Path Planning and Energy Management of Hybrid Air Vehicles for Urban Air Mobility

Satyanarayana G. Manyam, David W. Casbeer, Swaroop Darbha, Isaac E. Weintraub, and Krishna Kalyanam

***Abstract*—A novel coupled path planning and energy management problem for a hybrid unmanned air vehicle is considered, where the hybrid vehicle is powered by a dual gas/electric system. Such an aerial robot is envisioned for use in an urban setting where noise restrictions are in place in certain zones necessitating battery only operation. We consider the discrete version of this problem, where a graph is constructed by sampling the boundaries of the restricted zones, and develop a path planning algorithm. The planner simultaneously solves the path planing along with the energy mode switching control, under battery constraints and noise restrictions. This is a coupled problem involving discrete decision making to find the path to travel, and determining the state of charge of the battery along the path, which is a continuous variable. A sampling based algorithm to find near optimal solution to this problem is presented. To quantify the efficacy of the solution, an algorithm that computes tight lower bounds is also presented. The algorithms presented are verified using numerical simulations, and the average gap between the feasible solutions (upper bounds) and the lower bounds are, empirically, shown to be within 15% of each other.**

***Index Terms*—Aerial systems: applications, energy and environment-aware automation, motion and path planning.**

I. Introduction

In THE area of urban air mobility (UAM) and drone delivery, many commercial ventures are considering electric propulsion aircraft [1], [2]. Given the deficiencies in state-of-the-art lithium-ion battery energy density and fuel cell technology [3], [4], it is prudent to consider alternative technologies that can help reduce our carbon footprint in the near term. Furthermore, as the world looks for faster modes of transportation and quicker delivery of goods, our skies will become saturated with the noise from these drones [5]. In the most prevalent use cases, these vehicles will operate in locations where such droning background noise is unacceptable. A gasoline-electric hybrid aerial robotic vehicle is well suited for UAM or drone delivery applications, where the gasoline engine provides long endurance and electric motor facilitates the low noise mode [6]. The perceived noise level when the aerial vehicle is powered by electric motor is considerably less compared to a gasoline engine [7], [8].

Manuscript received 24 February 2022; accepted 22 June 2022. Date of publication 18 July 2022; date of current version 2 August 2022. This letter was recommended for publication by Associate Editor J. Capitan and Editor H. Kurniawati upon evaluation of the reviewers' comments. *(Corresponding author: Satyanarayana G. Manyam.)*

Satyanarayana G. Manyam is with the Infoscitex corporation, A DCS Company, Dayton, OH 45431 USA (e-mail: msngupta@gmail.com).

David W. Casbeer and Isaac E. Weintraub are with the Controls Science Center of Excellence, Air Force Research Laboratory, WPAFB, OH 945433 USA (e-mail: david.casbeer@afrl.af.mil; isaac.weintraub.1@us.af.mil).

Swaroop Darbha is with the Texas A&M University, College Station, TX 77843 USA (e-mail: dswaroop@tamu.edu).

Krishna Kalyanam is with the Aviation Systems Division, NASA Ames Research Center, Moffett Field, CA 94043 USA (e-mail: krishnak@ucla.edu).

Digital Object Identifier 10.1109/LRA.2022.3191810

In this letter, we consider a noise constrained path planning problem for a hybrid gasoline-electric unmanned air vehicle. We assume the robotic vehicle is equipped with a series hybrid architecture [9], where the propellers are powered by an electric motor that draws power from either the gasoline engine-generator or from the battery. The gasoline engine, when run at full capacity (maximum flow rate of fuel), can run the motor and also charge the battery; or the engine can be run at a limited capacity to produce only sufficient power to run the motor. However, due to frictional losses, it is more efficient to run at full capacity and charge the battery to full whenever possible, and then power the motor with the battery. Therefore, we assume that the fuel rate is at maximum whenever the gasoline mode is chosen. We further assume that the architecture facilitates *instant* effortless switching between these two modes. The robotic vehicles considered can make sharp turns similar to quad-rotors, and therefore, we do not consider the kinematic constraints.

The path planning problem involves finding a path between a pre-specified start and goal locations in the presence of quiet zones. The aerial robot is allowed to pass through quiet zones, however it must be powered by the electric mode (gasoline-engine turned off) while flying above such zones. A candidate path can be divided into several segments, where each segment could be either gasoline or electric mode. The cost we have considered in this letter is the fuel cost, and therefore the objective is to minimize the length of the segments that are traveled in gasoline mode. This cost is appropriate for commercial applications where the objective is to minimize the fuel consumption. However, the framework presented here can be modified easily for any other application that aims to minimize a different objective, for example, travel time. The decision making involves finding the path, while simultaneously determining the switching points from gas to electric and vice-versa. To do so, the planner must determine the segments along the path where the power source is the gasoline engine or the electric motor, such that state of charge remains within the capacity limits.

To find the optimal paths, one needs to model the battery characteristics, the rate of charge of the battery while the robot travels in gasoline mode and the rate of discharge in electric

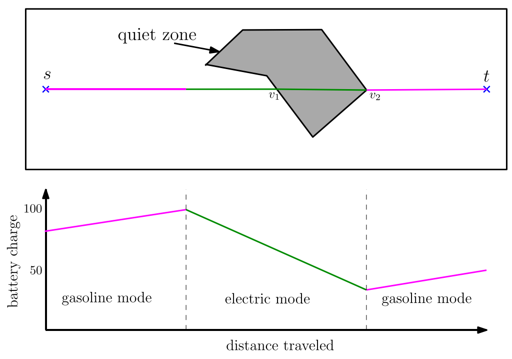


Fig. 1. An example of a feasible path for hybrid navigation.

mode. For simplicity, in the operational limits of the battery charge level, we assume that the rates of discharge and recharge with respect to distance traveled are constants. Let $q(s)$ be the variable representing battery charge along a path, where the parameter s represents the length of the path traveled, then

$$q'(s) = \begin{cases} -\alpha & \text{if electric mode,} \\ \beta & \text{if gasoline mode,} \end{cases} \tag{1}$$

$$q_{\min} \leq q(s) \leq q_{\max}, \tag{2}$$

where α and β are the rates of discharge and recharge per unit distance traveled. This model allows for quick evaluation of feasibility for a given state of charge at initial and final position of a segment of the path. The path planning algorithm we present in this letter can accommodate higher fidelity battery models.

Toy Example: Suppose the initial charge is 80%, and final charge level is required to be at least 50%. A feasible path and the corresponding charge profile along the path are shown in Fig. 1, where there exists one quiet zone, shown in grey. The path is demarcated and shown in magenta and green for the gasoline and electric modes, respectively. Note that the charge profiles are linear due to the model assumed in (1). In this example, the robot travels using the gasoline mode and switches to electric mode before it enters the quiet zone and continues in the electric mode throughout the quiet zone, and switches to the gasoline mode after exiting from the quiet zone. It is clear that the mode of the segment in the quiet zone is electric, and the other segments could be either gasoline or electric. *The planning algorithm must choose the modes and switching points along a path that minimizes the total fuel cost.*

Let us represent a path as a series of segments $\{(v_s, v_1), (v_1, v_2), \ldots (v_n, v_t)\}$. For a given path, one solution approach would be to determine the charge levels $\{q_1, \ldots q_n\}$ at the vertices $\{v_1, \ldots v_n\}$ that are feasible with respect to the battery model, and the state of charge satisfies (2) everywhere along the path. However, for the problem considered in this letter, the path itself is a decision variable; this coupling between the path planning and charge profile planning poses a challenge.

On its own, path planning in the presence of quiet zones resides in a continuous space resulting in an infinite dimensional problem. Sampling the continuous space is a popular technique used to translate the problem to a discrete planning problem. We discretize the problem by sampling the boundaries of the quiet

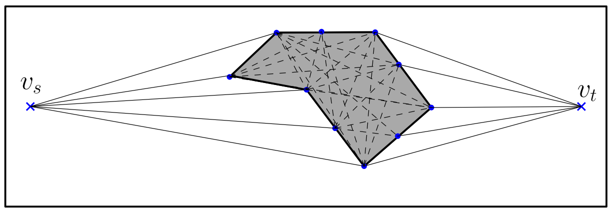


Fig. 2. A graph generated by sampling the boundaries of the quiet zones.

zones, and generate a graph similar to a visibility graph [10], shown in Fig. 2. We address the discrete version of the problem by formulating the path planning and the energy management problem on the discrete graph. It allows us to reduce the infinite dimensional path planning problem into a finite dimensional discrete problem. There is a loss of optimality using this approach, however, it yields a simplified and potentially tractable approach to solve the coupled infinite dimensional problem. Note that, the loss of optimality depends on the sampling intervals along the boundaries, and therefore the loss could be made sufficiently small with sufficiently large sampling rate. We also develop an algorithm that produces tight lower bounds to this problem, and thereby corroborates the quality of the feasible solutions. *The key idea to obtain this lower bound is to partition the domain of a set of continuous variables, that determine the state of charge at each vertex, into a set of sub-intervals. The state of charge at these vertices is allowed to be discontinuous,* i.e.*, the vehicle can arrive at and exit a vertex with different charge levels; but the two charge levels are forced to lie in one of the sub-intervals. This problem is relaxed compared to the original because we are allowing it to violate the continuity of the state of charge at the vertices.*

Related literature: The path planning problems are solved using sampling based algorithms such as RRT [11], RRT* [12], BIT* [13], road-map based search techniques such as PRM methods [14], incremental graph search methods such as D* Lite [15]. Other path planning techniques include visibility graph, Voronoi diagram based or potential field methods [16]. There exists few results in the area of energy aware path planning. In [17], a planning problem is considered where a robot needs to accomplish a set of goals while maintaining a minimum energy threshold. In [18], an energy aware coverage planning problem is addressed, where a robot needs to cover an area that minimizes the total energy consumption. A planning problem to minimize energy consumption in the presence of disturbances is addressed in [19]. The paper uses a model predictive approach to estimate safety critical states, and presents a self triggering schedule to re-plan, and is compared to periodic re-planning. Zhang *et al.* addressed a route planning for a plug-in hybrid vehicle is addresses in [20] that minimizes energy consumption. The authors addressed the coupled routing problem that aims to simultaneously optimize the decision making for path and the power management.

The problem considered in this letter addresses the path planning problem for a hybrid aerial robot, in the presence of (noise-) restricted zones. Here, we aim to optimize the travel cost for a robotic vehicle that can switch between two different modes. To the best of our knowledge, a path planning that allows a robot to switch between travel modes, with constraints on the modes in certain regions, has not addressed before in the literature. The application of this novel problem to urban air mobility and drone delivery is of particular relevance and deserves attention. In sampling based and incremental search techniques, a tree is iteratively constructed by following a sampling procedure. To determine the switch points along the path while simultaneously growing the tree is not possible without decoupling the path planning and energy management problems. The novelty of the proposed algorithm lies in addressing the coupled problem that involves path planning and energy management. Moreover, the presented technique facilitates computation of the lower bound to the optimal solution, which ratifies the quality of the feasible solutions produced.

The main contributions of this work are: i) we present a novel path planning and energy management problem for a hybrid robot suited for an urban air mobility application, ii) we formulate the coupled problem on the discrete graph that involves discrete decision making for the path, and continuous variables for the state of charge along the path, iii) we present a sampling based approach to find near optimal solutions, iv) we develop a partitioning algorithm to find tight lower bounds to the optimal solution, v) we validate the presented algorithms using numerical experiments on some benchmark areas of operation.

The rest of the paper is organized as follows. In Section II, a graph is constructed by sampling the boundaries of the quiet zones, and the hybrid path planning is defined on this graph. The algorithms to compute near optimal solutions and tight lower bounds using sampling and partitioning approach, respectively, are presented in Section III. The algorithms presented were tested using computational experiments, the results are presented in Section IV, and some concluding remarks are provided in Section V.

II. Problem Formulation

Before formalizing the problem, we define the graph, G_s using Algorithm 1, which takes as inputs the start and goal locations, v_s and v_t, a set of quiet zones, $\mathcal{O}$, and a sampling interval, δL. The boundaries of each quiet-zone are sampled uniformly by choosing a point at every δL units of distance. The vertex set, V_s, is built by creating a vertex corresponding to these samples, the start and goal positions (steps 4–6 of Algorithm 1). An edge is added if the two vertices are visible;[1] note that, the edges between the vertices that belong to the same quiet zone are added too (step 8). The edge set consists of edges both inside and outside the quiet zones. The sampled graph of the example problem is shown in Fig. 2. The dotted lines inside the quiet zone represent the feasible segments of the path, where the mode of travel is restricted to electric. Though we consider only the polygonal quiet zones in the examples, this method directly applies to non-polygonal quiet zones too. For non-convex shapes, one may consider convex-hull as done

[1]Two vertices are considered visible if the straight line connecting them does not intersect with any other quiet zones.

Algorithm 1: Construction of the Sampling Graph.

```
1:  Function SamplingGraph$v_s, v_t, \mathcal{O}, \delta L$
2:    $V_s \leftarrow$ INITIATENODESET$()$
3:    $E_s \leftarrow$ INITIATEEDGESET$()$
4:    $V_s \leftarrow V_s \cup \{v_s, v_t\}$
5:    for $O_i \in \mathcal{O}$ do
6:      $V_s \leftarrow V_s \cup$ DISCRETESAMPLING$(O_i, \delta L)$
7:    for $v_i, v_j \in V_s$ do
8:      if CHECKEDGEFEASIBILITY$(v_i, v_j)$ then
9:        $E_s \leftarrow E_s \cup (v_i, v_j)$
10:   $G_s \leftarrow$ CREATEGRAPH$(V_s, E_s)$
11:   return $G_s$
```

in [21], and generate the graph. If one needs to consider obstacles along with quiet zones, it is sufficient to add the vertices of the obstacles to V_s, and corresponding edges that does not intersect with the obstacles.

We formulate and solve the path planning and energy management problem using the sampled graph, $G_s(V_s, E_s)$. With this approximation, the path planning reduces to finding an ordered sequence of nodes $S_p := \{v_s, v_{s_1}, \dots v_{s_k}, v_t\}$ on graph G_s along with the state of charge of the battery at each of these nodes, $\{q_s, q_{s_1}, \dots, q_{s_k}, q_t\}$. Let v_l indicate the l^{th} node in S_p. The following must hold for S_p and the corresponding battery charge at each node $v_l \in S_p$: i) the state of charges, $q_{s_l}, q_{s_{l+1}}$ on every edge $(v_l, v_{l+1}) \in S_p$ satisfies the battery dynamics (1) and (2), ii) the segments of the path in the quiet zones are in electric mode, iii) the state of charge of the battery at the terminal node is greater than a specified value, q_{goal}, and iv) the total cost of travel is minimized. The cost we aim to minimize is the total fuel consumed while traveling in gasoline mode. Therefore, each edge has a zero cost if it is traveled completely in electric mode.

Let $I := \{1, \dots, |V_s|\}$ be the set of indices of the vertices in V_s. For any $i, j \in I$, let x_{ij} denote the binary variable such that $x_{ij} = 1$ if an edge (v_i, v_j) is chosen to be on the path, and $x_{ij} = 0$, otherwise. Given the states of charge of the battery q_i and q_j at the vertices v_i and v_j, respectively, let $c_{ij}(q_i, q_j)$ be the cost of travel from v_i to v_j. Let $\mathbf{x}$ represent a matrix of all binary variables, and $\mathcal{X}$ be the set of all feasible paths from v_s to v_t in G_s. The optimization problem, $\mathcal{P}_1$, is stated as the following.

$$\mathcal{P}_1 : \min_{\mathbf{x} \in \mathcal{X}, q_k \in [q_{\min}, q_{\max}], \forall k \in I} \sum_{i,j \in I} x_{ij} c_{ij}(q_i, q_j) \tag{3}$$

In the following sections, we present the algorithms that simultaneously optimizes the two sets of variables, $\mathbf{x}$ and $\{q_i, i \in V_s\}$. The approach involves sampling of the state of charge at every vertex in V_s, constructing a graph $G_u(V_u, E_u)$, and solving a shortest path problem on this graph; the solution to the shortest path problem on G_u produces a feasible solution (upper bound to optimal solution) to $\mathcal{P}_1$. For a given map with quiet zones, the construction of the base graph without the start and goal can be done offline. We only need to add the start and goal vertices and corresponding edges to compute the shortest path, which significantly reduces the online computation time. We also present a partitioning approach, similar to the upper bounding algorithm, that produces tight lower bounds. In this way, we provide both upper and lower bounds to the optimal solution and hence the gap between the two is the maximum gap between the optimal and feasible (upper bound) solutions.

III. Technical Approach

The algorithms to compute the upper bounds and lower bounds follow a method similar to Algorithm 1 of creating nodes and edges. These algorithms *sample* and *partition* the state of charge at each vertex in V_s. In the upper bounding algorithm, the state of charge at each vertex is uniformly sampled, creating a new node for each sample of state of charge at each vertex. In contrast, the lower bounding algorithm partitions the feasible interval of state of charge into small sub-intervals and builds a graph where each node represents a sub-interval of the state-of-charge at each vertex in V_s. This partitioning approach is similar to that found in [22]–[25]. For coupled optimization problems involving both discrete and continuous variables, this approach is found to produce tight lower and upper bounds, and therefore guarantees the quality of the upper bounds with respect to the optimal solution.

A. Feasible Solution (Upper Bounds)

In problem $\mathcal{P}_1$, at any vertex $v_k \in V_s \backslash \{v_s, v_t\}$, the charge q_k is a continuous variable and $q_k \in [q_{\min}, q_{\max}]$. To compute near optimal feasible solutions, for every $v_k \in V_s \backslash \{v_s, v_t\}$, we chose a discrete set of values, Q_u^k, sampled uniformly in the interval $[q_{\min}, q_{\max}]$. Such a sampling procedure transforms $\mathcal{P}_1$ into the discretized problem $\mathcal{P}_2$ below:

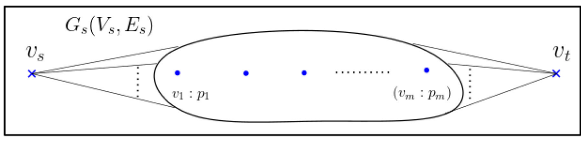
$$\mathcal{P}_2 : \min_{\mathbf{x} \in \mathcal{X}, q_k \in Q_u^k \forall k \in I} \sum_{i,j \in I} x_{ij} c_{ij}(q_i, q_j). \tag{4}$$

By construction, any solution to $\mathcal{P}_2$ is a feasible solution to $\mathcal{P}_1$. To solve $\mathcal{P}_2$, we construct a graph $G_u(V_u, E_u)$, where the set of nodes, V_u, consists of all nodes corresponding to every combination of $v_k \in V_s \backslash \{v_s, v_t\}$ and $q_k \in Q_u^k$. One can choose the discrete set Q_u^k to be the same for every node v_k, and let it be Q_u. For the start node, there is only one value (v_s, q_{init}), and the feasible charges for the goal node is sampled from the set $[q_{goal}, q_{\max}]$; denote this set of charges as Q_{goal}. Then V_u contains the Cartesian product of the sets $V_s \backslash \{v_s, v_t\}$ and Q_u, and the nodes corresponding to the start and goal nodes, that is $V_u := \{V_s \backslash \{v_s, v_t\} \times Q_u\} \cup \{(v_s, q_{init})\} \cup \{\{v_t\} \times Q_{goal}\}$. An illustration of the graphs G_s and G_u is shown in Fig. 3(a) and (b).

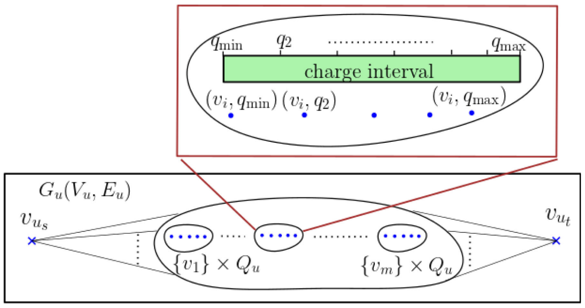
For any nodes, $v_l \in V_u$, let the position corresponding to the node v_l be p_l and state of charge of the battery, q_l. We check if a feasible edge exists between a pair of nodes that complies with the battery dynamics (1), and compute the cost of such edge if it exists. For a given pair of nodes $v_{u_i}, v_{u_j} \in V_u$, where $v_{u_i} = (v_i, q_i), v_{u_j} = (v_j, q_j)$, Algorithm 2 checks if edge (v_{u_i}, v_{u_j}) is feasible with respect to the battery dynamics. The algorithm returns the cost of the corresponding edge as $zero$ if the travel mode is completely electric. Notice that the gasoline engine runs at full capacity whenever it is turned on, and therefore it is sufficient to find the length of the segments that are traveled in gasoline mode to evaluate the cost. If an edge is traveled in both gasoline and electric modes, the algorithm computes the distance traveled in gasoline mode, λd_{ij}, and returns the fuel cost of travel, $c_f \lambda d_{ij}$, where c_f is cost of fuel per unit distance traveled.

Algorithm 2 checks if "all electric" mode is feasible in step 5. The variable λ indicates how much of the edge is traveled in gasoline mode. When an edge is traveled in partly gasoline and partly electric mode, without loss of generality, we assume that the robot travels in gasoline mode first and then switches to electric. Further, we allow a maximum of three switch points on an edge. When an edge is sufficiently long, more than three switch points might be necessary; however, such cases can be accommodated by breaking the long edge into smaller edges by introducing artificial vertices. If an edge is traveled using both modes, the value of λ is computed in steps 9–14, and the corresponding cost is computed in step 18.

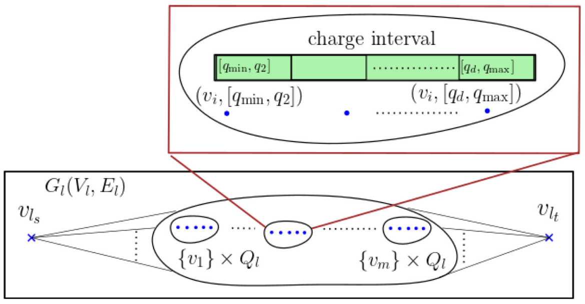
To solve the problem $\mathcal{P}_2$, we construct graph $G_u(V_u, E_u)$, such that a shortest path on this graph produces a solution to the problem $\mathcal{P}_1$. The pseudocode of the algorithm that solves $\mathcal{P}_2$ is presented in Algorithm 3. The problem prescribes a state of charge of the robot at the start, and therefore we can create a node, v_{u_s}, correspondingly, and add to V_u, shown in step 6. In step 7, the discrete set, Q_u, is obtained by sampling



(a) Graph constructed by sampling the boundaries of the quiet zones



(b) $G_u(V_u, E_u)$ obtained by sampling charge at every $v_i \in V_s$



(c) Graph $G_l(V_l, E_l)$ obtained by partitioning charge into sub-intervals

Fig. 3. Graph construction to compute the upper bounds and lower bounds to $\mathcal{P}_1$.

Algorithm 2: Evaluation of an Edge.

```
1:  Function SOCFeasibility$v_{u_i}, v_{u_j}$
2:    $d_{ij} = |position(v_{u_i}) - position(v_{u_j})|$
3:    if $q_j > \min(q_{\max}, q_i + \beta d_{ij})$ then
4:      FeasCheck ← false
5:    else if $q_i - \alpha d_{ij} > q_{\min}$ & $q_j \leq q_i - \alpha d_{ij}$ then
6:      FeasCheck ← true                        ▷ electric mode
7:      $\lambda \leftarrow 0$
8:    else                                      ▷ partly gasoline/electric
9:      $\lambda_1 \leftarrow (q_{\max} - q_i)/(\beta * d_{ij})$
10:     $\lambda_2 \leftarrow (q_{\max} - q_j)/(\alpha * d_{ij})$
11:     if $\lambda_1, \lambda_2 \geq 0\&amp;\lambda_1 + \lambda_2 \leq 1$ then
12:       $\lambda \leftarrow \lambda_1 + (\frac{\alpha}{\alpha+\beta}) * (1 - \lambda_1 - \lambda_2)$
13:     else
14:       $\lambda \leftarrow \frac{q_j - q_i + \alpha d_{ij}}{(\alpha+\beta) d_{ij}}$
15:     if $0 \leq \lambda \leq 1$ then
16:       FeasCheck ← true
17:   if FeasCheck
18:     $cost \leftarrow c_f \lambda d_{ij}$ ▷ fuel cost
19:   return FeasCheck, $cost$
```

Algorithm 3: Construction of the Graph, G_u.

```
1:  Function SamplingGraphG$G_s, \delta q$
2:    $V_u \leftarrow$ INITIATENODESET()
3:    $E_u \leftarrow$ INITIATEEDGESET()
4:    $v_{u_s} \leftarrow$ CREATENODE$(v_s, q_{init})$
5:    $v_{u_t} \leftarrow$ CREATENODE$(v_t)$
6:    $V_u \leftarrow V_u \cup \{v_{u_s}\}$
7:    $Q_u \leftarrow$ DISCRETESAMPLING$([q_{min}, q_{max}], \delta q)$
8:    $Q_{goal} \leftarrow$ DISCRETESAMPLING$([q_{goal}, q_{max}], \delta q)$
9:    for $v_i \in V_s, q_j \in Q_u$ do
10:       $V_u \leftarrow V_u \cup$ CREATENODE$(v_i, q_j)$
11:   for $q_j \in Q_{goal}$ do
12:       $V_{goal} \leftarrow V_{goal} \cup$ CREATENODE$(v_t, q_j)$
13:   $V_u \leftarrow V_u \cup V_{goal}$
14:   for $v_{u_i}, v_{u_j} \in V_u$ do
15:     if CHECKEDGEFEASIBILITY$(v_{u_i}, v_{u_j})$ then
16:         $feas, cost \leftarrow$ SOCFEASIBILITY$(v_{u_i}, v_{u_j})$
17:         if feas then
18:             $E_u \leftarrow E_u \cup (v_{u_i}, v_{u_j})$
19:             $c_{ij} \leftarrow$ cost
20:   $V_u \leftarrow V_u \cup \{v_{u_t}\}$
21:   for $v_{u_k} \in V_{goal}$ do
22:       $E_u \leftarrow E_u \cup (v_{u_k}, v_{u_t})$
23:   $G_u \leftarrow$ CREATEGRAPH$(V_u, E_u)$
24:   $path_u \leftarrow$ SHORTESTPATH$(G_u, v_{u_s}, v_{u_t})$
25:   return $path_u$
```

the interval $[q_{\min}, q_{max}]$. In steps 9–10, for every combination of $(v_i, q_j), v_i \in V_s \backslash \{v_s, v_t\}, q_j \in Q_u$, we create a node and add to V_u. A minimum state of charge, q_{goal}, is required at the goal position, and to satisfy that, we sample uniformly in the interval $[q_{goal}, q_{max}]$, and create the set of nodes, V_{goal} (shown in step 12), that correspond to the goal position and a state of charge $q_j \in Q_{goal}$. In steps 14–19, we construct the set of edges, E_u, by adding an edge for every pair of nodes in V_u, if Algorithm 2 returns a feasible solution; the costs returned by the algorithm are set as the weights of those edges. Note that there could be multiple nodes in V_{goal}, that correspond to the goal position, and satisfy the minimum charge required at goal. Therefore, any path from v_{u_s} to a node in V_{goal} is a feasible path. To find the minimum cost path, we add an another node v_{u_t}, that corresponds to the goal, and add zero cost edges between all nodes in V_{goal} and v_{u_t}, shown in steps 20–22. Finally, we use Dijkstra's algorithm to find the optimal shortest path from v_{u_s} to v_{u_t} in step 24. The time complexity of Dijkstra's algorithm is $\mathcal{O}(|V_u|^2)$, and it returns the minimum cost path on the graph G_u. The trajectory for the robot is constructed using the position of the vertices in the shortest path.

The algorithm returns the shortest path, $path_u$, which is comprised of a series of nodes $\{v_{s_1}, \dots v_{s_p}\}$, and each of these nodes correspond to the vertices in V_u. This series of vertices in V_u is the path taken by the robot. The states of charge corresponding to each of the nodes in $path_u$ are the charges at corresponding vertices in V_u. The feasibility check in step 16 of Algorithm 3 ensures the feasibility of the whole path. Since, the weight of each edge is the cost of travel in gasoline mode, the shortest path is the path of minimum fuel cost. The optimal solution of $\mathcal{P}_2$ is a feasible solution to $\mathcal{P}_1$, but may not be optimal due to the discrete sampling. A tight lower bound could corroborate the quality of a feasible solution, and an algorithm to compute tight lower bounds is presented in the next section.

B. Lower Bounds

To compute lower bounds, it is a common practice to relax a set of constraints and the optimal solution of the resulting relaxed problem gives a lower bound to the original problem. The Held-Karp bounds for traveling salesman problem [26], is a good example of this technique. In recent work, for coupled problems involving discrete and continuous decision variables, a technique was developed where a set of constraints are relaxed, and another set of 'loose' constraints are added, such that it produces tight lower bounds. This was shown to produce very tight lower bounds for routing problem with turn radius constraints [23], [24] and for neighborhood traveling salesman problem [25]. The problem in this letter is also a coupled problem; and we develop an algorithm using a similar idea of partitioning the continuous decision variables, that produces tight lower bounds.

To compute the lower bound, we pose a relaxation of the problem $\mathcal{P}_1$. Any feasible solution to $\mathcal{P}_1$ would be given as a sequence of nodes $\{v_1, \dots v_p\}$, and a state of charge at each of these nodes, $\{q_1, \dots q_p\}$. The continuity of the state of charge dictates that at any intermediate node v_j along the path, the state of charge at the end of prior edge is same as the state of charge at the beginning of the following edge. For example, if a feasible solution contains two successive edges, (v_i, v_j) and (v_j, v_k), let q^e_{ij} be the charge at the end of edge (v_i, v_j), and q^s_{jk} be the charge at the start of edge $v_j v_k$. The position of the end of the edge (v_i, v_j) is same as the start of the edge (v_j, v_k); therefore, the continuity of the charge profile dictates that $q^e_{ij} = q^s_{jk}$. We relax this continuity constraint and allow q^e_{ij} and q^s_{jk} to be different but we restrict them to lie in an interval, i.e., $q^e_{ij}, q^s_{jk} \in (q_p, q_{p+1})$. We refer to this relaxed problem as $\mathcal{P}_3$. Since, this is a relaxation to $\mathcal{P}_1$, every feasible solution to $\mathcal{P}_1$ is also feasible to the relaxed problem $\mathcal{P}_3$. Therefore, the optimal solution of $\mathcal{P}_3$ is a lower bound to the optimal solution of $\mathcal{P}_1$.

To this end, at every node, we partition the feasible *interval* of state of charge into n_l *sub-intervals*. At the vertices $v_i \in V_s \backslash \{v_s, v_t\}$, the set of intervals would be $Q^i_l = \{[q_{\min}, q_1], [q_1, q_2], \dots [q_{n_l-1}, q_{\max}]\}$. At the goal node, the minimum charge required is q_{goal}, and therefore the intervals would be $\{[q_{goal}, q_1], \dots [q_{n_g-1}, q_{\max}]\}$, for some n_g. Let $\bar{q}_i$ represent an interval of states of charge at a node v_i. Now the relaxed problem $\mathcal{P}_3$ is stated as follows:

$$\mathcal{P}_3 : \min_{\mathbf{x} \in \mathcal{X}, \bar{q}_k \in Q^k_l, \forall k \in I} \sum_{i,j \in I} x_{ij} c_{ij}(\bar{q}_i, \bar{q}_j). \tag{5}$$

We solve $\mathcal{P}_3$ by constructing a graph, $G_l(V_l, E_l)$, similar to the $G_u(V_u, E_u)$. The nodes in V_l are the combination of the vertices $v_i \in V_s$ and the intervals $\bar{q}_j \in Q^j_l$. For a pair of nodes $v_{l_i}, v_{l_j} \in V_l$, let (q^i_k, q^i_{k+1}) and (q^j_m, q^j_{m+1}) be the corresponding charge intervals. The cost of the corresponding edge in E_l is the minimum cost of the edge,

$$\min_{q_i \in (q^i_k, q^i_{k+1}), q_j \in (q^j_m, q^j_{m+1})} c_{ij}(q_i, q_j). \tag{6}$$

Due to the linear battery dynamics, this cost could be found by using Algorithm 2 with the upper limit of the interval at the first node, and lower limit of the interval at the second node, q^i_{k+1}, q^j_m. The graph construction of $G_l(V_l, E_l)$ is similar to the one presented in Algorithm 3, however it differs only in two aspects: (i) in steps 7–8, the discrete sampling is replaced with continuous partition of the interval $[q_{min}, q_{max}]$ and $[q_{goal}, q_{max}]$, respectively, and (ii) the edge cost, in step 16, are assigned using the solution of (6). Similar to the steps 10, 12 of Algorithm 3, nodes are created corresponding to a combination of vertices $v_i \in V_s$ and intervals $\bar{q}_j \in Q^i_l$. The rest of the graph construction goes similar, and therefore, to avoid the repetition, we do not present the pseudocode for the lower bounding algorithm. An illustration of the construction of the graphs G_u and G_l is shown in Figs. 3(b)–3(c). In the following theorem, we formally prove that the optimal solution to $\mathcal{P}_3$ is a lower bound to the optimal solution of $\mathcal{P}_1$.

Theorem 1: The optimal solution of $\mathcal{P}_3$ is a lower bound to the optimal solution of $\mathcal{P}_1$.

Proof: It is sufficient to show that every feasible solution to $\mathcal{P}_1$ is also a feasible solution to $\mathcal{P}_3$, and the cost of the solution to $\mathcal{P}_3$ is less than or equal to the cost of the solution to $\mathcal{P}_1$. Let $V_{feas} := \{v_s, v_{s_1}, \dots v_{s_p}, v_t\}$ be the sequence of vertices in a feasible solution to $\mathcal{P}_1$, and $Q_{feas} := \{q_s, q_{s_1}, \dots q_{s_p}, q_t\}$ be the corresponding states of charge at those vertices. For each vertex $v_k \in V_{feas}$ and the corresponding state of charge q_k, there exists an interval $\bar{q}_k \in Q^k_l$, such that $q_k \in \bar{q}_k$. Construct a path in G_l by identifying the nodes corresponding to v_k and $\bar{q}_k$. This gives a feasible path from v_{l_s} to v_{l_t} in G_l, and let V^l_{feas} be the sequence of nodes. The cost of each edge between successive nodes in V^l_{feas} is less than or equal to the cost of corresponding edge in V_{feas} due to (6), and thus cost of the path in V^l_{feas} is less than or equal to the one in V_{feas}. □

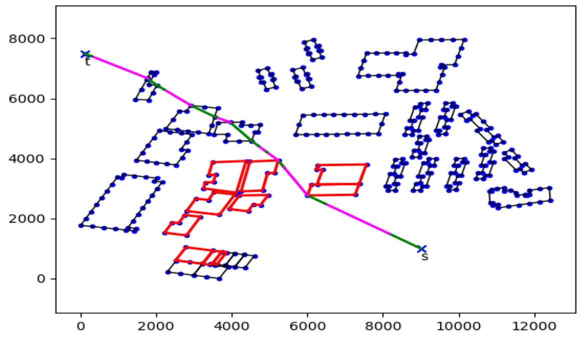
IV. Computational Results

To evaluate their performance, we have tested the algorithms to compute the upper bounds and the lower bounds using several scenarios constructed from randomly generated maps and benchmark maps. Since the problem of *path planning in the presence of obstacles* is closely related to the problem we consider, we use previously established benchmark maps [27] to test our methods. We constructed the maps by randomly generating polygonal restricted zones, where the centers of the polygons are sampled from an uniform distribution. The number of sides are also randomly generated for each restricted zone. There are 10, 15, 20 and 25 restricted zones in $map1$, $map2$, $map3$ and $map4$, respectively. We have constructed two more maps using the benchmark instances 'boston2' and 'newyork0' from [27]. These are based on real world maps of a region in the cities Boston and New York; we identified the regions directly above the buildings as restricted zones, and they are appropriate for drone delivery applications as discussed in Section I. An interested reader can access the Julia code to extract the buildings from the OpenStreetMaps data from https://github.com/manyamgupta/HybridPathPlanning.git.

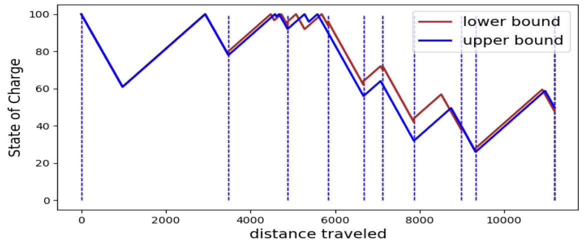
We have generated 50 scenarios with each of the above maps, where the start and goal positions are chosen from a random distribution such that the straight line distance between them is greater than a specified limit. Further, the instances are run with different levels of discretization of the states of charge. The rate of discharge, α, and the rate of recharge, β are chosen such that the ratio $\frac{\alpha}{\beta}$ is equal to two, *i.e.*, the battery discharges twice as fast as the it recharges per unit distance traveled.

In Fig. 4(a), a path of the feasible solution is shown for a scenario generated in the 'newyork0' map, and the charge profiles of the feasible path and the lower bound are shown in Fig. 4(b). We consider no-fly zones in this scenario, shown in red, and were addressed as explained in Section II. The vertical lines are the positions of the vertices in Fig. 4(b), and one may observe the charge profile is not continuous for the lower bound path. This is expected due to the relaxation of the charge continuity, and the resulting solution is a lower bound, rather than a feasible solution. For this scenario, the cost of the feasible path produced by Algorithm 3 is 5318 and the lower bound is given as 5137, therefore, the gap between upper bound and lower bound is around 3.5%. This infers that the feasible solution is within less than 3.5% from the optimal solution.

The percent gap between the lower bounds and upper bounds is a measure of the quality of the feasible solutions. This is the maximum gap between the feasible solution and the optimal solution. For each of the maps, there are 50 scenarios, and each scenario is solved with 20, 30 and 40 discretizations. The box plot of the percentage gap is shown in Fig. 5(a). Clearly, with higher sampling rate, the algorithm produces better solutions as evident from the reducing gap with higher discretizations. The maps *Boston* and *New York* have a few quiet zones with very small edges, and because of these, the lower bound graph

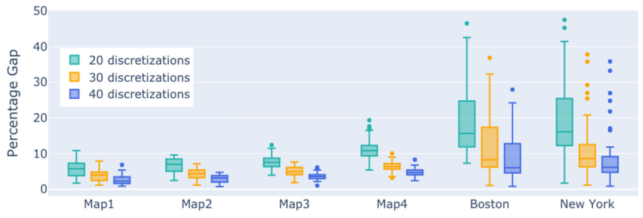


(a) Solution produced by Algorithm 3

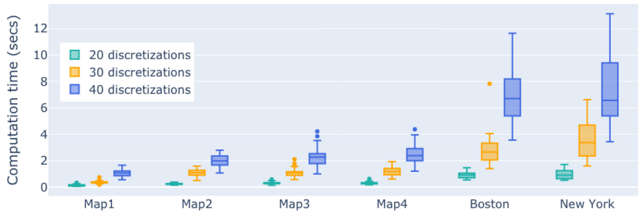


(b) Charge profile of the path

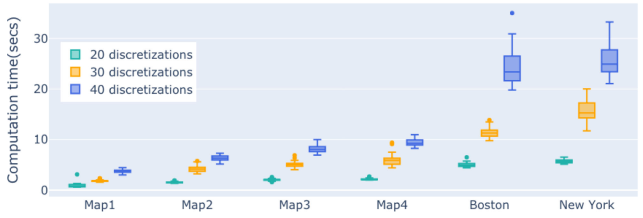
Fig. 4. Results of a scenario generated in the ‘newyork0’ map with 35 quiet zones.



(a) Average of percent gap between upper and lower bounds



(b) Online computation time required for upper bound



(c) Computation time required for lower bound

Fig. 5. Computational results from benchmark and random maps.

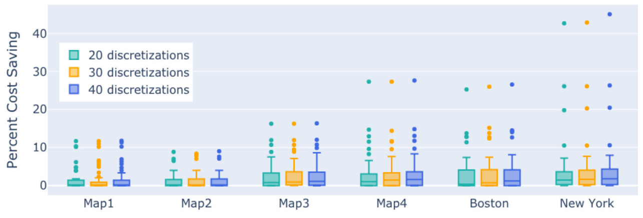


Fig. 6. Percentage cost reduction compared to path planning that avoids quiet zones.

consists of many zero cost edges. For example, let $[q^a, q^b]$ be the charge interval corresponding to the vertices v_k and v_l in G_l. If the charge required to travel on the edge (v_k, v_l) is less then $q^b - q^a$, this edge can be on a path with zero cost, and have same charge at the start and end of the edge. This resulted in loose lower bounds, and hence the higher gap. However, the algorithm produces tighter lower bounds by choosing sufficiently large number of sub-intervals n_l while constructing the graph G_l. But, this comes at a higher cost in computational time required. The computation time required to find the upper bounds and lower bounds are shown as box and whisker plots in Figs. 5(b) and 5(c). The higher computation times for the *Boston* and *New York* maps is due to the higher number of quiet zones.

A significant part of the computational effort in Algorithm 3 is spent constructing the graph $G_u(V_u, E_u)$. However, for UAM applications the restricted zones, start and end locations are know *a priori* and can be computed ahead of time. For other applications, like package delivery, the restricted zones and the problem parameters are known *a priori*, but the positions of the start and goal may not be known. In practice, one may construct the parts of the graph $G_u(V_u, E_u)$ offline without the nodes corresponding to the start and goal position, and edges incident on them. And when the robot’s start and goal positions are specified, the corresponding edges of the graph could be constructed and added to the graph . Therefore, to validate the feasibility of online implementation of this algorithm, it is sufficient to analyse the computational effort of the online part. The online computation time required by Algorithm 3 is shown in Fig 5(b). Though the effort required increases with higher number of restricted zones and higher sampling rate, it is still in the order of seconds, and therefore, is viable for on-board implementation.

To evaluate the cost savings from the proposed framework, we solved the related but different path planning problem where the quiet zones are considered to be no-fly zones (i.e., feasible paths must completely avoid the quiet zones). This is done by removing the quiet zone edges from the graph G_u, and solving for the shortest path thereafter. Note that, we still consider the hybrid mode of the robotic vehicle, and the switching between the gasoline and electric mode still exists. If we were to restrict this to gasoline mode only, the savings would be much larger. The percentage reduction in cost using Algorithm 3 is presented in Fig. 6. The cost savings are around 5 to 10% for most cases, with some cases having much higher savings. This large variance is due to the randomly generated start and goal locations; the cost reduction depends on the difference in length between the shortest path that avoids the "no-fly zones" and path generated from our work that passes through the restricted zones.

V. Conclusion

A novel hybrid path planning problem that arises from urban air mobility is presented. In this path planning problem, the hybrid vehicle is required to run in electric mode in certain regions to comply with noise restrictions. The path planner needs to generate a path and schedule for switching between gasoline and electric modes. Algorithms based on sampling and partitioning are presented yielding upper and lower bounds to the coupled path planning and energy management problem. The paper assumes a linear battery model, but a higher fidelity model could be easily integrated with the algorithms presented here. The solutions produced by the presented algorithms are empirically shown to yield upper and lower bounds that are within 15% of one another, indicating that the feasible solutions are of high quality.

As a future research direction, one may develop an iterative scheme to compute lower bounds that refines the partitioning in each iteration only where it is necessary, and thus overcome the cost of higher sampling. Another direction of future research includes the adaptive sampling of the boundaries of the quiet zones that chooses higher number of samples on the boundaries that are more likely to be on the path.

References

[1] S. Hasan, "Urban air mobility (UAM) market study," NASA, Washington, D.C, USA, Tech. Rep. HQ-E-DAA-TN65181, 2019.

[2] M. D. Patterson et al., "An initial concept for intermediate-state, passenger-carrying urban air mobility operations," in *Proc. AIAA Scitech Forum*, no. 1626, 2021, Art. no. 1626.

[3] K. Button, "Faith in batteries," *Aerosp. America*, pp. 36–42, Oct. 2021.

[4] V. Smil, "Decarbonization algebra: The COP26 calls for impossibly steep cuts in carbon emissions: Numbers don't lie," *IEEE Spectr.*, vol. 59, no. 2, pp. 20–21, Feb. 2022.

[5] S. A. Rizzi et al., "Urban air mobility noise: Current practice, gaps, and recommendations," NASA, Washington, D.C, USA, Tech. Rep. NASA/TP-20205007433, 2020.

[6] W. J. Fredericks, M. D. Moore, and R. C. Busan, "Benefits of hybrid-electric propulsion to achieve 4x cruise efficiency for a VTOL UAV," in *Proc. Int. Powered Lift Conf.*, 2013, Art. no. 4324, doi: 10.2514/6.2013-4324.

[7] R. Cabell, F. Grosveld, and R. McSwain, "Measured from small unmanned aerial vehicles," in *Proc. Inter-Noise Noise-Con Congr. Conf. Proc.*, 2016, pp. 345–354.

[8] H. D. Kim, A. T. Perry, and P. J. Ansell, "A review of distributed electric propulsion concepts for air vehicle technology," in *Proc. IEEE/AIAA Electric Aircr. Technol. Symp.*, 2018, pp. 1–21.

[9] J. Lieh, E. Spahr, A. Behbahani, and J. Hoying, "Design of hybrid propulsion systems for unmanned aerial vehicles," in *Proc. AIAA/ASME/SAE/ASEE 47th Joint Propulsion Conf. Exhibit*, 2011, Art. no. 6146.

[10] M. de Berg, M. van Kreveld, M. Overmars, and O. C. Schwarzkopf, *Computational Geometry: Algorithms and Applications*. Berlin, Heidelberg: Springer, 2000, pp. 307–317.

[11] J. Kuffner and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE 2000 ICRA. Millennium Conf. Int. Conf. Robot. Automat.. Symposia Proc. (Cat. No.00CH37065)*, 2000, pp. 995–1001.

[12] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[13] J. D. Gammell, T. D. Barfoot, and S.S. Srinivasa, "Batch informed trees (BIT*): Informed asymptotically optimal anytime search," *Int. J. Robot. Res.*, vol. 39, no. 5, pp. 543–567, 2020.

[14] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[15] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 354–363, Jun. 2005.

[16] O. Souissi, R. Benatitallah, D. Duvivier, A. Artiba, N. Belanger, and P. Feyzeau, "Path planning: A 2013 survey," in *Proc. Int. Conf. Ind. Eng. Syst. Manage.*, 2013, pp. 1–8.

[17] T. X. Lin, E. Yel, and N. Bezzo, "Energy-aware persistent control of heterogeneous robotic systems," in *Proc. Amer. Control Conf.*, 2018, pp. 2782–2787.

[18] C, D. Franco, and G. Buttazzo, "Energy-aware coverage path planning of UAVs," in *Proc. IEEE Int. Conf. Auton. Robot Syst. Competitions*, 2015, pp. 111–117.

[19] N. Bezzo, K. Mohta, C. Nowzari, I. Lee, V. Kumar, and G. Pappas, "Online planning for energy-efficient and disturbance-aware UAV operations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 5027–5033.

[20] Q. Zhang, K. Wu, and Y. Shi, "Route planning and power management for PHEVs with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 4751–4762, May 2020.

[21] H. Huang and A. V. Savkin, "Viable path planning for data collection robots in a sensing field with obstacles," *Comput. Commun.*, vol. 111, pp. 84–96, 2017.

[22] S. G. Manyam, S. Rathinam, D. Casbeer, and E. Garcia, "Tightly bounding the shortest dubins paths through a sequence of points," *J. Intell. Robot. Syst.*, vol. 88, no. 2, pp. 495–511, 2017.

[23] S. G. Manyam and S. Rathinam, "On tightly bounding the dubins traveling salesman's optimum," *J. Dyn. Syst., Meas. Control*, vol. 140, no. 7, 2018, Art. no. 0 71013.

[24] S. Rathinam, S. G. Manyam, and Y. Zhang, "Near-optimal path planning for a car-like robot visiting a set of waypoints with field of view constraints," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 391–398, Apr. 2019.

[25] P. Váňa and J. Faigl, "On the dubins traveling salesman problem with neighborhoods," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Hamburg, Germany, 2015, pp. 4029–4034.

[26] M. Held and R. M. Karp, "The traveling-salesman problem and minimum spanning trees," *Oper. Res.*, vol. 18, no. 6, pp. 1138–1162, 1970.

[27] N. Sturtevant, "Benchmarks for grid-based pathfinding," *Trans. Comput. Intell. AI Games*, vol. 4, no. 2, pp. 144–148, Jun. 2012.