2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

# Energy-Aware Aerial Vehicle Deployment via Bipartite Graph Matching

Lantao Liu and Nathan Michael

*Abstract*— **This paper proposes a multi-robot path planning and optimal deployment strategy for a team of micro air vehicles with limited energy reserves and finite recharge times. We focus on deployments which seek to balance individual and cooperative vehicle task requirements with overall travel and energy costs and charging station availability toward enabling extended duration operation. We formulate the deployment approach as a matching problem that builds upon a deterministic navigation graph of both edge and vertex weighted. By relating the charging stations to the weighting policy of graph vertices, a set of navigation paths transiting nearby charging stations can be obtained for those low energy aerial vehicles. Simulation results validate the proposed deployment approach and analyze performance variability due to changes in available energy resources and team size.**

## I. Introduction

We consider the problem of developing cooperative vehicle deployments that enable a team of micro air vehicles (MAVs) to pursue an objective such as surveillance, monitoring, or exploration of an indoor environment with cognizance of available onboard energy resources and predeployed recharging platforms. The efficacy of MAVs in applications requiring operation over long durations is fundamentally limited due to available onboard energy reserves and vertical take-off and landing systems such as quadrotors continuously consume energy resources during operation. Strategies for autonomous recharging of air vehicles include the use of static and mobile docking stations [12] as well as deployment methods that adapt vehicle plans toward anticipating energy depletion via routes that ensure charging opportunities [1, 6, 11]. A challenge in the effective application of such techniques is the online assignment of vehicles to charging stations that balances the evolving requirements of the collective objective and individual energy requirements.

In this work, we propose an online deployment methodology for a team of micro air vehicles to visit goal locations while considering the availability of nearby ground charging stations and the current vehicle energy levels. The proposed strategy seeks to address the requirement that all vehicles maintain a sufficient energy reserve to ensure continued operation. Similar to existing works focusing on task allocation-based multi-vehicle exploration and surveillance [9, 13, 15], we consider a deployment where vehicles are assigned to paths leading to real-time allocated goals toward furthering the collective objective [7]. The primary contribution of this work is the introduction of energy constraints in the task allocation formulation that permits online deployment adaptation to the evolving system state and energy levels.

The authors are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA 15213, USA, {lantao, nmichael}@cmu.edu

To reach the deployed goal locations, the aerial vehicles follow their routing paths planned on a constructed deterministic navigation map (an undirected graph) that describes the environment [2, 14]. We address this energy constrained routing problem through formulating the navigation map with weighted vertices and edges that associate vehicle energy requirements and path cost to charging requirements. By casting the navigation map as a bipartite graph matching problem, the path planning and task allocation are combined together. The resulting vehicle deployment paths conservatively assign vehicles to goals (tasks) or charging stations based on the current system need. The polynomial complexity approach permits online task reassignment based on the evolving energy resources of individual vehicles and charging stations. Tasks originally assigned to low energy vehicles can also be relayed to other aerial vehicles that possess better energy conditions.

Related to this study is the topic of simultaneous multi-vehicle path planning methods on environment graph structures [8]. Our prior work [5] addressed the problem of multi-robot navigation by adapting the graph matching variant of the Hungarian algorithm [4]—originally designed to solve the optimal assignment problem—to construct routing paths in a spatial topology. The approach has several useful features including being particularly effective at generating multiple non-interfering paths. Although some preliminary path behaviors are demonstrated by uniformly scaling the diagonal of adjacency matrix representing a Euclidean graph, paths constrained by certain properties have not been exploited. This proposed approach seeks to minimize the total travel cost while ensuring continued vehicle energy resources by routing vehicles through charging stations (as in [6, 11]).

## II. Energy Constrained Deployment and Path Planning

An objective of this work is to design an effective planning method to deploy $n$ aerial robots to $m$ goal locations ($n \leq m$), while accounting for nearby energy charging stations and resource availability. Specifically, the planning objectives capture considerations such as limiting travel cost toward reducing energy depletion, maintaining onboard energy levels, and ensuring the availability of replacement vehicles to which a task may be relayed to continue the overall system objective (see Fig. 1).

Additionally, we make the following assumptions the problem formulation: (1) charging stations are stationary (following a deployment phase); (2) the number of charging stations is less than the number of aerial robots and each

978-1-4799-2376-2/14/$31.00 ©2014 IEEE

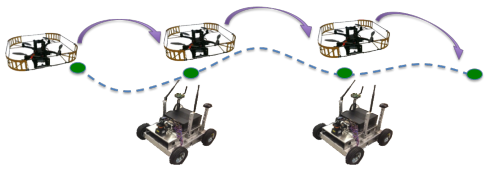

Fig. 1. Tasks are relayed along a path. The rightmost vehicle continues the unfinished task that is planned for the leftmost vehicle. The two vehicles on the left eventually occupy the charging stations (specially equipped mobile ground robots).

charging station can charge one air vehicle at a time; and (3) the aerial robot navigation paths are planned in 2-D. Consistent with assumptions (1) – (2), we envision mobile ground platforms as charging stations that are able to redeploy throughout the evolution of the exploration or surveillance application. The final assumption reflects the so-called re-projective or 2.5-D assumption and treats the structured 3-D world as a 2-D environment [10].

### A. Planning on an Edge and Vertex Weighted Graph

We use an undirected graph $G = (V, E)$ to describe the navigable environment based on topological map. Specifically, each vertex $v \in V$ represents a location in the environment, whereas an edge $e(v, u) \in E$ denotes the traversability between $u, v$ with an edge weight $w_e(u, v) > 0$ representing the travel cost. We call $G$ the *navigation graph*, which will be used for multi-vehicle path planning.

Different from a standard graph, the vertices of $G$ are subject to importance of differing levels. For instance, if a charging station with an available charging outlet is currently staying on a vertex, then this vertex acts as an energy supplier and is more important than other unoccupied vertices. We use $w_v(v) > 0$ to denote the weight/importance of vertex $v \in V$. To fit the *importance maximization* in our *cost minimization* planning objective, the negated value $-w_v(v)$ is used.

The proposed formulation differs from the traditional routing methods for two reasons. First, general routing algorithms only accept graphs with either weighted edges or weighted vertices (not both). In our problem, the graph has both edges and vertices weighted, and each vehicle eventually obtains a unique path $P = (V_P, E_P) \in G$ with the path cost

$$c_P = \sum_{e(v_i,v_j)\in E_P \in P} w_e(v_i, v_j) - \sum_{v_i \in V_P \in P} w_v(v_i). \tag{1}$$

Second, we wish to deploy multiple MAVs simultaneously thus a set of paths $\mathbb{P} = \{P_1, P_2, \cdots, P_n\}$ starting from vertices $V_s = \{v_{s1}, \cdots, v_{sn}\}$ and ending at $V_t = \{v_{t1}, \cdots, v_{tn}\}$ must be generated, with minimized overall cost

$$\begin{aligned} \text{obj:} \quad & \min \sum_{k=1}^{n} c_{P_k} \\ & = \min \sum_{k=1}^{n} \left( \sum_{e(v_i,v_j)\in P_k} w_e(v_i, v_j) - \sum_{v_i \in P_k} w_v(v_i) \right) \\ \text{s.t.} \quad & P_k \cap P_{k'} = \varnothing \text{ if } k \neq k'. \end{aligned} \tag{2}$$

Existing methods are not readily identified that enable computing multiple paths subject to (2).

### B. Vertex Weight Elimination

To simplify the weighting rule, we keep the edge weights but eliminate the vertex weights by transforming $G$ to another graph representation—the bipartite (matching) graph $\tilde{G} = (V, V', \tilde{E})$:

$$f : G = (V, E) \to \tilde{G} = (V, V', \tilde{E}). \tag{3}$$

A bipartite graph $\tilde{G}$ has two sets of nodes $V$ and $V'$, where $V'$ is simply a copy of $V$ such that $|V| = |V'|$, and an edge $\tilde{e}(v, v') \in \tilde{E}$ connects the vertices $v \in V$ and $v' \in V'$. More formally, if there is an edge $e(v_i, v_j) \in E \in G$, we construct a pair of bipartite graph edges,

$$\begin{aligned} \tilde{e}(v_i, v'_j) \in \tilde{E} \in \tilde{G}, \quad v_i \in V, v'_j \in V' \\ \tilde{e}(v'_i, v_j) \in \tilde{E} \in \tilde{G}, \quad v'_i \in V', v_j \in V \end{aligned} \tag{4}$$

both of which are weighted the same as the counterpart edge $e(v_i, v_j)$

$$w_{\tilde{e}}(v_i, v'_j) = w_{\tilde{e}}(v'_i, v_j) = w_e(v_i, v_j) \tag{5}$$

As $V_s$, $V_t$ are the starting and ending vertices and are predetermined before the paths are sought, there is no need to weight them. Thus, $\forall v_i \in V \backslash \{V_s, V_t\}$ a set of edges are created:

$$\tilde{e}(v_i, v'_i) \in \tilde{E} \in \tilde{G}, \tag{6}$$

which are weighted by

$$w_{\tilde{e}}(v_i, v'_i) = w_v(v_i). \tag{7}$$

Although the graph transformation increases the dimensions of both the vertex set and the edge set, the vertex weights $w_v(v_i)$ are eliminated by transferring them onto the newly added bipartite graph edges $\tilde{e}(v_i, v'_i)$.

### C. Computing Navigation Paths

After $G$ is transformed to $\tilde{G}$, each internal vertex $v$ on path $P \in G$ is substituted with a pair of connected vertices $(v, v')$. For example, $P : v_s \to \cdots v_l \cdots \to v_t$ is changed to $P' : v_s \to \cdots \to v_l \to v'_l \cdots \to v_t$. With uniform edge weights, the path cost in (1) is written as

$$c_P = c_{P'} = \sum_{i\neq j, \tilde{e}(v_i,v'_j)\in P'} w_{\tilde{e}}(v_i, v'_j) - \sum_{\tilde{e}(v_i,v'_i)\in P'} w_{\tilde{e}}(v_i, v'_i). \tag{8}$$

Let $\mathcal{S}_v$ be the weight sum of all edges $\tilde{e}(v_i, v'_i)$ constructed in (6),

$$\mathcal{S}_v = \sum_{v_i \in V \backslash V_s} \tilde{e}(v_i, v'_i) \tag{9}$$

By adding (9) to the total cost of all paths $\mathbb{P}' = \{P'_1, \cdots, P'_n\}$, we get

$$\begin{aligned}
&\sum_{k=1}^{n} c_{P'_k} + \mathcal{S}_v \\
&= \sum_{i\neq j, \tilde{e}(v_i, v'_j)\in\mathbb{P}'} w_{\tilde{e}}(v_i, v'_j) + \left(\mathcal{S}_v - \sum_{\tilde{e}(v_i, v'_i)\in\mathbb{P}'} w_{\tilde{e}}(v_i, v'_i)\right) \\
&= \sum_{i\neq j, \tilde{e}(v_i, v'_j)\in\mathbb{P}'} w_{\tilde{e}}(v_i, v'_j) + \sum_{\tilde{e}(v_i, v'_i)\notin\mathbb{P}'} w_{\tilde{e}}(v_i, v'_i)
\end{aligned} \tag{10}$$

A close observation of the above equation reveals that each vertex $v_i \in V$ appears only once in the summation, and so does $v'_i \in V'$. This fact implies that a *matching* between vertices in $V$ and vertices $V'$ is formed, where each vertex $v_i \in V$ is uniquely matched to a vertex $v'_i \in V'$.

Therefore, minimizing (10) produces a *perfect matching*, which corresponds to the optimal assignment solution. Consequently, the multi-robot path planning problem (with minimized weights of both vertices and edges) can be solved by existing optimal assignment algorithms such as the Hungarian method [4] which has a benchmark time complexity of $O(|V|^3)$.

### D. Controlling Path Behaviors

In (10), we transform the path planning problem into the optimal assignment problem through the addition of the term $\mathcal{S}_v$. Hence, $\mathcal{S}_v$ is a parameter that affects the path planning results. As $\mathcal{S}_v$ is the weight sum for all edges constructed in (6), which are exactly the weights of vertices $v_i \in V \setminus \{V_s, V_t\}$, differing paths can be generated by controlling these vertex weights. More formally, given a specified set of vertex weights $\{w_v(v_i)\}$, we wish to extract paths $\mathbb{P}$ for all vehicles:

$$\begin{aligned}
\mathbb{P} &= \arg\min_{\mathbb{P}'} \left(\sum_{k=1}^{n} c_{P'_k} + \mathcal{S}_v\right) \\
&= \arg\min_{\mathbb{P}'} \left(\sum_{i\neq j, \tilde{e}(v_i, v'_j)\in\mathbb{P}'} w_{\tilde{e}}(v_i, v'_j) + \sum_{\tilde{e}(v_i, v'_i)\notin\mathbb{P}'} w_{\tilde{e}}(v_i, v'_i)\right) \\
&= \arg\min_{\mathbb{P}} \left(\sum_{e(v_i, v_j)\in\mathbb{P}} w_e(v_i, v_j) + \sum_{v_i\notin\mathbb{P}} w_v(v_i)\right)
\end{aligned} \tag{11}$$

The above optimization rule attempts to simultaneously minimize the weights of edges *on the paths* $\mathbb{P} \in G$ as well as the weights of vertices *off the paths*. If the weight of each vertex $v_i \in V$ is set to be 0, the objective changes into the minimization of $\sum_{e(v_i, v_j)\in\mathbb{P}} w_e(v_i, v_j)$, which becomes a standard shortest path algorithm involving only weighted edges of a navigation graph.

Intuitively, paths defined by (11) tend to transit vertices with larger weights. This is because any positive weights of off-path vertices deteriorate the minimization objective while the on-path vertices have no effect to the objective, so passing a positively weighted vertex mitigates the deterioration. Therefore, we can regard the setting of large weights on selected vertices to be a form of external "force" to pull a nearby path to transit them.

## III. Weighting Policy and Deployment Algorithm

### A. Vertex Weighting Policy

Dispatch priority is inversely related to available onboard energy levels with highest priority given to those vehicles approaching critical energy depletion. We wish each low energy MAV to choose a path that passes a nearby charging station, even if it is off the shortest traversal path. For the simplest case, assume only one air vehicle must be deployed and its shortest path cost without concerning energy is

$$c_{P_0} = \sum_{e(v_i, v_j)\in P_0} w_e(v_i, v_j). \tag{12}$$

Further assume there is a charging station on a vertex $v_{ch}$ off the path $P_0$, and its vertex weight is $\delta$ and all other vertices are 0-weighted. If the aerial vehicle does not pass the charging station, then the overall cost in (11) is

$$c_{P_0} + \mathcal{S}_v = \sum_{e(v_i, v_j)\in P_0} w_e(v_i, v_j) + \delta. \tag{13}$$

In contrast, if the aerial vehicle pass the charging station $v_{ch}$ following a path $P_1 \neq P_0$, the overall cost becomes

$$c_{P_1} + \mathcal{S}_v = \sum_{e(v_i, v_j)\in P_1} w_e(v_i, v_j). \tag{14}$$

As a result of the minimization objective, a path switching from $P_0$ to $P_1$ only occurs if $c_{P_1} + \mathcal{S}_v < c_{P_0} + \mathcal{S}_v$, *i.e.*,

$$\delta > \sum_{e(v_i, v_j)\in P_1} w_e(v_i, v_j) - \sum_{e(v_i, v_j)\in P_0} w_e(v_i, v_j). \tag{15}$$

The subtraction on right hand side of (15) is the extra path cost for making a detour in order to reach a charging station, whereas $\delta$ can be regarded as the payoff for such choice. Therefore, a higher $\delta$ indicates a larger likelihood of path route change. However, $\delta$ cannot be arbitrarily large. Let $\mathcal{N}(v_{ch})$ denote the neighbors of $v_{ch}$ and vertex $v_{near} = \text{argmin}_v\{w_e(v_{ch}, v) | v \in \mathcal{N}(v_{ch})\}$ be the neighbor with the least connected edge weight, then,

***Theorem** 3.1:* Vertex $v_{ch}$ does not affect the nearby path $P_0$ if its weight satisfies $\delta > 2w_e(v_{ch}, v_{near})$ and if $v_{near} \notin P_0$.

*Proof:* If $v_{near} \notin P_0$, a closed path (cycle) $P_c$ between vertices $v_{ch}$ and $v_{near}$ in the transformed bipartite graph can be written as $v_{ch} \to v'_{near} \to v_{near} \to v'_{ch}$ where $v'_{ch}$ corresponds to $v_{ch}$ in the original untransformed navigation graph. Since $w_{\tilde{e}}(v_{near}, v'_{near}) = 0$, the total cost of $P_c$ is $c_c = w_{\tilde{e}}(v_{ch}, v'_{near}) + w_{\tilde{e}}(v_{near}, v'_{ch}) = 2w_e(v_{ch}, v_{near})$. Based on (13), if $\delta > 2w_e(v_{ch}, v_{near})$,

$$\begin{aligned}
c_{P_0} + \mathcal{S}_v &= \sum_{e(v_i, v_j)\in P_0} w_e(v_i, v_j) + \delta \\
&> \sum_{e(v_i, v_j)\in P_0} w_e(v_i, v_j) + 2w_e(v_{ch}, v_{near}).
\end{aligned} \tag{16}$$

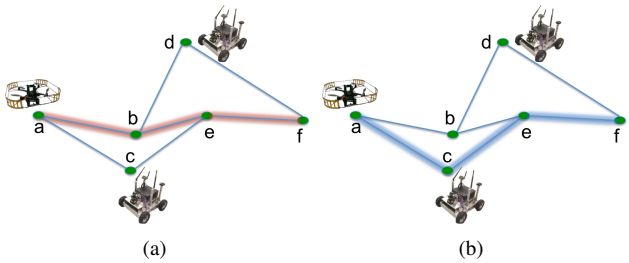


Fig. 2. Instead of following the shortest path to reach the destination, a low energy aerial vehicle detours to a nearby charging station to recharge.

Path $P_0$ eventually switches to the less costly $P_c$. However, $v_{ch}, v_{near} \notin P_0$, thus $P_0$ remains unchanged.

■

Theorem 3.1 bounds the range of $\delta$ within an interval $[0, 2w_e(v_{ch}, v_{near})]$. In practice, the weight $\delta$ is normalized by the importance of the corresponding charging station on this vertex. For instance, if a relay candidate is currently recharging on the charging station and its battery level is $90\%$, we set $\delta = 90\% \times 2w_e(v_{ch}, v_{near})$ since the unfinished task can be continued by the relay candidate with $90\%$-full energy.

Figure 2 illustrates an example. At vertex $a$ an aerial vehicle is alerted for low battery before it reaches its destination $f$. In between the current and goal locations there are two available charging stations $c$ and $d$ (with relay candidates) off the shortest path $P_0 : a \to b \to e \to f$, and $c$ and $d$ are on paths $P_c : a \to c \to e \to f$ and $P_d : a \to b \to d \to f$, respectively. If we raise the weights of $c$ and $d$ simultaneously from 0 to $\delta > 0$, once $\delta > \min(c_{P_c} - c_{P_0}, c_{P_d} - c_{P_0})$, the least cost charging station ($c$ in Fig. 2) will be selected to pass by.

### B. Online Deployment Algorithm

We use two thresholds $\tau$ and $\tau_0$ to classify the aerial vehicles into three categories ($0 < \tau_0 < \tau < 100\%$). Specifically, $\tau_0$ is a threshold that represents critically low energy, and $\tau$ is situated between $\tau_0$ and full charge. We assume that the charging stations are pre-deployed in the environment based on certain objectives (*e.g.*, maximization of communication coverage).

Firstly, an aerial vehicle that hits $\tau_0$ needs to stop executing ongoing tasks and search for the nearest energy resource. In such a case, charging stations become direct goal points. Secondly, if current battery level is in between $\tau_0$ and $\tau$, then the energy constrained path planning is carried out following aforementioned discussions. Finally, aerial vehicles with battery level above $\tau$ plan paths over the navigation graph with uniformly 0-weighted vertices. In this way, the produced paths have globally shortest path lengths which minimize flying energy consumption (11). The re-planning process is triggered when a task/goal is finished or when an aerial vehicle hits the $\tau_0$ battery threshold. Note, since the deployment/assignment algorithm has a time complexity of $O(|V|^3)$, the planning and re-planning can be carried out online.

**Algorithm III.1** Online Deployment Algorithm

```
/* Navigation graph G = (V, E) is built, charging platforms are deployed on G.*/
if Re-planning is triggered then
    if battery level is above τ then
        Aerial vehicles of this class plan the globally shortest paths toward the goals. Vertices are 0-weighted.
    if battery level is below τ0 then
        These aerial vehicles plan paths directly toward the nearest charging stations.
    if battery level is in [τ0, τ] then
        Aerial vehicles in this category plan energy-constrained paths on vertex-weighted G. Task is relayed if relay candidate has a better energy status.
```

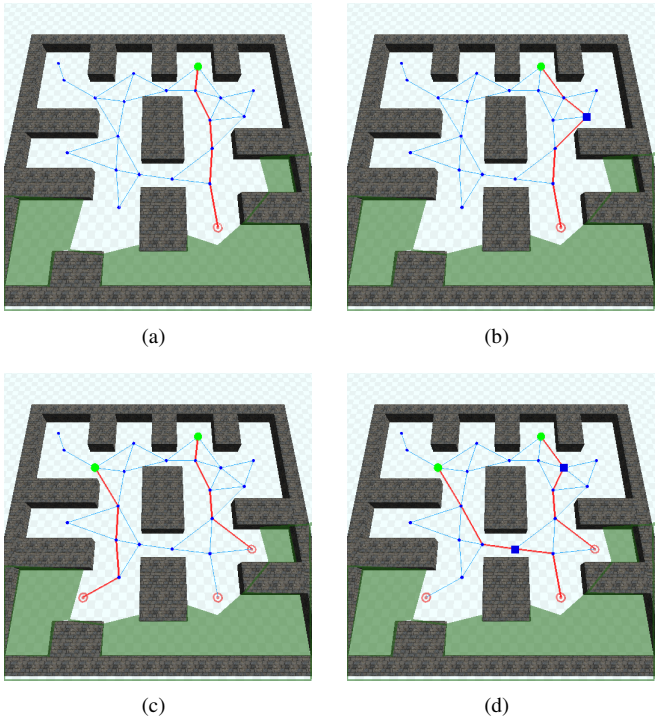


Fig. 3. Demonstration of path behaviors in an exploration scenario. White space denotes the area that has been explored; dark green areas represent the unexplored regions. Charging stations are the blue squares. (a) (c) The globally shortest paths; (b) (d) Paths are modified after embedding the energy constraints.

## IV. Simulation Results

Validation of this method is implemented in simulation. We wrote a custom simulator using *OpenGL* and ran it in a GNU/Linux environment. In our experiment, the navigation graph is simulated by generating vertices in obstacle-free space and edges that connect nearby vertices.

Figure 3 demonstrates the path changes caused by nearby charging stations. In Fig. 3(a), the path between the task-free vehicle (big green node on top) and a goal point (red target symbol at bottom) is the shortest and thus most energy-economic. Figure 3(b) shows an example that if the vertex weight of nearby charging station (big blue square node) is

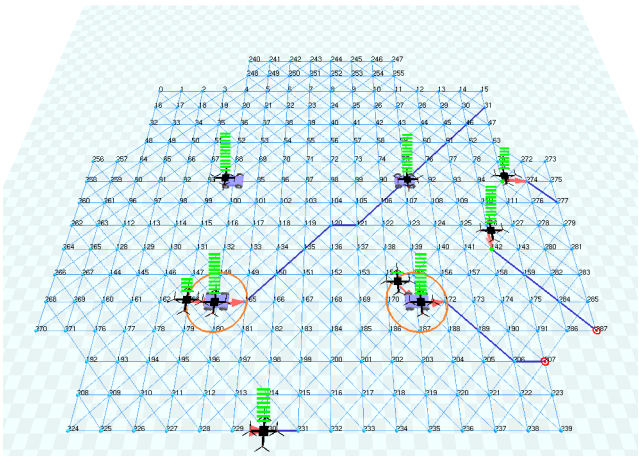

Fig. 4. Navigation paths on a larger scale navigation graph. Four charging stations are first deployed in the environment and are thereafter stationary. Nine aerial vehicles are deployed to the task locations randomly generated on the boundary of the graph. Upon task completion, new tasks are generated and re-planning is triggered. The green energy bar above each aerial vehicle is an indicator of the current battery status. Unfinished tasks are relayed to the relay-candidates (circled) that are involved in the same paths.

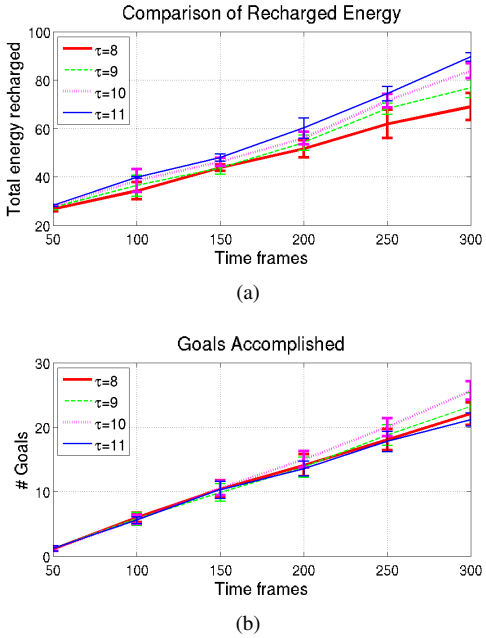


Fig. 5. (a) The total energy recharged by the aerial vehicles. (b) The total number of goal points accomplished by aerial vehicles. The maximum number of goals are reached when $\tau = 10$.

increased, a detour is made to reach the charging station first. Figures 3(c) and 3(d) illustrate the situation when multiple vehicles need to be simultaneously assigned. In this example, asymmetric case is considered, *i.e.*, the number of vehicles is not equal to the number of tasks ($n < m$). The result in Fig. 3(c) shows that two of the three target locations are optimally selected, and the globally shortest paths are produced. In Fig. 3(d), both low energy vehicles need to make detours to some nearby charging stations, and we can also see that, the goal point for the left vehicle is also changed due to energy consideration.

High fidelity simulations with more vehicles in larger scale navigation graphs have been conducted. Figure 4 shows a navigation graph consisting of $\sim$ 300 vertices that are well aligned to better reveal the path behaviors. At any moment at least 5 goal points are randomly generated on the border of the navigation graph, and each goal point requires an aerial vehicle to reach and serve it. Hence, there are redundant aerial vehicles. The relaying mechanism deploys the aerial vehicles of better energy conditions to fly to the goal points while keeping the low-energy ones on the charging stations*. The battery discharging curve is approximated as two piecewise linear segments where the first segment slowly decreases from 12V to 9V, and the second segment rapidly drops from 9V to 0V (the real curve can be found in [3]). The recharging curve here is assumed as a mirror/inverse process against the discharging curve.

With the proposed deployment and relay mechanisms, the limited recharging platforms are maximally used. This is different from the planning without energy constraints, where at some particular period a charging station might not recharge anything if all aerial vehicles have battery levels above $\tau_0$. Figure 5(a) shows the total energy (in voltage) obtained by all aerial vehicles. We can see that with a larger $\tau$, totally more energy is recharged from the charging stations. More recharged energy reflects longer flying time for accomplishing the missions.

Figure 5(b) shows the number of goals served/reached by the aerial vehicles. The curves manifest themselves that a too large $\tau$ produces bad results however. This is because a too large $\tau$ implies a too conservative planning strategy, which generates unnecessary detours to reach the charging stations even if the current energy is sufficient.

To investigate the path behaviors, we compare the path costs with regard to varying vertex weights, see Fig. 6(a). Specifically, as the vertex weights gradually increase, the total path cost (for aerial vehicles in the current deployment) grows too. Both Figs. 6(a) and 6(b) are histograms with bundles of grouped bars, with each bundle containing five bars, representing different numbers of available charging stations in the system. (From left to right, the numbers are 2 to 10 with an increment of 2.) Figure 6(b) shows the relationship between the average number of relays on each path versus the vertex weights. This also indicates that, even though the weights can be tuned to be infinitely large, it is not necessary that all charging stations will be included in the paths, which validates Theorem 3.1.

## V. Discussion, Future Work, Conclusion

We present the approach under a centralized framework for a clear description purpose. In practice, communication among vehicles might be limited by hardware and space. The proposed method can also be decentralized as long as the navigation graphs are constructed locally. Local navigation

*A video of the simulation results is available at `http://tinyurl.com/LiuICUAS2014`

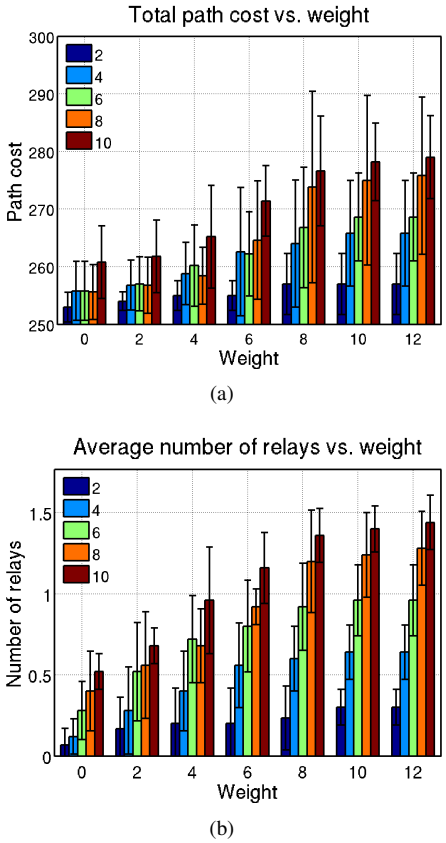


Fig. 6. The effects of vertex weighting on the total path cost and the number of relays. The different bar colors represent the different number of charging stations.

graphs allow the navigation to be planned in a localized manner, and the merging of local graphs allows better quality routing solutions to be obtained.

In future work, we are interested in investigating more complex weighting rules for the charging stations. By further identifying the diversity of different charging stations (*e.g.*, their locations, distribution patterns, and the battery status of relay candidates on them), better quality navigation paths might be obtained. In addition, we also wish the ground vehicles to be more active rather than staying stationary at some pre-deployed positions: a ground vehicle can also move toward the low energy vehicle after the energy supply assignment is made.

To conclude, this paper presents a new planning strategy that embeds the energy constraints in the multi-MAV deployments. The method integrates the path planning and optimal assignment problems through transforming graphs of two different types. Navigation paths involving energy supplies are produced via a strategic vertex weighting scheme. Simulation results are provided for a validation of the proposed method.

## REFERENCES


[1] B. Bethke, J. P. How, and J. Vian. Group health management of UAV teams with applications to persistent surveillance. In *Proc. of the Amer. Control Conf.*, pages 3145–3150, Seattle, WA, June 2008.

[2] G. Chaohui, S. Tully, G. Kantor, and H. Choset. Multi-agent deterministic graph mapping via robot rendezvous. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, pages 1278–1283, Saint Paul, MN, May 2012.

[3] J. Derenick, N. Michael, and V. Kumar. Energy-aware coverage control with docking for networked robots. In *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, pages 3667–3672, San Francisco, CA, Sept. 2011.

[4] H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistic Quarterly 2:83–97*, 2:83–97, 1955.

[5] L. Liu and D. A. Shell. Physically Routing Robots in a Multi-robot Network: Flexibility through a Three Dimensional Matching Graph. *Intl. J. Robot. Research*, 32(12):1475–1494, 2013.

[6] N. Mathew, S. L. Smith, and S. L. Waslander. A Graph-Based Approach to Multi-Robot Rendezvous for Recharging in Persistent Tasks. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, pages 3497–3502, Karlsruhe, Germany, May 2013.

[7] L. E. Parker. Multiple Mobile Robot Systems. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*, chapter 40. Springer, 2008.

[8] M. R. K. Ryan. Exploiting subgraph structure in multi-robot path planning. *J. Artif. Intell. Res.*, 31:497–542, 2008.

[9] S. Sariel and T. R. Balch. Efficient bids on task allocation for multi-robot exploration. In *FLAIRS Conf.*, pages 116–121, 2006.

[10] S. Shen, N. Michael, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained MAV. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, pages 20–25, Shanghai, China, May 2011.

[11] K. Sundar and S. Rathinam. Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots. *IEEE Trans. Autom. Sci. Eng.*, 11(1):287–294, Jan. 2014.

[12] T. Toksoz, J. Redding, M. Michini, B. Michini, J. P. How, M. Vavrina, and J. Vian. Automated Battery Swap and Recharge to Enable Persistent UAV Missions. In *AIAA Infotech@Aerospace Conference*, 2011. (AIAA-2011-1405).

[13] R. Vincent, D. Fox, J. Ko, K. Konolige, B. Limketkai, B. Morisset, C. Ortiz, D. Schulz, and B. Stewart. Distributed multirobot exploration, mapping, and task allocation. *Annals of Mathematics and Artificial Intelligence*, 52(2-4):229–255, 2008.

[14] K. M. Wurm, C. Stachniss, and W. Burgard. Coordinated multi-robot exploration using a segmentation of the environment. In *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, pages 1160–1165, 2008.

[15] R. M. Zlot, A. Stentz, M. B. Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, volume 3, pages 3016 – 3023, May 2002.