumping rovers.

90 The paper is organized as follows: §2 introduces the robot team and problem formulation, §3 describes
91 the development of path planning and task allocation algorithm. The simulation and experimental results
92 are presented in §4. Finally, we address the conclusions and future work in §5.

2 ROBOT TEAM MODEL AND PROBLEM STATEMENT

93 2.1 Jumping Rovers and Charging Station

94 There are multiple requirements to verify the improved mobility and energy efficiency of a jumping
95 rover team in real-world operations. First and foremost, the rolling motion must produce ground-based
96 locomotion and the jumping motion is required to produce vertical displacement. Second, it must record
97 energy consumption from both types of motion to validate the proposed algorithm properly. The wheeled,
98 jumping rovers are based upon a Parrot Jumping Sumo robot chassis (jum, 2019). The rover is made out of
99 different materials, including acrylic, nylon, aluminum, and 3D printed parts because the rover must be
100 lightweight to achieve an acceptable jumping height. The center of mass sits on the force vector of the
101 jumping mechanism to ensure the rover does not flip after actuation. In addition, “whiskers” in the front
102 enable the rover to descend from the top of obstacles without flipping over. Each jumping rover has a fixed
103 jumping height determined by the stiffness of the compressed spring. Two jumping rovers with different
104 energy consumption rates and jumping heights are demonstrated in Figure 1A. The left rover in Figure 1A
105 can jump higher and consumes more energy than the other one.

106 The charging station is built to automatically charge the two rovers, as shown in Figure 1B. Each rover
107 has a unique charging port and the charging station has two charging port arms. The charging service arms
108 can rotate to capture the port of the rover. Moreover, magnet tips are used for the charging station outlets to
109 secure the docking of rovers with the charging station via magnetic forces. We use the “HX-A3 Hexfly
110 Lipo Battery Charger” for each rover. Depending on the battery level of the rover, the charging time can
111 be different for the same amount of charged energy. We also use voltage and current sensors to measure
112 how much energy is provided to the rover from the chargers and these measurements are transmitted to the
113 mission control computer via a wireless modem.

The Parrot Jumping Sumo robot that can be purchased off the shelf is a manually operated rover. The overall system is shown in Figure 1C. We modified the original model with a Micro controller Unit (MCU) and communication system to achieve autonomous operation, including a customized jumping mechanism. The jumping rovers are controlled wirelessly through serial communication with a mission control computer. The commands are then relayed to a motor driver that controls the rotational speed of the wheel motors. Signals for the jumping mechanism are applied directly from the MCU to a servo motor. Each rover and the charging station have voltage and current sensors to measure its power consumption or charging power. Waypoints for the planned path are output from the MATLAB simulation and followed by the physical UGV through a 3D space coordinate measured by a motion capture system.

2.2 Rover's Kinematics and Control

In order to make the rovers follow the desired trajectories, we use the basic kinematics of the two-wheeled robot (Chen et al., 2008; Ichihara and Ohnishi, 2006). The rolling and jumping motion working mechanism is shown in Figure 2A and 2B. For the rolling motion, its coordinate is represented by X and Y . The center point velocity of the rover is denoted by v . Then, \dot{X} and \dot{Y} are expressed as,

$$\begin{aligned}\dot{X} &= v \cos \theta \\ \dot{Y} &= v \sin \theta\end{aligned}\quad (1)$$

where θ is the heading angle of the rover and the angular velocity ω is the heading angle changing rate, denoted by $\dot{\theta}$. Then, we can express the velocity terms in a matrix form, expressed as

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2)$$

The relationship between the translational and angular velocity of the rover and the translational velocities (v_l, v_r) of each wheel can be expressed by

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{L}{2} & -\frac{L}{2} \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \quad (3)$$

where L is the width of the rover, and ω_l and ω_r are the angular velocity of the left and right wheel, respectively. Then, we can derive the relationship between the velocity and the angular velocity of the wheel.

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} \cos \theta & \frac{R}{2} \cos \theta \\ \frac{R}{2} \sin \theta & \frac{R}{2} \sin \theta \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \quad (4)$$

This relationship shows that the rover's motion is non-holonomic. For the rolling motion, the four differential-drive primitives of a Balkcom-Mason curve are considered to simplify its motion with straight line and zero-radius rotation. When the rover changes its heading, time and energy consumption for rotation are negligible. Therefore, only line segment driving is considered when calculating the energy consumption.

Due to the limited size of rovers and weight limit for the jumping motion, there are no encoders implemented in the rovers to measure the rotational speed of their wheels. Therefore, to adjust the rovers' heading angle, their wheel speed is under control such that the rovers can maintain their heading angle when following a straight line. A proportional and derivative controller is applied to maintain a

143 desired heading angle, expressed as

$$u = K_p e(\theta) + K_d \frac{de(\theta)}{dt} \quad (5)$$

144 where u is the differential input for the rotational speed of a rover, $e(\theta)$ is the error of angle between a
145 desired heading angle and the current heading angle, and k_p and k_d are the proportional and derivative
146 gains, respectively.

147 For the jumping motion, we need to determine the maximum jumping height based on the energy
148 conservation principle. By calculating the spring potential energy, the maximum jumping height can be
149 determined by,

$$h_{\max} = \frac{k_{\text{spring}}((l_r - l_c) \cos \theta_c)^2}{2W} \quad (6)$$

150 where k_{spring} is the spring constant, l_r is the spring length when it is released, l_c is the spring length when it
151 is compressed, θ_c is the angle between the rover and the vertical axis, and W is the weight of the rover. The
152 characteristics of energy consumption and jumping capability of two rovers are shown in Table 1, where
153 UGV 1 jumps higher than UGV 2 and consumes about four times as much energy as UGV 2 for every
154 jump.

155 2.3 Problem Statement

156 The objective for a team of UGVs is to visit a set of targets using the most energy-efficient route, where one
157 stationary charging station is provided for the power supply. The targets are denoted as $T = T_{\{1,2,\dots, m\}}$,
158 where m is the total number of visiting targets. The charging station is denoted by S . All UGVs start at
159 T_1 and rendezvous at the same destination. The m targets can be located anywhere in the mission area,
160 including the top of obstacles. In addition to the specified targets, we consider n obstacles randomly
161 scattered in the mission area, denoted by $O = O_{\{1,2,\dots, n\}}$. Each obstacle is a rectangular prism with a flat
162 surface and given dimensions. These obstacles are treated as solid objects that cannot be passed through
163 or moved. They will not overlap with each other; however, their boundaries can intersect adjacent ones
164 such as O_4 , O_5 , and O_6 shown in Figure 3. The UGVs are indexed as $z = 1, 2, \dots, p$. UGVs in the team can
165 reach different jumping heights. To generate a feasible jumping path, we need to consider the kinematics
166 of two-wheeled vehicles and the obstacles' geometry. The jumping path associated with an obstacle is
167 positioned perpendicular to the border of the obstacle. Each target point can only be visited by one UGV to
168 reduce the cost associated with the travel, except for the initial and the destination targets. The number of
169 visiting a charging station is not limited.

170 Consider a single UGV that operates with two operational modes. One is the rolling mode that rotates
171 motors as a general ground vehicle, and the other mode is the jumping mode. The UGV stops at the jumping
172 position and compresses and releases its spring mechanism to jump. With a constant velocity during
173 straight forward rolling and constant angular speed during zero radii rotation for UGV z , $z = 1, \dots, p$, the
174 energy consumption rate of the jumping rover in straight forward motions is denoted by P_z^l . The energy
175 consumption for straight forward and jumping motion is much higher than the one required for rotation,
176 which makes the energy consumption for rotating motion negligible. In addition, the passive energy drawn
177 from the vehicle's electronic components, such as the MCU, is a fixed value and denoted by P_z^a . The
178 height of the jumping motion for each UGV is held constant, denoted by h_z , with fixed energy expenditure,
179 denoted by J_z , associated with the corresponding jumping rover $z = 1, \dots, p$. The energy consumption for

180 each jumping rover traveling from target i to target j , $i, j = 1, \dots, m$, $i \neq j$, is determined by

$$c_{ij,z} = (P_z^l + P_z^a)t_{ij,z}^l + J_z N_{ij,z}, \quad (7)$$

181 where $t_{ij,z}^l$ is the time duration along the rolling motion between targets i and j , $N_{ij,z}$ is the overall number
182 of jumps for UGV z , $z = 1, \dots, p$, between targets i and j . The time used to travel between two targets can
183 be calculated by the velocity of each rover and the distance between the targets, where constant velocity
184 is used for each rover. By adding the energy consumption of all jumping rovers along path segments
185 connecting all targets, we can find the overall UGV team energy usage during the visiting mission.

186 In this paper, each UGV is confined to the energy consumption constraint due to the limited battery
187 capacity. Therefore, a UGV needs to visit the charging station before its stored energy becomes exhausted.
188 Assuming the energy initially stored in the battery for each UGV is $E_{0,z}$, a UGV's stored energy will
189 reach the initial amount every time it is charged at the station. Before approaching the charging station, a
190 designated UGV should have sufficient energy to drive to the charging station. During the charging process,
191 $\Delta E_{l,z}$ amount of energy is provided from the station at the l th charging sequence to UGV z such that the
192 stored energy will reach the initial amount.

193 The path planning problem for the multi-waypoints visiting mission can be represented by a complete
194 graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ which consists of a set $\mathcal{V} = T_{\{1,2,\dots,m\}} \cup S$ as graph vertices and an edge set \mathcal{E} connecting
195 any two target points. The associated edge cost for $(i, j) \in \mathcal{E}$ is determined by the energy consumption
196 amount required to travel from target i to j , where $i, j = 1, 2, \dots, m$, S and $i \neq j$. In addition, $x_{ij,z}$ is a
197 binary variable that is determined by

$$x_{ij,z} = \begin{cases} 1, & \text{edge } (i, j) \in \mathcal{E} \text{ will be visited by UGV } z \\ 0, & \text{edge } (i, j) \in \mathcal{E} \text{ will not be visited by UGV } z \end{cases}$$

198 All vehicles start at the first target, denoted as T_1 , and end at the same destination point T_k , where the
199 index k , $1 < k \leq m$, is unknown and will be determined by the path planning algorithm. T' is a set
200 of targets excluding the initial target such that $T = T_1 \cup T'$. Let V_z be the collection of target indices,
201 excluding the charging station, visited by UGV z , where elements of V_z are sorted according to the visiting
202 sequences. The number of targets, excluding the charging station, visited by UGV z is denoted as n_z ,
203 which is also the length of V_z . Then, the path planning problem for the energy-efficient multi-waypoints
204 visiting mission can be formulated as

$$\min \sum_{z=1}^p \sum_{j \in T', j \neq i} \sum_{i \in T \cup S} c_{ij,z} x_{ij,z} \quad (8)$$

$$\text{s.t.} \quad \sum_{z=1}^p \sum_{j \in T' \cup S} x_{1j,z} = p \quad (9)$$

$$\sum_{z=1}^p \sum_{j \in T' \cup S} x_{jk,z} = p, \quad j \neq k, \quad k \in T' \quad (10)$$

$$\sum_{z=1}^p \sum_{i \in T \cup S} x_{ij,z} = 1, \quad j \in T', \quad i \neq j, \quad j \neq k \quad (11)$$

$$\sum_{z=1}^p \sum_{j \in T \cup S} x_{ij,z} = 1, \quad i \in T', \quad i \neq j, \quad i \neq k \quad (12)$$

$$\sum_{i,j \in V_z} (x_{iS,z} - x_{Sj,z}) = 0, \quad z = 1, \dots, p \quad (13)$$

$$\sum_{i,j \in V_z(1, \dots, l_z)} (c_{iS,z}x_{iS,z} + c_{ij,z}x_{ij,z} + c_{Sj,z}x_{Sj,z} - x_{iS,z}\Delta E_{i,z}) \leq E_{0,z} - E_{min,z}, \\ l_z = 2, \dots, n_z - 1, \quad z = 1, \dots, p \quad (14)$$

205 where (8) is the cost function representing the overall energy consumption for the jumping rover team to
 206 visit all the assigned targets, as well as the charging station. Constraints (9) and (10) indicate that all p
 207 UGVs start at target T_1 and end at target T_k . Constraints (11) and (12) require that one target should only
 208 be visited once, except for the starting and ending targets and the charging station. Constraint (13) indicates
 209 that at least one pair of edges exists to connect the charging station with two adjacent targets i and j in
 210 the set of V_z if UGV z visits the charging station. A UGV may visit the charging station multiple times
 211 according to its energy consumption characteristics. Constraint (14) specifies that the energy in the battery
 212 of each UGV is required to maintain above $E_{min,z}$ for all the time, where $x_{iS,z}\Delta E_{i,z}$ indicates that if UGV
 213 z gets charged after visiting target i , it will gain $\Delta E_{i,z}$ to reach the initial energy amount, denoted as $E_{0,z}$,
 214 and $V_z(1, \dots, l_z)$ represents the first l_z elements in the set V_z .

3 PATH PLANNING AND TASK ALLOCATION ALGORITHM

215 The multi-waypoints visiting problem formulated in (8)-(14) also needs to consider the n obstacles in the
 216 mission area. First, a refined RRT* algorithm is proposed to determine the optimized paths between any
 217 two target points. Then, a customized GA is applied to search for optimal sequences to visit all the target
 218 points, as well as the charging station when it is necessary for a UGV to charge its battery.

219 3.1 Refined RRT*

220 When applying the RRT algorithm (LaValle, 1998; LaValle and Kuffner Jr, 2001) to search for a feasible
 221 path, each tree begins at an initial point, x_{init} , and attempts to make a connection between the origin and
 222 a random point, x_{rand} , in a specified area. The length of the connection is dictated by an established unit
 223 length, Δx . The connection of the random point is made with the nearest point in the tree x_{near} to a new
 224 point, x_{new} , which can be reached. Basically, a unit vector multiplied by a scalar, Δx , in the direction of
 225 the random point. This configuration is added to the result data, and a new connection is made without
 226 violating the collision constraint. The process is repeated for a number of desired iterations, K , and the
 227 selected points in the sequence are saved in R . If the final destination point, x_{dest} , is provided, then x_{dest}
 228 connects to their nearest random point of a generated tree without violating the collision constraint before
 229 the function terminates. While effective in finding a solution with a fast speed, RRT cannot guarantee that
 230 solutions are efficient in terms of the length of the tree path from x_{init} to x_{dest} . Thus, RRT*, the optimized
 231 version of RRT, takes each point in a tree, finds the points within a radius of each point, and replaces
 232 existing edges with the most efficient path without violating the collision constraints.

233 However, RRT* is restricted to optimization within a radius around a vertex or within a “neighborhood”.
 234 Due to this limitation, RRT* may not provide a smooth solution that is traversable between target locations.
 235 To compensate for the deficiency, we propose a refined RRT* method with the process shown in Algorithm
 236 1 from line 15 to line 23. We aim to shorten the final path through the refinement process and make it more
 237 energy efficient by excluding unnecessary tree segments through the smoothing and elimination process.

238 The smoothing process selects two random points on two distinct edges that link RRT* vertices and creates
 239 a smoother path that does not violate the collision constraints. The smoothing process continues until it
 240 reaches the desired number of smoothing iterations, U . The elimination process checks all edges from the
 241 initial to the final vertices. If an edge does not collide with any obstacles, the second vertex together with
 242 the edge that links to it will be eliminated, and the third vertex will become the second one. The elimination
 243 process removes unnecessary vertices.

Algorithm 1 : Refined RRT* Algorithm

```

1: function RRT_STAR( $x_{\text{init}}$ ,  $x_{\text{dest}}$ ,  $\Delta x$ ,  $K$ )
2:    $o \leftarrow \text{Obstacle Search}()$ 
3:   for  $q \leftarrow 1, 2^o$  do
4:      $R.\text{initialize}(x_{\text{init}}, x_{\text{dest}})$ 
5:     for  $k \leftarrow 1, K$  do
6:        $x_{\text{rand}} \leftarrow \text{Random\_State}()$ 
7:        $x_{\text{near}} \leftarrow \text{Nearest\_Point}(x_{\text{rand}}, R)$ 
8:        $x_{\text{new}} \leftarrow \text{New\_Config}(x_{\text{rand}}, R, \Delta x)$ 
9:       if CollisionFree then
10:         $x_{\text{near}} \leftarrow \text{Near}(R_t, x_{\text{new}}, |vertices|)$ 
11:         $x_{\text{min}} \leftarrow \text{Parent}(x_{\text{near}}, x_{\text{nearest}}, x_{\text{new}}, x_{\text{dest}})$ 
12:         $R \leftarrow \text{Rewire}(R, x_{\text{near}}, x_{\text{nearest}}, x_{\text{new}}, x_{\text{dest}})$ 
13:      end if
14:    end for
15:    for  $w \leftarrow 1, U$  do
16:       $p_1, p_2 \leftarrow \text{Random\_Edges\_Vertex}(R)$ 
17:      if CollisionFree then
18:         $R \leftarrow \text{Path\_Smoothing}(p_1, p_2, R)$ 
19:      end if
20:    end for
21:    while CollisionFree do
22:       $R \leftarrow \text{Vertices\_Elimination}(R)$ 
23:    end while
24:    if Jump Search then
25:       $R \leftarrow \text{Adjust\_Jump\_position}(R)$ 
26:    end if
27:  end for
28:   $R \leftarrow \text{cal\_min}$ 
29:  return  $R$ 
30: end function
```

244 With the jumping capability, the jumping rovers can decide whether to jump over or avoid an obstacle.
 245 Jumping over an obstacle leads to obstacle elimination and then changes a non-traversable path into a
 246 feasible one. Therefore, additional feasible path segments will be created for the multi-waypoints visiting
 247 mission with the jumping option. If the number of obstacles lying on the straight line path between targets
 248 T_i and T_j is n_f , there will be 2^{n_f} number of feasible paths to connect T_i and T_j . For example, in Figure
 249 4, two obstacles are lying on the straight line connecting targets T_1 and T_2 . There are four feasible paths
 250 between T_1 and T_2 .

251 The results from the refined RRT* algorithm are affected by the number of samples. To improve the
 252 performance of the refined RRT* to obtain a result closer to the global optimal solution, it is necessary to
 253 examine the effects on the convergence rate and cost value of the refined RRT* results when selecting a
 254 different number of samples. Using one part of Scenario 3 to find the collision avoidance path between

Targets T_2 and T_7 as an analysis example, when changing the number of samples from 300 to 10000 and running the refined RRT* 50 times for each sample number, the convergence rate is shown in Figure 5A. Therefore, at least 4000 samples are required to guarantee convergence of the refined RRT* algorithm. According to the convergence analysis, we assign the minimum number of samples for the refined RRT* algorithm according to the effective area including every two targets in the mission. Moreover, the cost value under different numbers of samples is compared. For the original RRT* algorithm, it is obvious that the increased number of samples leads to shorter paths with a reduced cost value. For the refined RRT* algorithm proposed in this paper, it always obtains a near-optimal solution once it converges, as shown in Figure 5B. With these analysis results, 4000 samples are assigned when calculating paths between every two targets in the refined RRT* algorithm, which guarantees convergence and the converged result is a near-optimal solution.

3.2 Path Generation for Grouped Obstacles

Section 3.1 considers the cases where obstacles are separated from each other. This section shows the generation of traversable path segments for obstacles that are grouped together. Using the scenario in Figure 3 as an example, obstacles O_4 , O_5 , and O_6 are grouped together and the elevation of O_5 is higher than the other two adjacent neighbors.

Under such type of scenarios, we need to determine directions to approach and depart the grouped obstacles as well as the routes traveling on top of them. We first approximate every obstacle as a combination of unit blocks. For example, O_4 and O_6 in Figure 6A are composed of three-unit blocks. Moreover, the grouped obstacles are surrounded by the same unit blocks in four directions, denoted as W , S , E , N . Based on the geometry approximation of the grouped obstacles and their surrounding areas, an adjacency matrix is established to represent the adjacency relationship between an obstacle and a surrounding area, as well as the relationship between any two obstacles in the group. For example, the adjacency matrix, denoted as A_d , for the case in Figure 6B is shown in Figure 6C.

In general, for a group with n_g obstacles, the size of its adjacency matrix is $(n_g + 4) \times (n_g + 4)$, where $A_d(i, j) = 1$ indicates elements i and j share a borderline. According to the adjacency matrix, new feasible paths are determined in the next step to overcome grouped obstacles. By permutation of the grouped obstacle indices and their surrounding areas that are adjacent to each other, all feasible paths approaching and departing the grouped obstacles from one surrounding area to the other, as well as the path segments on top of the obstacles, can be determined. Similar to the cost value computation for the path segment in Section 3.1, the cost value is assigned to each segment of the newly generated paths. Then, the one with the minimum combined cost is selected for a specific jumping rover to travel over the grouped obstacles.

3.3 Optimal Visiting Sequences

When all feasible routes with and without jumping options are generated between any two targets using the methods described in Sections 3.1 and 3.2, a customized GA is applied to search for the optimal sequences to visit all the target points, as well as the charging station when it is necessary. Three parts of the chromosome are included in the customized GA. In Figure 7, the first part represents the target visiting sequences for all UGVs, denoted as $[V_1(1), \dots, V_1(n_1), V_2(1), \dots, V_2(n_2), \dots, V_p(1), \dots, V_p(n_p)]$. The charging sequences for each UGV are involved in the second part of the chromosome in Figure 7. Associated with each element in the set of V_z , a binary variable is assigned and set as one if a corresponding UGV goes to the charging station after visiting a specific target point, otherwise it is set as zero. Thus, the number of chromosomes in the second part is the same as the length of the first part. The third part of the chromosome represents the number of targets visited by each vehicle, denoted as $[n_1, n_2, \dots, n_p]$ with

298 $\sum_{z=1}^p n_z = m + 2p - 2$. For example, according to the chromosome in Fig 7, the target visiting sequences
 299 of UGV 1 is 1-3-2-4-5, and it will be charged after visiting target 2 and then continue to visit target 4 after
 300 charging. Then, the overall visiting sequences for UGV 1 is 1-3-2-S-4-5.

301 From the initial population, the tournament selection picks a random subset of the population and then
 302 chooses the best-fitted chromosome in the selected population set. Among the selected population, the
 303 crossover and mutation process will be applied based on the probability of the process. The crossover
 304 process only affects the first part of the chromosome, indicating each UGV's target visiting sequences. We
 305 introduce an ordered crossover rather than using a single or two-point crossover to avoid generating invalid
 306 solutions. Its child is generated by copying a random number of successive genes and its position from one
 307 parent. The remaining genes are implanted in the order of another parent. Then, the mutation process will
 308 affect the first and third parts by swapping one gen with another. These operations prevent the genes from
 309 being trapped in a local solution.

310 When generating offspring for the next generation, energy constraints and collisions between rovers are
 311 examined. Only collision-free offspring satisfying the energy constraints are chosen for the GA operations
 312 described above. Specifically, to satisfy the energy constraint, formulation in (14) is examined based on the
 313 new sequences including the charging sequences. Any chromosome that violates the energy constraint is
 314 abandoned and the process is repeated until the sequence satisfies the energy constraints. Next, we examine
 315 the collision-free constraints between any two rovers. From RRT, we can obtain time costs as well as energy
 316 costs when traveling between two targets. By examining a sequence from GA, we can determine whether
 317 one path intersects with another or if there are multiple paths within a rover's width. If those conflicting
 318 paths are assigned to any rovers, the time for corresponding UGVs at the intersection point is calculated. If
 319 there is an intersection in time, new paths will be generated by applying a square shape avoidance zone
 320 with the length of the square equals to the rover's width at the intersection point. We then run the GA
 321 algorithm again to determine the new sequence. This process is repeated until there is no collision among
 322 robots. Finally, the fitness value is calculated for every selected population. The fitness value is set the
 323 same as the cost function expressed in (8). The final solution will be determined if the ratio of the best
 324 solution in the selected population exceeds the rate of 97%.

325 For the GA algorithm, the crossover and mutation percentages involved in the evolving operations will
 326 affect the cost value. Using different combinations of crossover and mutation percentages ranging from
 327 65% to 95%, and 0.5% to 1%, respectively, we aim to find the best combination leading to fast convergence
 328 and an optimal solution. Figure 5C shows that most cases converge to a close-optimal solution. However,
 329 optimality is not guaranteed for any combination of crossover and mutation percentages. For example, if
 330 a low mutation percentage of 0.5% is used, its convergence speed is slower than other combinations in
 331 Figure 5C, associated with a higher cost value. For the worst-case with the crossover percentage of 95%
 332 and the mutation percentage of 0.5%, Figure 5C shows that the cost cannot be reduced within the first 20
 333 iterations and the cost value is much higher than the optimal one when it converges. Therefore, we choose
 334 the mutation percentage of 1% and the crossover percentage of 85% when solving this problem.

335 The entire path planning and task allocation algorithm is summarized into two steps, as shown in Figure
 336 8. In the first step, the optimized path segments with and without jumping options for each UGV are
 337 calculated with corresponding cost determined according to the UGV's characteristics. Next, the optimal
 338 visiting sequences from the initial target to the final target, as well as to the charging station, are determined
 339 using the customized GA.

4 SIMULATION AND EXPERIMENTS

340 4.1 Simulation

341 To verify the enhanced mobility and energy efficiency of the jumping rover team with a charging station,
 342 simulation examples in three scenarios are presented. For Scenario 1, two UGVs are required to visit nine
 343 target points without energy constraint, shown in Figure 9A and Figure 9B. There are seven obstacles, and
 344 two of them (O_2 and O_3) have higher elevations that can only be reached by UGV 1 with the jumping
 345 option. The two jumping rovers introduced in §2 will execute the mission. The characteristics of energy
 346 consumption and jumping capability are shown in Table 1, where UGV 1 jumps higher than UGV 2 and
 347 consumes about four times as much energy as UGV 2 for every jump.

348 In Scenario 1, we verify the improved energy efficiency and mobility by the UGVs' jumping capability,
 349 where the charging station is not considered. Figure 9A demonstrates the path planning and target allocation
 350 results without the jumping option, while Figure 9B shows the corresponding results with the jumping
 351 option. With the jumping option, the UGVs choose to jump over some obstacles instead of navigating
 352 around the obstacle to visit some targets in Figure 9B. As shown in Table 2, the jumping option leads to
 353 reduced energy consumption by about 18% and the byproduct is the reduced mission time around 71%.

354 In Scenario 2, we seek to verify the improved mission duration via the charging station when considering
 355 the energy constraint of each UGV. The jumping option is considered in Scenario 2 and some of the targets
 356 are placed on top of the obstacles. Without the energy limitation, the two UGVs' visiting sequences are
 357 (1, 6, 3) for UGV 1 and (1, 2, 7, 5, 4, 8, 3) for UGV 2, as shown in Figure 9C. Their overall energy
 358 consumption is 113.44 J (UGV 1: 47.20 J, UGV 2: 66.24 J). When considering the energy constraint (65
 359 J) for each UGV, their visiting sequences are (1, 6, 3, S, 8) for UGV 1 and (1, 2, 7, 5, 4, 8) for UGV 2, as
 360 shown in Figure 9D. The overall energy consumption is 78.62+65 J (UGV 1: 19.70+65 J, UGV 2: 58.92
 361 J), where 65 J is supplied by the charging station for UGV 1. The path segments before traveling to the
 362 charging station for the two scenarios are the same. Due to the energy constraint, UGV 2 ends its mission
 363 at target 8, and UGV 1 extends its duration by charging its battery after visiting target 3.

364 In Scenario 3, an extended area including 14 targets and four rovers is considered. We also pursue an
 365 enhanced mission duration under energy consumption constraint, where each UGV's battery capacity is
 366 140 J. UGV 3 is identical to UGV 1 used in Scenario 1, and UGV 4 is the same as UGV 2 used in Scenario
 367 1, in terms of performance features and power consumption characteristics. Without the battery capacity
 368 constraint, the visiting sequences of the robot team are (1, 2, 3, 14, 7) for UGV 1, (1, 6, 4, 11, 9, 13, 7)
 369 for UGV 2, (1, 8, 7) for UGV 3, and (1, 5, 12, 10, 7) for UGV 4, where the planned path for each UGV
 370 is shown in Figure 9E. Their overall net energy consumption is about 302 J (UGV 1: 92.04 J, UGV 2:
 371 161.42 J, UGV 3: 68.02 J, UGV 4: 145.48 J). When considering the energy constraints, the visiting
 372 sequences of the robot team are (1, 14, 3) for UGV 1, (1, 8, 4, 12, S, 10, 7, 5, 3) for UGV 2, (1, 2, 3) for
 373 UGV 3, and (1, 6, 9, 11, 13, 3) for UGV 4, as shown in Figure 9F. Their overall net energy consumption is
 374 270 J (UGV 1: 62.81 J, UGV 2: 30.83 + 140 J, UGV 3: 50.80 J, UGV 4: 126.01 J). To meet the energy
 375 constraint, UGV 2 is required to visit the charging station after visiting Target 12 such that it gains extra
 376 energy to resume the traveling mission.

377 4.2 Experiment Verification

378 The experimental tests are conducted based on Scenario 2 that considers both the jumping option and the
 379 energy constraint. As shown in Figure 10A, the charging station provides two outlets that can simultaneously
 380 charge two different UGVs. **We use the Vicon motion capture system to track the motion of UGVs when**
 381 **executing the planned mission, as shown in Figure 10B.** From the motion capture system, we can measure

382 the position and heading angle of rovers, the direction of the charging station, and the location of charging
383 connectors. Then, we can control the rovers' heading to follow the calculated paths. Throughout the
384 experimental tests, energy consumption data is recorded at a fixed frequency using the current/voltage
385 sensors.

386 Figure 11 shows the simulation and experimental results of two UGVs in Scenario 2 using a perspective
387 view. In Table 3, we compare the energy consumption amount in the simulation and experimental results
388 with and without the energy constraint in Scenario 2. The comparison of simulation and experimental data
389 verifies that the experimental results closely match the simulation results. The small differences between
390 the two types of results come from **the volume of the charging station, which is approximated as a target**
391 **point in the simulation. This attributes extra energy consumption in the experiment for the docking motion.**
392 **Furthermore, the detaching maneuver requires extra energy than nominal driving as the rover needs to**
393 **overcome the magnetic force that captures the plug.**

394 In the experimental tests, the two UGVs' energy consumption at each target point in Scenario 2 without
395 and with the energy constraint is shown in Figure 12A and Figure 12B, respectively. The total energy
396 consumption without the energy constraint is 109.72 J, and the one with the energy constraint is 72.74+65
397 J, where 65 J is supplied by the charging station. As UGV 1 consumes more energy in both rolling and
398 jumping motion, only three targets are allocated to UGV 1 and the remaining six targets are allocated to
399 UGV 2. Since target 6 is located at the top of an obstacle with a high elevation that can only be reached
400 by UGV 1, UGV 1 visits target 6 and terminates at target 3. When considering the energy constraint, the
401 energy consumption of UGV 1 exceeds the energy constraint after visiting target 3 and it does not have
402 sufficient energy to visit the remaining target points. Different from the case without energy constraint,
403 UGV 2 in this case does not have sufficient energy to visit (1, 2, 7, 5, 4, 8, 3) since visiting these target
404 points consumes 66.24 J, which is larger than the energy constraint, 65 J. Therefore, the charging station
405 extends the UGV mission endurance and thus has the potential to increase the number of targets in an
406 assigned mission area with obstacles. As indicated in Figure 12B, during the battery charging, the actual
407 energy consumption is dropping for UGV 1. Moreover, with a charging station involved in the robot team,
408 it does not require the UGVs to carry heavy batteries, which makes the vehicles lighter. A video file is
409 included as supplementary material for the experimental test.

5 CONCLUSION

410 This paper presents a path planning and task allocation method for a multi-waypoints visiting mission
411 using a group of unmanned ground vehicles with jumping capability and a charging station. The goal is
412 to search for energy-efficient routes to explore a mission area with obstacles. A refined RRT* method
413 and a customized genetic algorithm are developed to determine the energy-efficient path and visiting
414 sequences to the assigned targets points and the charging station. The simulation and experimental results
415 verify the advantages of jumping options and the involvement of a charging station in terms of improved
416 mobility, energy efficiency, and extended duration. Future studies will consider more complicated obstacle
417 geometries and investigate the mechanism of controllable jumping height.

AUTHOR CONTRIBUTIONS

418 MJ developed the path planning and task allocation algorithm, simulated virtual examples, constructed and
419 conducted the experimental tests, and wrote the original manuscript. KT programmed the RRT* algorithm
420 and built one of the jumping rovers. RD initiated the jumping rover team concept, refined the algorithm,
421 and revised the manuscript. All authors contributed to this article.

FUNDING

422 This work was supported in part by NSF grants ECCS-1815930.

DATA AVAILABILITY STATEMENT

423 The raw data and materials supporting this article will be made available by the authors.

REFERENCES

- 424 [Dataset] (2019). Parrot jumping sumo-races, slaloms, acrobatics, you can do it all. <https://www.parrot.com/us/minidrones/parrot-jumping-sumo>
- 425 Abousleiman, R., Rawashdeh, O., and Boimer, R. (2017). Electric vehicles energy efficient routing using
426 ant colony optimization. *SAE International Journal of Alternative Powertrains* 6, 1–14
- 427 Archetti, C., Speranza, M. G., and Hertz, A. (2006). A tabu search algorithm for the split delivery vehicle
428 routing problem. *Transportation science* 40, 64–73
- 429 Barzegaran, M., Zargarzadeh, H., and Mohammed, O. (2017). Wireless power transfer for electric vehicle
430 using an adaptive robot. *IEEE Transactions on Magnetics* 53, 1–4
- 431 Behl, M., DuBro, J., Flynt, T., Hameed, I., Lang, G., and Park, F. (2019). Autonomous electric vehicle
432 charging system. In *2019 Systems and information engineering design symposium (SIEDS)* (IEEE), 1–6
- 433 Belmecheri, F., Prins, C., Yalaoui, F., and Amodeo, L. (2013). Particle swarm optimization algorithm for
434 a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows. *Journal of
435 intelligent manufacturing* 24, 775–789
- 436 Chen, D., Bai, F., and Wu, L. (2008). Kinematics control of wheeled robot based on angular rate sensors.
437 In *2008 IEEE Conference on Robotics, Automation and Mechatronics* (IEEE), 598–602
- 438 Ding, Y. and Park, H.-W. (2017). Design and experimental implementation of a quasi-direct-drive leg
439 for optimized jumping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems
440 (IROS)* (IEEE), 300–305
- 441 Hockman, B. and Pavone, M. (2020). Stochastic motion planning for hopping rovers on small solar system
442 bodies. In *Robotics Research* (Springer). 877–893
- 443 Ichihara, Y. and Ohnishi, K. (2006). Path planning and tracking control of wheeled mobile robot considering
444 robots capacity. In *2006 IEEE International Conference on Industrial Technology* (IEEE), 176–181
- 445 Iwamoto, N. and Yamamoto, M. (2015). Jumping motion control planning for 4-wheeled robot with a tail.
446 In *IEEE/SICE International Symposium on System Integration (SII)*. 871–876
- 447 Karaman, S., Walter, M. R., Perez, A., Frazzoli, E., and Teller, S. (2011). Anytime motion planning using
448 the rrt. In *IEEE International Conference on Robotics and Automation*. 1478–1483
- 449 Kavraki, L. E., Svestka, P., Latombe, J.-C., and Overmars, M. H. (1996). Probabilistic roadmaps for path
450 planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation* 12,
451 566–580
- 452 Kingry, N., Liu, Y.-C., Martinez, M., Simon, B., Bang, Y., and Dai, R. (2017). Mission planning for
453 a multi-robot team with a solar-powered charging station. In *IEEE/RSJ International Conference on
454 Intelligent Robots and Systems (IROS)*. 5233–5238
- 455 Kok, A., Hans, E. W., Schutten, J. M., and Zijm, W. H. (2010). A dynamic programming heuristic
456 for vehicle routing with time-dependent travel times and required breaks. *Flexible services and
457 manufacturing journal* 22, 83–108
- 458 LaValle, S. (1998). Rapidly-exploring random trees: a new tool for path planning. *The annual research
459 report*

- 461 LaValle, S. M. and Kuffner Jr, J. J. (2001). Randomized kinodynamic planning. *The international journal*
462 *of robotics research* 20, 378–400
- 463 Mathew, N., Smith, S. L., and Waslander, S. L. (2015). Multirobot rendezvous planning for recharging in
464 persistent tasks. *IEEE Transactions on Robotics* 31, 128–142
- 465 Michaud, F. and Robichaud, E. (2002). Sharing charging stations for long-term activity of autonomous
466 robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. vol. 3, 2746–2751
- 467 Mizumura, Y., Ishibashi, K., Yamada, S., Takanishi, A., and Ishii, H. (2017). Mechanical design of a
468 jumping and rolling spherical robot for children with developmental disorders. In *IEEE International*
469 *Conference on Robotics and Biomimetics (ROBIO)*. 1062–1067
- 470 Morad, S., Kalita, H., and Thangavelautham, J. (2018). Planning and navigation of climbing robots in
471 low-gravity environments. In *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*.
472 1302–1310
- 473 Ou, W. and Sun, B.-G. (2010). A dynamic programming algorithm for vehicle routing problems. In
474 *International Conference on Computational and Information Sciences*. 733–736
- 475 Pettie, S. and Ramachandran, V. (2002). An optimal minimum spanning tree algorithm. *Journal of the*
476 *ACM (JACM)* 49, 16–34
- 477 Tan, K. C., Jung, M., and Dai, R. (2020). Motion planning and task allocation for a jumping rover team. In
478 *IEEE/RSJ International Conference on Robotics and Automation (ICRA)*
- 479 Ushijima, M., Kunii, Y., Maeda, T., Yoshimitsu, T., and Otsuki, M. (2017). Path planning with risk
480 consideration on hopping mobility. In *IEEE/SICE International Symposium on System Integration (SII)*.
481 692–697
- 482 Ye, C., Wang, B., Wei, B., and Tang, B. (2018). Modeling and analysis of a jumping robot with deforming
483 wheeled mechanism. In *IEEE International Conference on Mechatronics and Automation (ICMA)*.
484 980–985

FIGURE CAPTIONS

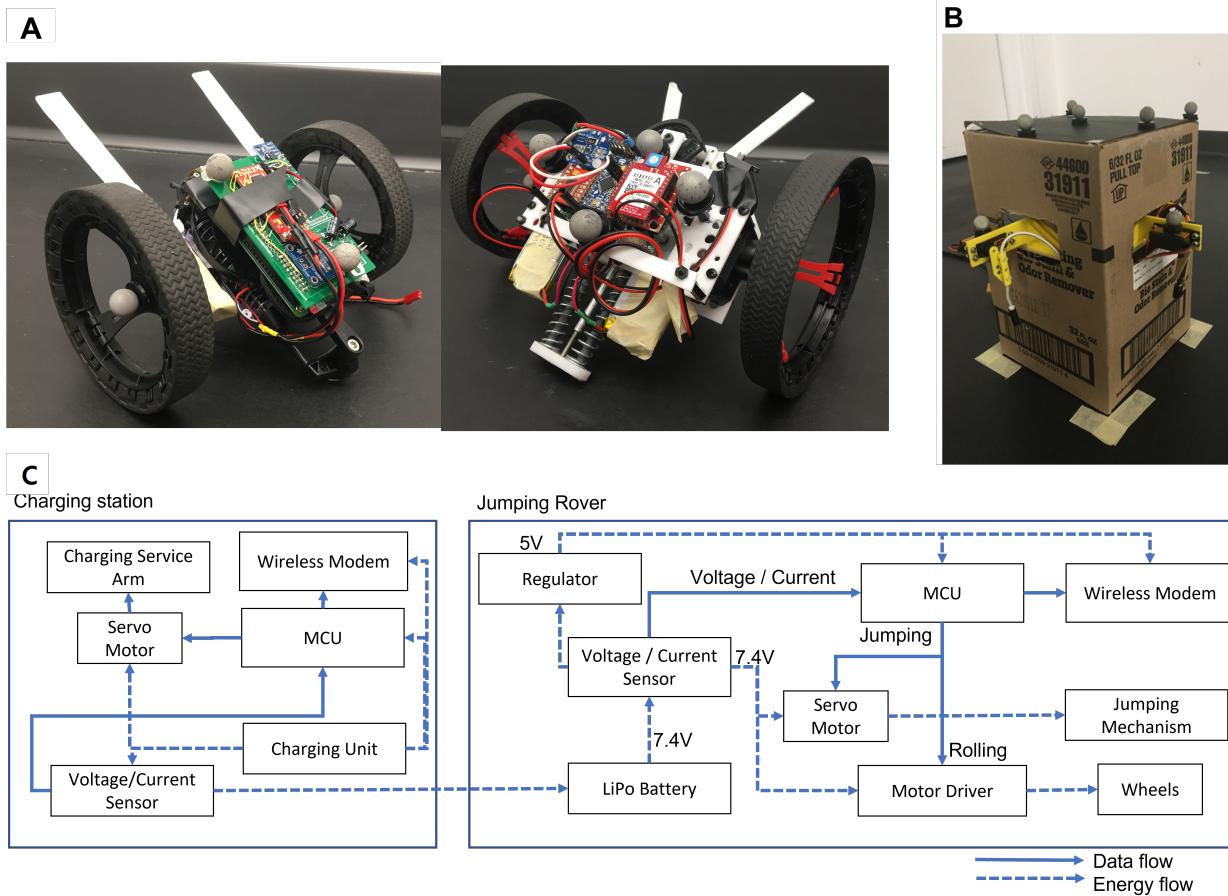


Figure 1. Jumping rovers and charging station (**A**): jumping rover 1 (UGV 1) (left) and jumping rover 2 (UGV 2) (right), **(B)**: charging Station **(C)**: overall data and power flow

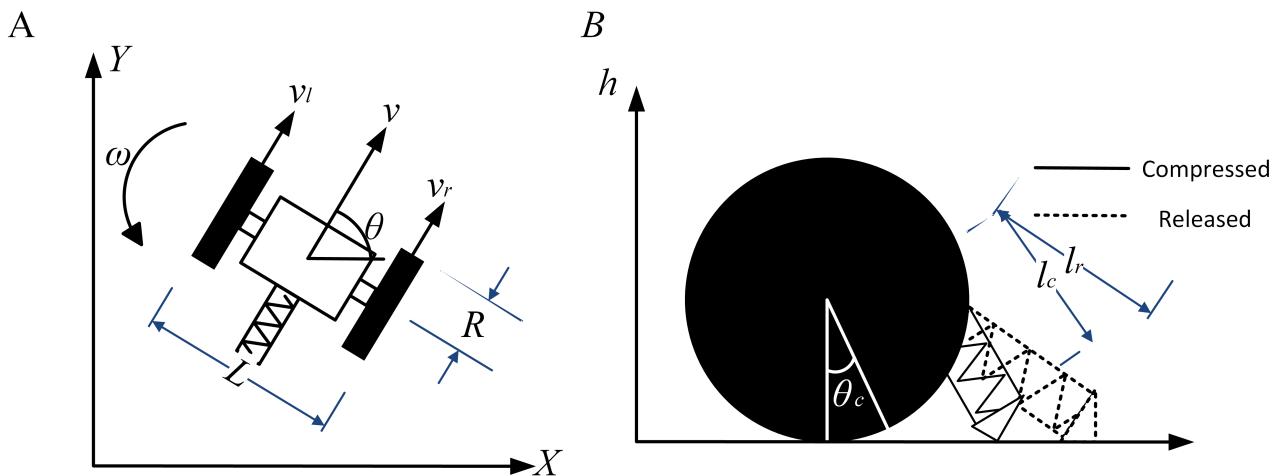


Figure 2. Jumping rover's motion (A) rolling motion, (B) jumping motion

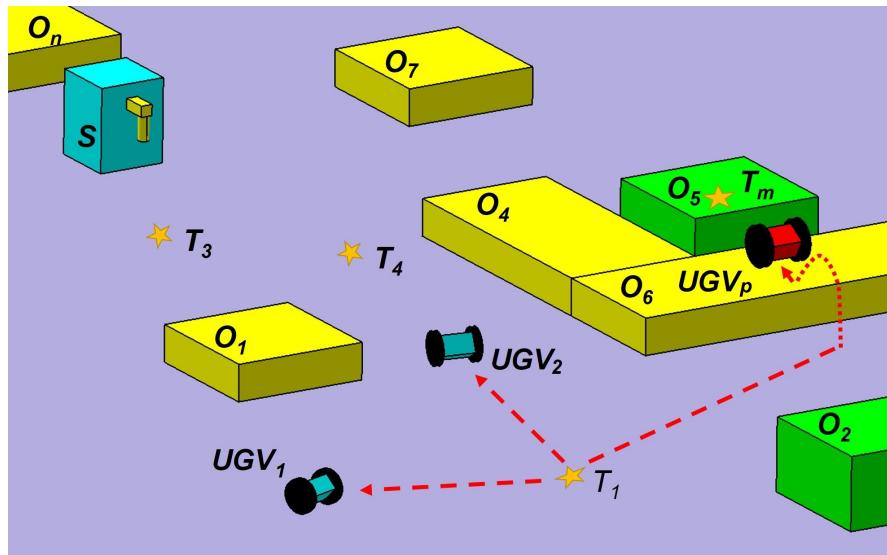


Figure 3. Mission overview

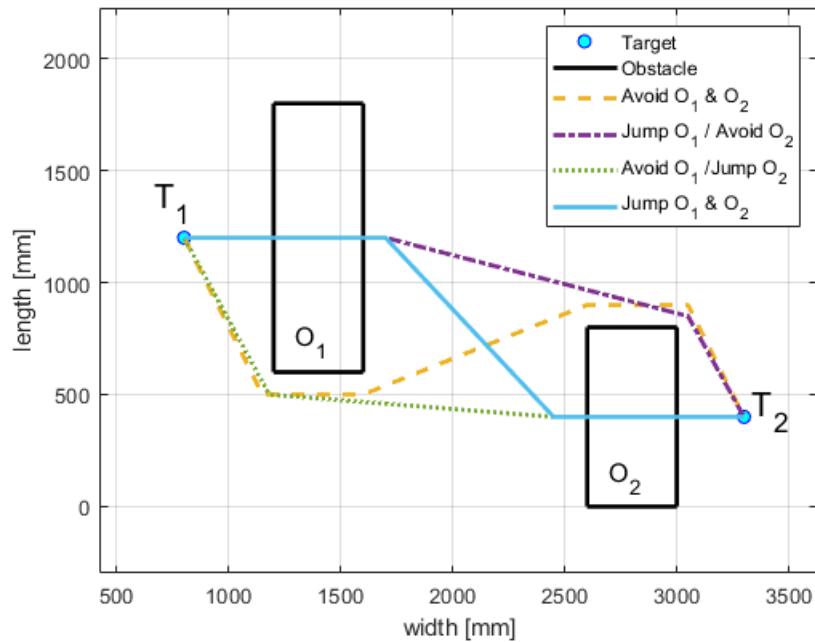


Figure 4. Traversable paths between targets 1 and 2 with jumping and avoiding options

	Rolling Energy [J/mm]	Jumping Energy [J/jump]	Jumping Height [mm]	Jumping Distance [mm]
UGV 1	0.0163	21.52	260	247.3
UGV 2	0.0061	5.49	150	226.5

Table 1. Energy and jumping characteristics of UGVs

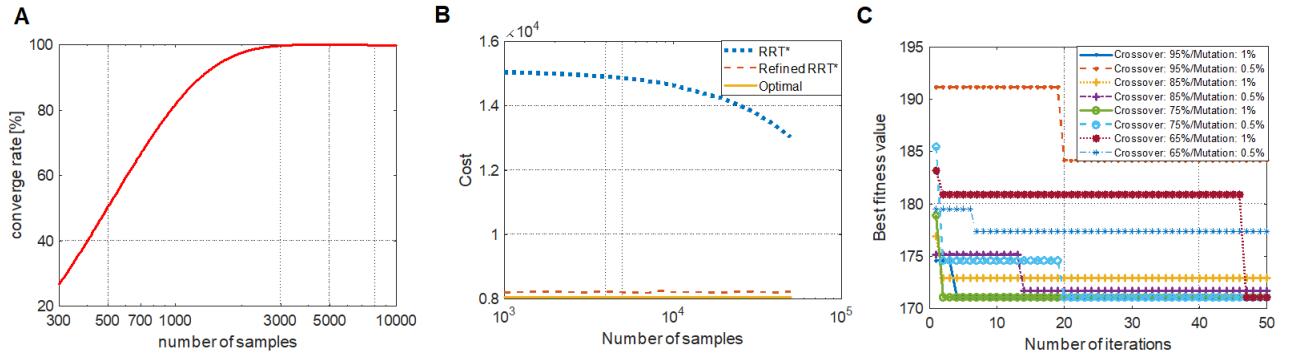


Figure 5. (A): RRT* convergence rate v.s. number of samples, (B): Cost value v.s. number of samples for RRT* and refined RRT*, (C) : GA performance under different combinations of crossover and mutation rates

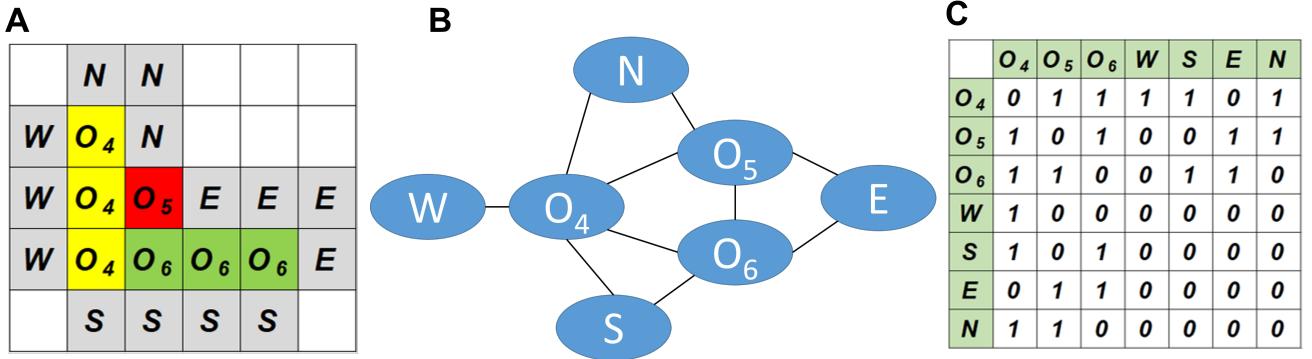


Figure 6. (A): geometry of grouped obstacles, (B): obstacles' adjacency relationship, (C) : an adjacency matrix.

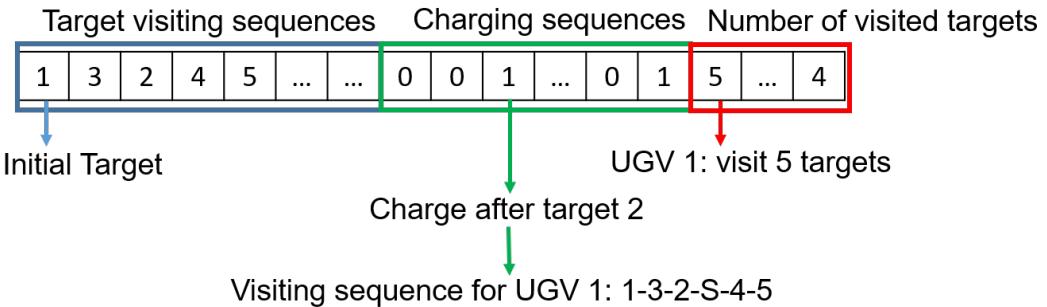


Figure 7. Chromosome representation for the UGV team with visiting and charging sequences

	without jumping		with jumping	
	Energy [J]	time [s]	energy [J]	time [s]
UGV 1	57.72	10.27	76.89	12.67
UGV 2	114.12	67.35	68.49	39.34
Overall	171.84	67.35	145.38	39.34

Table 2. Comparative results of energy consumption and mission time with and without jumping option in Scenario 1.

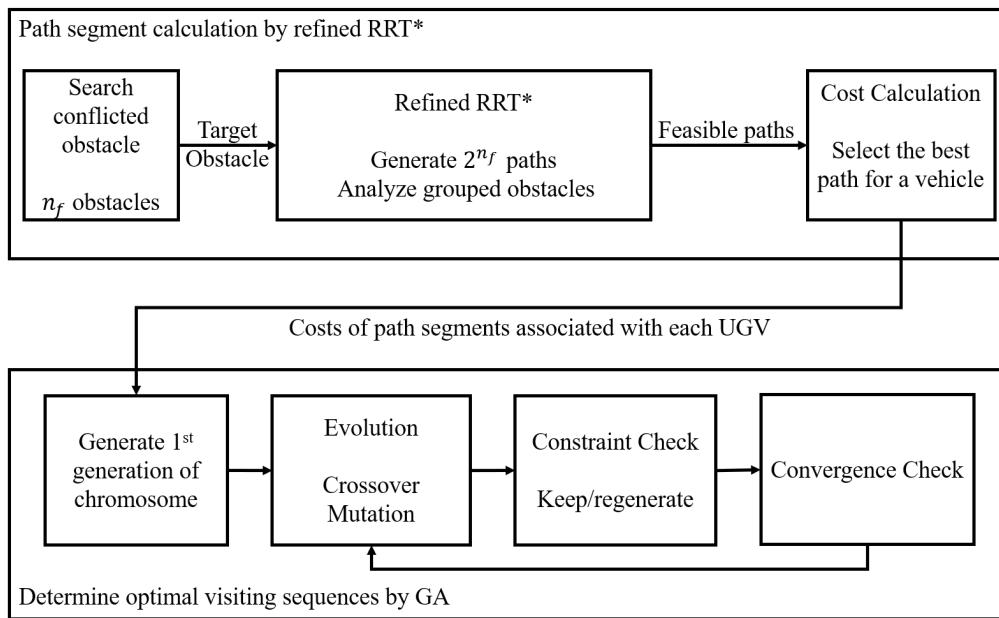


Figure 8. Two stages of path planning and task allocation

Energy [J]	without energy constraint			with energy constraint		
	Sim	Exp	error	Sim	Exp	error
UGV 1	47.20	45.15	4.34 %	19.70+65	12.46+65	8.35 %
UGV 2	66.24	64.57	2.59 %	58.92	60.28	2.31 %
Overall	113.44	109.72	3.28 %	78.62+65	72.74+65	5.14 %

Table 3. Energy consumption with and without energy constraint for Scenario 2 in simulation and experimental tests.

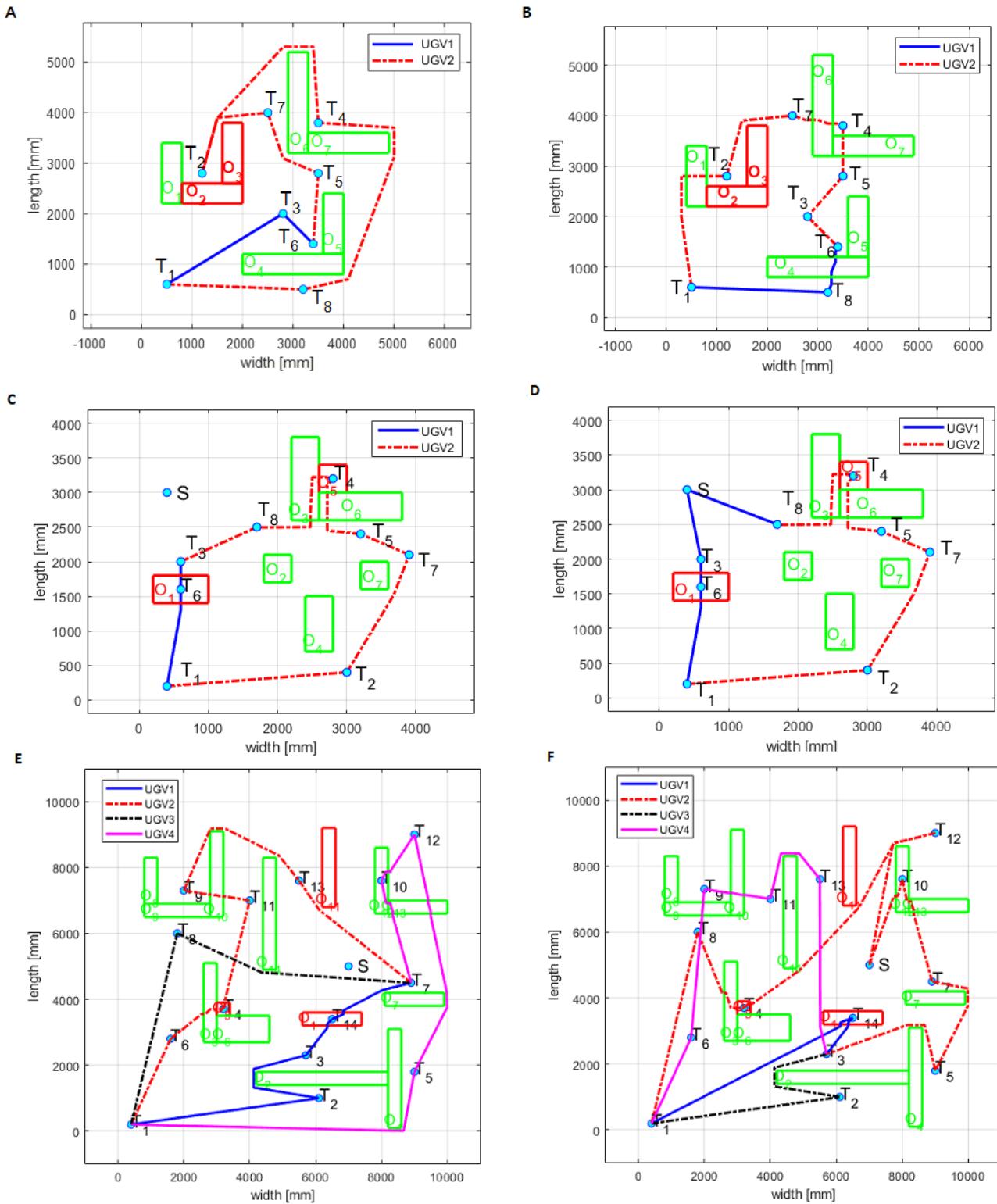


Figure 9. Path planning and target allocation simulation (**A**): without jumping option for scenario 1, (**B**): with jumping option for scenario 1 (**C**): without energy constraint for scenario 2, (**D**): with energy constraint for scenario 2 (**E**): without energy constraint for scenario 3, (**F**): with energy constraint for scenario 3

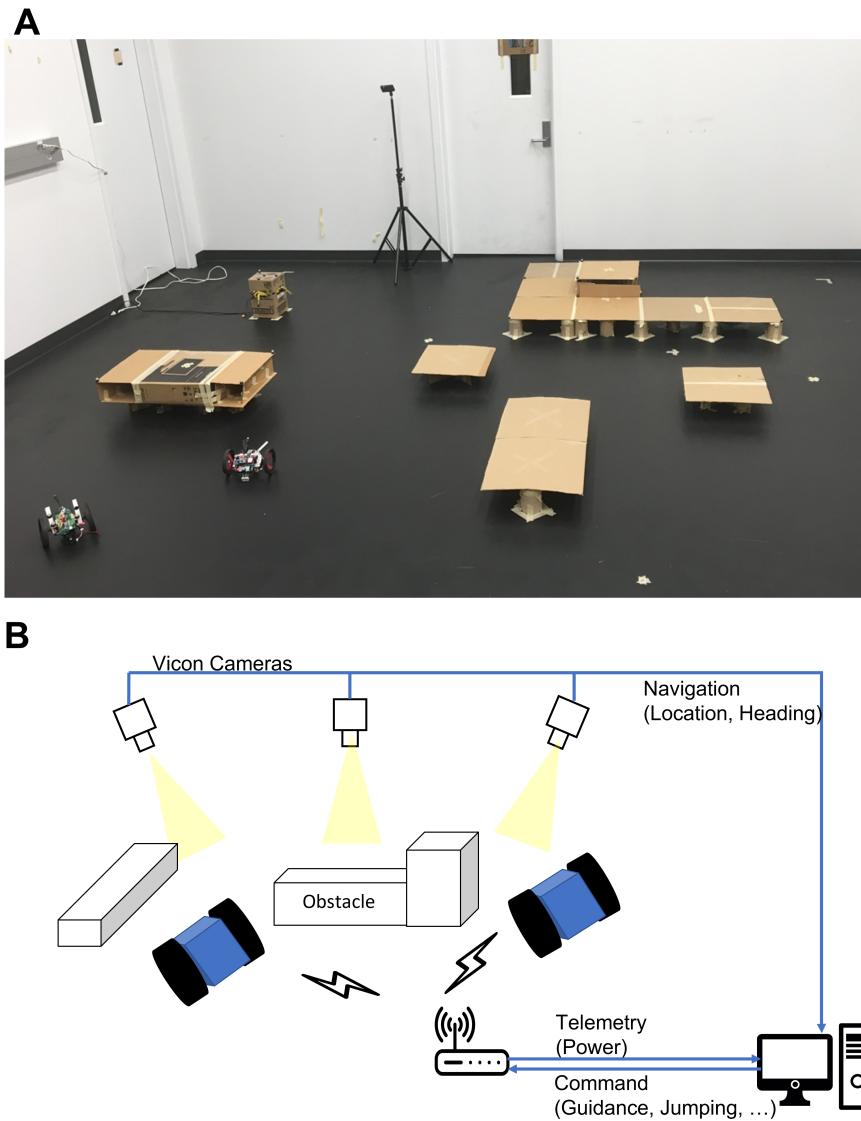


Figure 10. Experimental layout and charging station for Scenario 2, **(A)**: mission area, **(B)**: overall experimental configuration

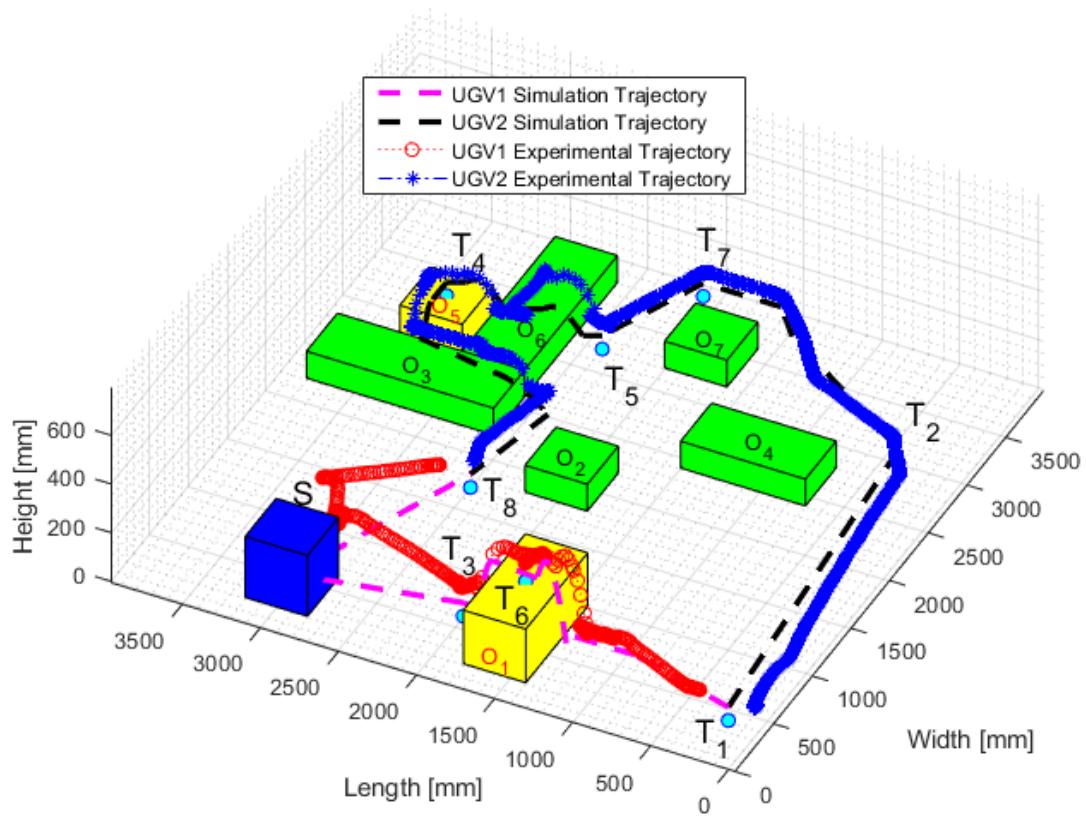


Figure 11. Simulation and experimental trajectories of UGVs in Scenario 2 with energy constraint

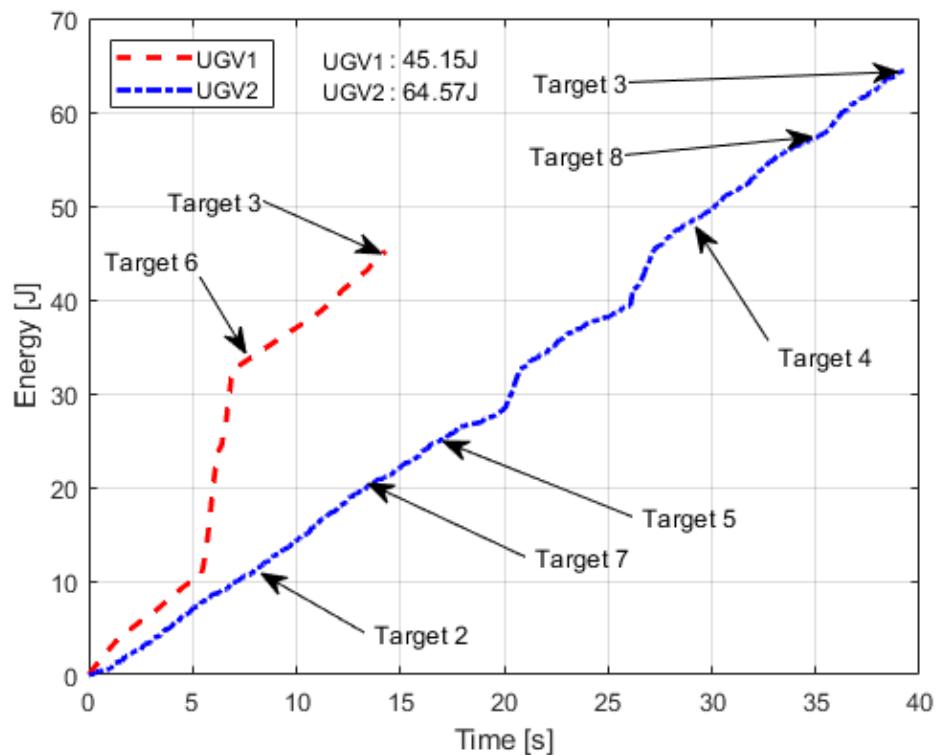
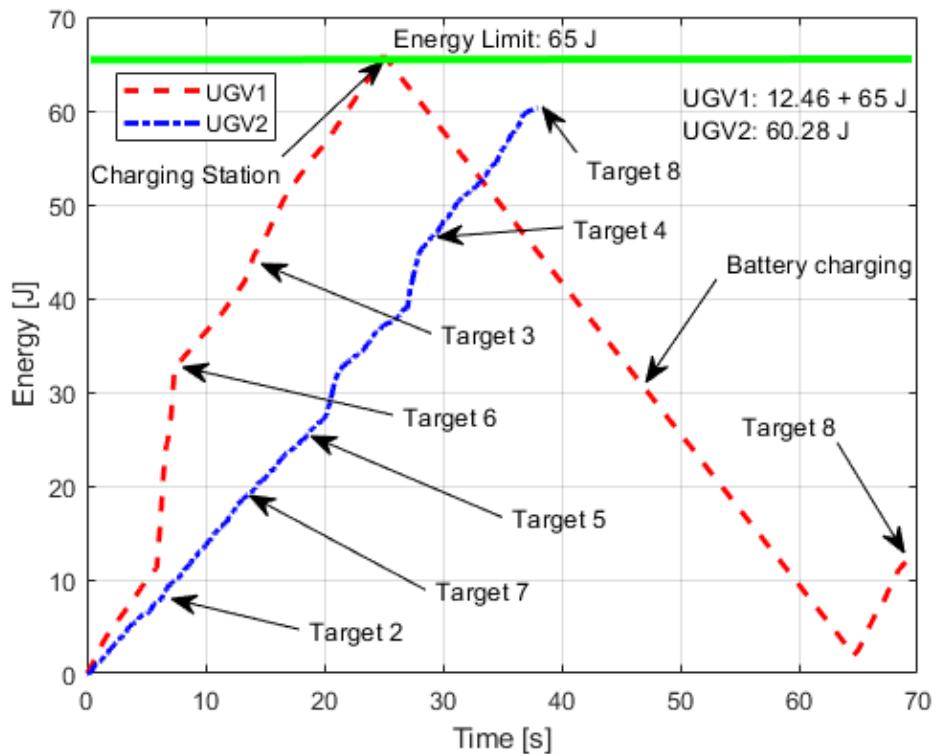
A**B**

Figure 12. Energy consumption at each target point in Scenario 2 (A): without energy constraint, (B): with energy constraint