Using Gradient Descent to Optimize Paths for Sustaining Wireless Sensor Networks

Srikanth K. V. Sudarshan and Aaron T. Becker
Dept. of Electrical and Computer Engineering
University of Houston,
Houston, TX 70004, USA
Email: skvenkatasudarshan@uh.edu, atbecker@uh.edu

***Abstract*—A structural-health wireless sensor network (WSN) should last for decades, but traditional disposable batteries cannot sustain such a network. Energy is the major impediment to sustainability of WSNs. Most energy is consumed by (i) wireless transmissions of perceived data, and (ii) long-distance multi-hop transmissions from the source sensors to the sink. This paper explores how to exploit emerging wireless power transfer technology by using robotic unmanned vehicles (UVs) to service the WSNs. These UVs cut data transmissions from long to short-distances, collect sensed information, and replenish WSN's energy. This paper presents path-planning and path optimization algorithms for sustaining WSNs.**

Keywords*—*Wireless sensor networks, wireless recharge, robot, unmanned vehicles

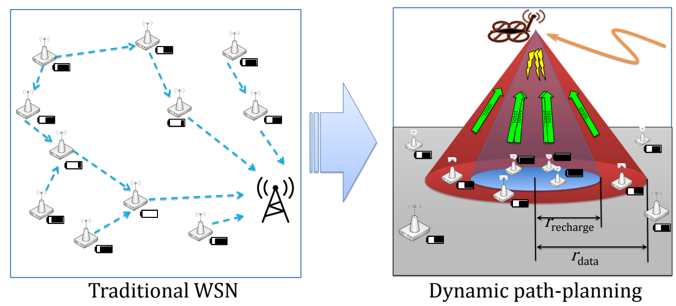


Fig. 1: Evolution from traditional wireless sensor networks (WSNs) to servicing WSN with UV(s). We present path-planning techniques that use unmanned vehicles (UVs) to gather aggregated data and recharge sensors.

I. Introduction

New wireless sensor technologies have enabled wireless sensor networks (WSNs) to proliferate in many different fields (e.g., battlefield surveillance, environmental sensing, biomedical observation [1], [17], [20], [28]). Although advances in processing and computing designs can endow sensors with a multitude of sensing modalities (temperature, pressure, light, magnetometer, infrared, etc.), advances in battery technology have been more modest. Energy constraints on battery-powered sensors limits the sustainability of WSNs. In WSNs, the majority of energy is consumed by (i) wireless transmission of perceived data [18], [28], and (ii) long-distance multi-hop transmissions from source sensors to the sink. Radio transmission and listening dominate power usage, as shown in Fig./ 2. Research efforts to address WSN energy concerns have focused on energy conservation [8], environmental energy harvesting [12], [27] and incremental sensor deployment [37]. However, energy conservation schemes only slow energy consumption, not compensate energy depletion. Harvesting environmental energy, such as solar, wind and vibration, is subject to their availability, and is often uncontrollable. Incremental sensor deployment makes WSNs neither sustainable nor environmentally friendly, since most disposable sensors' batteries contain cadmium, lead, mercury, copper, zinc, manganese, lithium, or potassium [10]. These heavy metals "*can leach into soil and water, polluting lakes and streams, making them unfit for drinking, swimming, fishing, and supporting wildlife, and even posing hazards to human health*" [9].

Fortunately, recent breakthroughs in the area of wireless power transfer technologies (e.g. inductive coupling, magnetic resonant, and RF energy harvesting) [23] provide promising

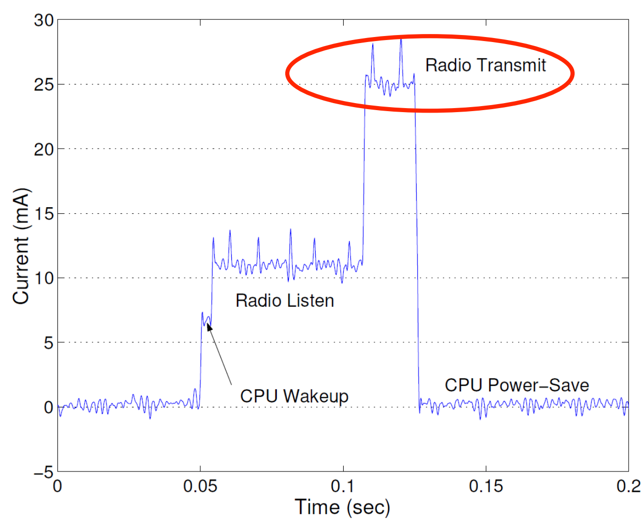


Fig. 2: Power usage in a wireless sensor node is dominated by transmission costs and listening costs. Figure modified from [29].

alternatives for deploying such WSNs. *Magnetic resonant wireless power transfer* [23] can wirelessly transfer electric power from the energy storage device to the receiving device efficiently within medium range (40% efficiency within 2 meters). It is also insensitive to the neighboring environment and does not require a line of sight between the charging and receiving devices. Researchers proposed that a mobile unmanned vehicle (UV) carrying a wireless charging device could visit and recharge each sensor to sustain a WSN [35].

However, one UV may not be able to visit every sensor if the WSN is deployed in harsh environments/terrains (e.g. dense forest, mountains, underwater), or the WSN is large-scale, consisting of a great number of sensors. Although these seminal studies replenished sensor energy, most of the energy was still wasted by long-distance wireless transmissions of perceived data, especially by relaying sensors. Due to charging and travel time of the UV, some bottleneck sensors may drain their residual energy while waiting for the UV. Great unsolved challenges on control remain, including how to select the optimal path for the UV to travel within WSNs and how to efficiently dispatch multiple UVs to recharge WSNs.

Assigning sensors to UVs using matching theory often assumes that energy costs due to power transmission greatly exceed the UV's transportation costs. This assumption might not fit for WSNs spread over large geographic areas, or terrain with obstacles, or where transportation costs are high, such as subsea or aerial UVs. This paper focuses on algorithms that make such WSNs sustainable by focusing on path-planing, trajectory optimization, and responding to dynamic network conditions, as shown in Fig. 1.

II. Related Work

The path-planning problem for UVs has been investigated from several angles. To minimize path length, the authors in [7] survey the multiple-Traveling Salesman Problem, itself a generalization of the vehicle routing problem [11]. Servicing a WSN is closely related to coverage problems, recent work includes methods for optimizing speed along given routes [30], and techniques to continually improve existing routes [31].

Much work has focused on the data ferrying problem, from minimizing the latency between visits to nodes [2], to maximizing the total data rate from sensors to sink using UVs [19], to minimizing overall delay while sharing bandwidth [16], to having a set schedule and opportunistically deviating from it [15].

Finally, using unmanned aerial vehicles to recharge other robots or sensor nodes has focused on physical design, which includes direct contact, such as swapping batteries [33], [34] or direct recharge [26], wireless resonant coupling [14], [21], and electromagnetic radiation [36], and algorithmic improvements using graph theory [25], linear programming [30], and gradient descent optimization [31].

III. Overview

This paper's goal is to explore path-optimization techniques. Previous work often uses optimization/matching theory to assign one UV/multiple UVs to WSN nodes, and use a Hamiltonian cycle to visit each node. This is reasonable if recharging nodes is the largest component of a UV's energy budget: $E_{\text{recharge}}|\text{nodes}| \gg \text{E}_{\text{movement}} * \text{path\_length}$. If this assumption is violated, path-planning becomes the key concern. A simplified form of this decision is written as

$$K_{dist} = \frac{E_{\text{movement}} * \text{path\_length}}{E_{\text{recharge}}|\text{nodes}|}. \tag{1}$$

Here K_{dist} represents the *tipping point*, the variable where the decision problem becomes fundamentally different. If K_{dist} is small, path-planning is inconsequential, and almost any solver is sufficient. However, when K_{dist} is large path-planning becomes the key consideration. Our eventual goal is to design full trajectories that optimize the path of each UV, by servicing multiple nodes simultaneously. However, even just the path-planning component is NP-hard [5]. To make progress, this paper focuses on path optimization techniques.

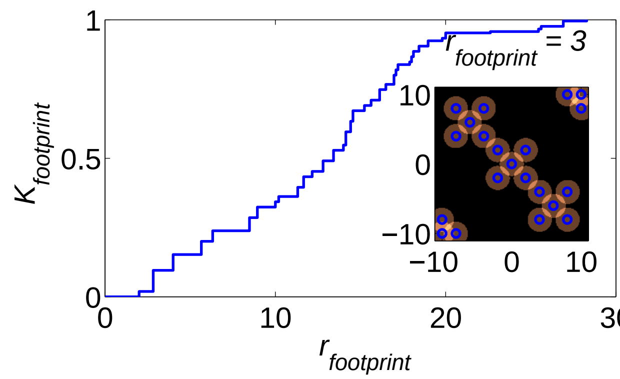


Fig. 3: As the recharging footprint or data-transfer footprint $r_{footprint}$ grows, more sensors can be recharged simultaneously. The plot above shows $K_{footprint}(r_{footprint})$, calculated by (2).

A UV has an associated recharging footprint and a data-transfer footprint, which can often be modeled as disks of radius $r_{recharge}$ and r_{data}, as illustrated in Fig. 4. If sensor nodes are clustered, a UV can service multiple clients simultaneously.

We represent the fraction of sensors that are clustered as

$$K_{footprint} = \frac{2}{N^2 - N} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(\|p_i - p_j\|_2 \leq r_{footprint} \right). \tag{2}$$

Here, p_i is the position of the ith node, and there are N nodes. $K_{footprint}$ is shown in Fig. 3 for a representative network.

In general, energy-efficient recharging requires closer proximity than data transmission, so this implies there are two tipping points related to node density, $K_{recharge}$, and K_{data}. Correspondingly, the WSN recharge problem has three regimes with differing solutions. Before the tipping points, nodes are sparse and not clustered. In this regime optimal paths are straight lines from node to node, and the optimal solution is a variant of the traveling salesman problem. As sensors get closer together, the optimal path may be *between* one or more sensors. In Fig. 4, path **A** is designed to visit each node, but path **B** is designed to recharge all nodes. Here, the optimal solution is often to weave between clusters of nodes. The third regime is when many nodes are close enough for transfer data, as shown in path **C**. The simulations in this paper take advantage of the non-zero $r_{footprint}$ to allow the UVs to pass near sensors without requiring them to visit each node.

IV. Path Optimization Algorithm

Our solution designs a closed-loop path that intersects the origin for each UV. The base technique is a variant of Lloyd's algorithm [6], [24]. Each path is represented by a finite number of waypoints, and these waypoints are both attracted to the

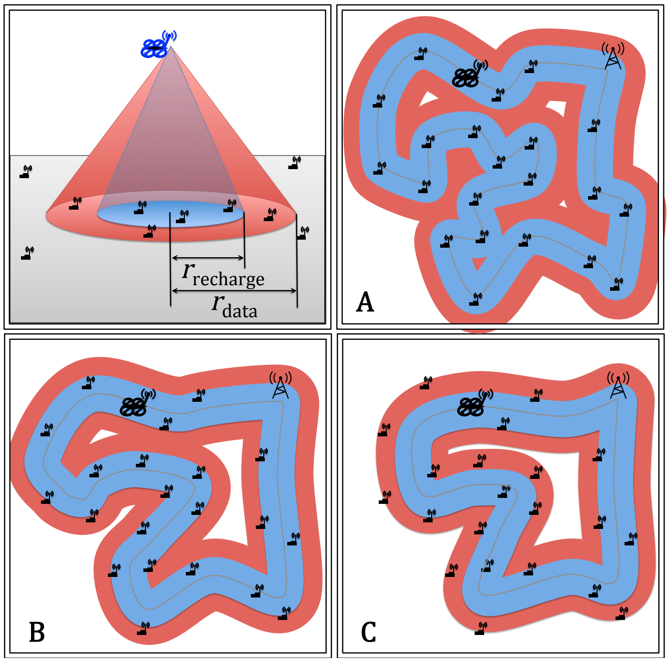
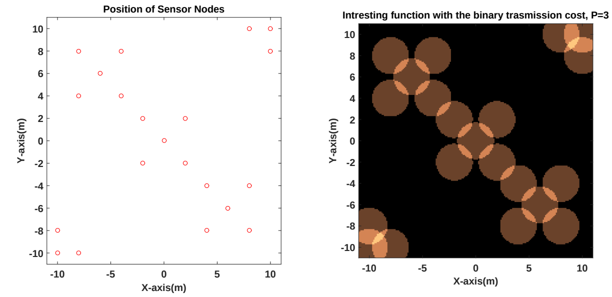


Fig. 4: A UV has an associated recharging footprint and a data-transfer footprint, which can often be modeled as disks of radius $r_{recharge}$ and r_{data}. Path **A** visits each node, but path **B** is shorter because it is designed merely to recharge all nodes. Path **C** is the least tortuous because it is designed to transfer data from all nodes, and $r_{data} > r_{recharge}$.



(a) (b)

Fig. 5: Function (3) determines the range from which a UV can service a sensor node.

centroid of all sensor nodes within their Voronoi cell, and attracted to their neighboring waypoints. The following sections describe how this path is initialized (IV-A), and then how the path is optimized by switching between a gradient descent optimization routine that finds local minimas (IV-B), and a genetic algorithm that rearranges the order of waypoints to improve the paths (IV-C). Our MATLAB implementation is available at mathworks.com/matlabcentral/fileexchange/49863 [32].

A. Initializing Path with Hilbert Curve

It is important to have an initial path that fills the map. This ensures the UVs will visit every node. We adapt the space-filling *Hilbert Curve*, which creates a fractal path that fills up a unit area space and serves as an initial path for the first iteration [13]. Figure 7 shows an initial path using a Hilbert curve. With multiple UVs, the Hilbert curve is scaled by a scalar to ensure the waypoints are unique, as shown in Fig. 8.

B. Gradient Descent on a Path Composed of Waypoints

The following algorithm is derived from [31], which focused on local optimization techniques that gradually improve the paths followed by robots during persistent tasks. This technique is amenable to WSN.

Consider N UVs servicing a Wireless Sensor Network in a convex, bounded area $\mathbf{Q} \subset \mathbb{R}^2$. Waypoints are a set of points that define the path for each UV. The UV travels in a straight line in between two neighboring waypoints. Let $\mathbf{p}_i^r$ be the position of the i^{th}, $i \in (1 \ldots n(r))$ waypoint of the r^{th} UV. Servicing includes recharging the nodes and collecting a part of the data that the nodes are about to transmit to the sink, thereby reducing the power expenditure in the sensor nodes. The algorithm forms a locally-optimal path to visit the sensor nodes in the WSN. At each step, we compute the Voronoi partition $\mathbf{V}_i^r$ defined by the waypoints, with one partition assigned to each waypoint.

We require a function $\phi(\mathbf{q})$ that designates how many sensor nodes can be recharged from the position $\mathbf{q}$. It tells the positions of the sensors and indicates the $r_{footprint}$ of each sensor. This information is necessary for minimizing the distance traveled by the UV and, when incorporated into the map, provides information for charging multiple sensor nodes from a single location. For the simulation experiments we use a binary $\phi(\mathbf{q})$. A simple function is used but $\phi(\mathbf{q})$ could also account for parameters including the maximum height at which the robot starts charging and the rate of charging to better mimic real time performance. Here $\phi(\mathbf{q})$ calculates the distance between each sensor node and $\mathbf{q}$ and indicates if this distance is less $r_{footprint}$.

$$\phi(\mathbf{q}) = \sum_{s=1}^{k} (x_s - \mathbf{q}_x)^2 + (y_s - \mathbf{q}_y)^2 < r_{footprint} \tag{3}$$

In Fig. 5 (x_s, y_s) correspond to the xy location of sensor s and $(\mathbf{q}_x, \mathbf{q}_y)$ correspond to the locations $\mathbf{q} \subset \mathbf{Q}$.

The cost function for each path is given by:

$$\begin{aligned}\mathbf{H} = \sum_{r=1}^{N}\sum_{i=1}^{n(r)} \int_{V_i^r} \frac{W_s}{2}\|\mathbf{q} - \mathbf{p}_i^r\|^2 \phi(\mathbf{q}) d\mathbf{q} \qquad + \\ \sum_{r=1}^{N}\sum_{i=1}^{n(r)} \frac{W_n}{2}\|\mathbf{p}_i^r - \mathbf{p}_{i+1}^r\|^2 \end{aligned} \tag{4}$$

W_s, W_n are positive scalar constants that are used to weight the sensing and neighbor distance, respectively, and depend on the experimental setup. The gradient descent algorithm minimizes the two-part cost function (4). The first part of the equation indicates a waypoint in a region far from sensor nodes is costly and the second part indicates having neighboring waypoints far away is also costly. A minimizing solution is a short path that mostly travels near WSNs.

We compute the mass, mass-moment, and centroid of the V_i^r (Voronoi partition for i^{th} waypoint of the r^{th} UV) as follows:

$$
\begin{aligned}
M_i^r &= \int_{V_i^r} \phi(\mathbf{q}) d\mathbf{q}, \quad \mathbf{L}_i^r = \int_{V_i^r} \mathbf{q}\phi(\mathbf{q}) d\mathbf{q}, \\
\mathbf{C}_i^r &= \frac{\mathbf{L}_i^r}{M_i^r}
\end{aligned} \tag{5}
$$

The control law for each waypoint is the summation of forces that pulls the waypoint toward the centroid of the Voronoi partition (weighted by $\phi(\mathbf{q})$):

$$
\mathbf{u}_i^r = \frac{K_i^r(M_i^r \mathbf{e}_i^r + \boldsymbol{\alpha}_i^r)}{\beta_i^r} \tag{6}
$$

Here, K_i^r is a positive definite matrix and is potentially-time varying. $\mathbf{e}_i^r = \mathbf{C}_i^r - \mathbf{p}_i^r$, the error, introduces the first primitive by obtaining the difference between the waypoint position and the weighted centroid of the Voronoi region. This tries to move the waypoint towards the interesting region, reshaping the path of the robot. The second term $\boldsymbol{\alpha}_i^r = W_n(\mathbf{p}_{i+1}^r + \mathbf{p}_{i-1}^r - 2\mathbf{p}_i^r)$ introduces the second primitive which pulls the neighboring waypoints together to obtain a short path. $\boldsymbol{\beta}_i^r = \mathbf{M}_i^r + 2\mathbf{W}_n$ normalizes the weight distribution between servicing sensors and staying close to neighboring waypoints.

The control is then applied to this waypoint to update its position:

$$
\mathbf{p}_i^r(k) = \mathbf{p}_i^r(k-1) + \mathbf{u}_i^r \tag{7}
$$

The control algorithm is described in Alg. 1, and is called for each waypoint in turn.

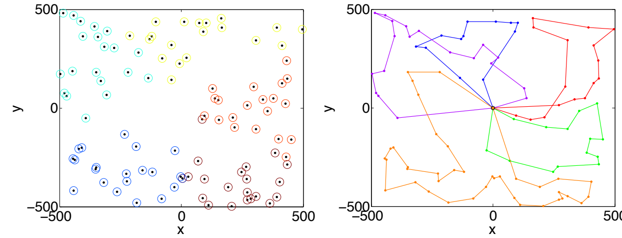
Algorithm 1 Gradient descent path optimization for the i^{th} waypoint $\mathbf{p}_i^r$ in robot r's path in a known environment (from [30], implemented at [32]).

Require: Ability to calculate Voronoi partition
Require: Knowledge of the location of neighboring waypoints $\mathbf{p}_{i-1}^r$ and $\mathbf{p}_{i+1}^r$

```
loop
    Compute the waypoints Voronoi partition
    Compute C_i according to (5)
    Obtain neighbor waypoint locations p^r_{i-1} and p^r_{i+1}
    Compute u^r_i according to (6)
    Update p^r_i according to (7)
```

C. multiple-Traveling Salesman Problem (mTSP)

The gradient descent algorithm (Alg. 1) can get stuck in local minima. To further improve the path we input the location of the waypoints obtained after running the gradient descent algorithm into a multiple-Traveling Salesman Problem (mTSP) search algorithm. Given a list of cities to visit, the classic *traveling salesman problem* (TSP) attempts to find an ordering of the cities that minimizes the total distance on a tour that visits all the cities once [3]. The solution is the shortest Hamilton cycle. By labeling our sensor nodes as cities, the solution to the traveling salesman problem gives the shortest length path. The mTSP solution straightens out loops in the path and can reduce the cost function. This often moves the solution out of the local minimum obtained after the executing the gradient descent algorithm.



(a) nodes partitioned by angle (b) near-optimal solution.

Fig. 6: Screenshots from mTSP solver aided by heuristic. Left, nodes partitioned according to angle from sink. Right, near-optimal solution from a mTSP solver aided by our heuristic.

This problem is NP-hard (Non-Deterministic Polynomial), but many powerful heuristics are available, and software packages can provide answers for tens of thousands of nodes (e.g., the Concorde TSP Solver [4]). A solution with multiple salesman is called a mTSP. The mTSP is still an NP-hard problem [5], so the solutions returned by the search algorithm given limited time may not be the global optimum.

A good heuristic can increase TSP solver performance. In our numerical simulations, priming an open-source genetic algorithm solver [22] by sorting the nodes by angle from the sink and dividing the sorted list equally between the UVs decreased path costs by 20%. Fig. 6 shows results from our simulation with 100 nodes and 5 UVs.

V. Results

We developed a MATLAB simulation using the three algorithms described in Section IV. The code is available at [32]. The next two sections describe the results with one UV and with multiple UVs.

A. One UV

A single UV system was simulated using MATLAB in Fig. 7. The initial path at iteration 0 was set to be a space-filling Hilbert's curve, to identify the location of the sensor nodes. If the UV follows this initial path it can learn the value function $\phi(\mathbf{q})$, and identifying the interesting points for the whole map. A waypoint at $[0, 0]$ is stationary, this represents the sink where the UV recharges and unloads data collected while servicing the sensor nodes. For the first 100 iterations the gradient descent algorithm moves the path waypoints, using the Voronoi diagram to identify locations with high sensory information. The path achieved after 100 iterations is usually a local-optimum and iterating further does not decrease the cost function. This sub-optimal solution obtained often contains loops. To further optimize our path and escape this local-optimum the path is inputted into a mTSP (multiple-Traveling Salesman Problem) solver for the next 100 iterations, this straightens the loops by reconnecting the waypoints without changing the waypoint location, $\mathbf{p}_i^r$. After 200 iterations the cost function has decreased due to straightening of the loops by the mTSP solver. These two algorithms are called in succession to optimize the cost function depending on the time available for calculation or until the cost function asymptotically converges. The cost function is shown in Fig. 9(a). This cost function monotonically decreases.

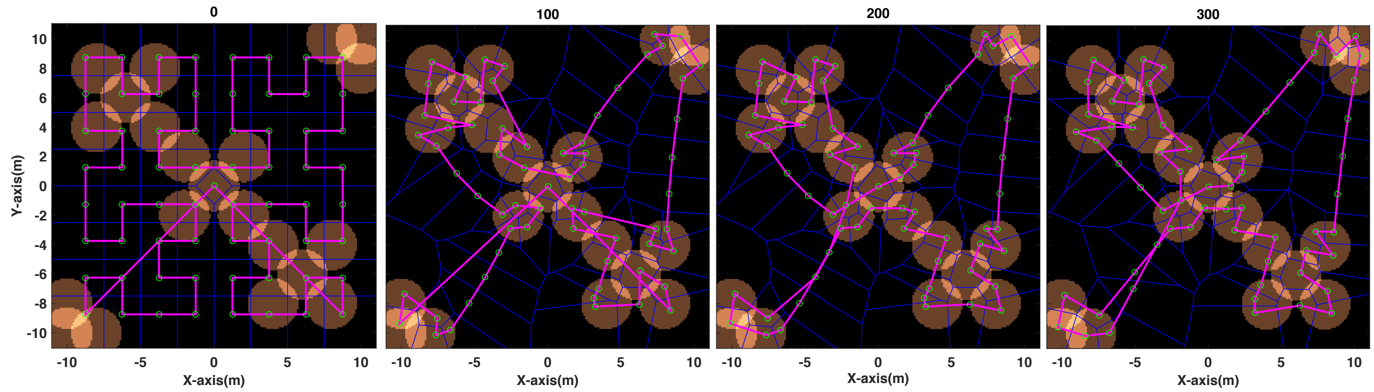


Fig. 7: Simulation results of optimization algorithm in Section IV for one UV. 1.) Iteration 0: Hilbert's Space Filling Curve. 2.) Iteration 100: gradient descent algorithm (first). 3.) Iteration 200: mTSP solver (first). 4.) Iteration 300: gradient descent algorithm (second). The waypoints are indicated by a set of linked ○ markers, the associated Voronoi diagram is in blue, the magenta lines represent the path the UV follows for servicing the sensor nodes, and the underlying density plot represents the interesting regions generated by the sensor nodes.

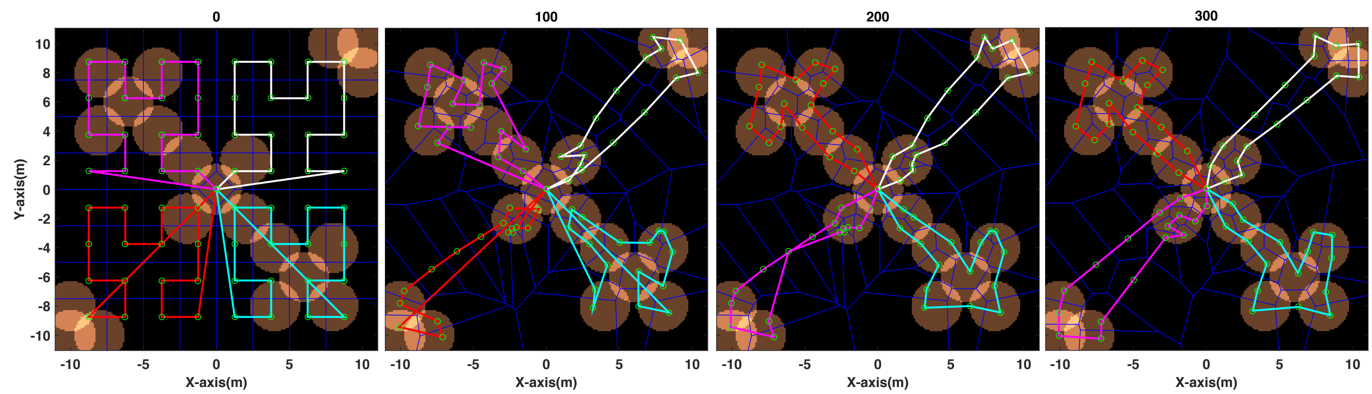
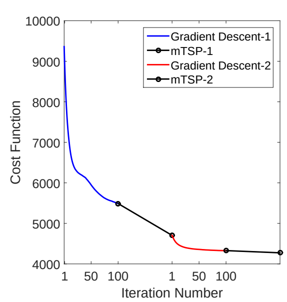
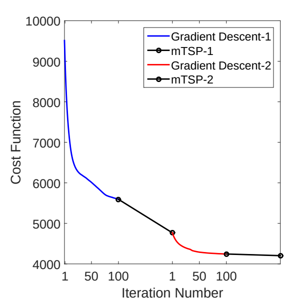


Fig. 8: Simulation results of optimization algorithm in Section IV for multiple UVs. 1.) Iteration 0: Hilbert's Space Filling Curve. 2.) Iteration 100: gradient descent algorithm (first). 3.) Iteration 200: mTSP solver (first). 4.) Iteration 300: gradient descent algorithm (second). The waypoints are indicated by green ○ markers, the associated Voronoi diagram is in blue, the white, cyan, magenta, and red lines represent the path the UVs follow for servicing the sensor nodes, and the underlying density plot represents the interesting regions generated by the sensor nodes.



(a) Cost-function plotted for the single robot case.



(b) Cost-function plotted for the four robot case.

Fig. 9: Cost function indicates a decreasing trend approving the optimization algorithm.

B. Multiple UVs

A two UV system was simulated on MATLAB. As shown in Fig. 8, UVs service different sets of nodes in the WSN. A multi-UV system is practical because a single UV might not be able to handle a large network. Similar to the one UV case, a space-filling Hilbert's curve is used to initialize the paths. Each path contains a stationary waypoint at $[0, 0]$, representing the sink where both UVs recharge and unload data collected while servicing the sensor nodes. The sensor nodes were placed in random locations to verify the robustness of the algorithm. The algorithm proceeds in a similar fashion to the one UV case. The simulation results show the optimization of the path and the minimization of the cost function.

VI. Conclusion and Future Work

An optimized path-planning algorithm was simulated for servicing a WSN. The path constructed is adaptive to the sensor node locations. The simulations above used a static WSN, but often sensor data transmission is dependent on transient phenomena. For example, a swarm of subsea sensors may track a school of fish, the progress of an oil slick, or seasonal drift of ocean currents. These are time-varying phenomena, and so the UV servicing the sensors should be able to adapt.

The same local optimization techniques presented in this paper can iteratively adapt the paths of UVs. A schematic of our the adaptive control law is shown in Fig. 10. Future work should extend our simulation to handle non-stationary sensor nodes, improve the convergence rate, and use our mTSP code to escape local minimal. We are in the process of implementing

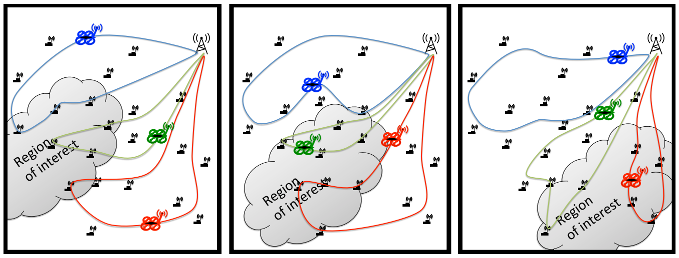


Fig. 10: The grey cloud represents a time-varying region of interest. Allowing the UVs to dynamically modify their routes in a distributed manner enables a robust response to changing conditions while maintaining service.

the algorithm on mobile-robots, with eventual implementation with a set of quadcopters.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks (Elsevier) Journal*, 38(4):169–181, September 2002.

[2] Soroush Alamdari, Elaheh Fata, and Stephen L Smith. Persistent monitoring in discrete environments: Minimizing the maximum weighted latency between observations. *Int. J. Rob. Res.*, 33(1):138–154, January 2014.

[3] David Applegate, Robert Bixby, Vašek Chvátal, and William Cook. *Finding tours in the TSP*. Citeseer, 1999.

[4] David Applegate, William Cook, Sanjeeb Dash, and André Rohe. Solution of a min-max vehicle routing problem. *INFORMS Journal on Computing*, 14(2):132–143, 2002.

[5] Esther M Arkin, Sándor P Fekete, and Joseph SB Mitchell. Approximation algorithms for lawn mowing and milling. *Computational Geometry*, 17(1):25–50, 2000.

[6] Aaron Becker. "Lloyd's Algorithm." MATLAB Central File Exchange, April 2013.

[7] Tolga Bektas. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3):209–219, 2006.

[8] F. Bouabdallah, N. Bouabdallah, and R. Boutaba. Cross-layer design for energy conservation in wireless sensor networks. In *Proc. of IEEE International Conference on Communications, ICC 2009)*, Dresden, Germany, June 2009.

[9] Natural Resources Defence Council. Proper disposal of batteries, electronics, and hazardous waste.

[10] The Green Cross and Blacksmith Institute. *The world's worst pollution problems 2012: assessing health risks at hazardous waste sites*. Blacksmith Institute, Zurich/New York, 2012.

[11] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.

[12] Kai-Wei Fan, Zizhan Zheng, and Prasun Sinha. Steady and fair rate allocation for rechargeable sensors in perpetual sensor networks. In *Proc. of the ACM conference on Embedded network sensor systems, SenSys 2008*, New York, NY, Novomber 2008.

[13] Fritz Peter Fischer. *Über die Darstellung der Hornhautoberfläche und ihrer Veränderungen im Reflexbild*. Springer, 1928.

[14] Brent Griffin and Carrick Detweiler. Resonant wireless power transfer to ground sensors from a uav. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2660–2665. IEEE, 2012.

[15] Shimin Guo, Majid Ghaderi, Aaditeshwar Seth, and Srinivasan Keshav. Opportunistic scheduling in ferry based networks. 2006.

[16] Shimin Guo and Srinivasan Keshav. Fair and efficient scheduling in data ferrying networks. In *Proceedings of the 2007 ACM CoNEXT Conference*, CoNEXT '07, pages 13:1–13:12, New York, NY, USA, 2007. ACM.

[17] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff. On energy provisioning and relay node placement for wireless sensor networks. *IEEE Transactions on Wireless Communications*, 4(5):2579–2590, September 2005.

[18] C. Hua and T. P. Yum. Optimal routing and data aggregation for maximizing lifetime of wireless sensor networks. *IEEE/ACM Transactions on Networking*, 16(4):892–903, August 2008.

[19] Zanjie Huang, Hiroki Nishiyama, Nei Kato, Fumie Ono, and Ryu Miura. Resource allocation for data gathering in uav-aided wireless sensor networks. In *IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC 2014)*, Beijing, China, September 2014.

[20] T. Jian, H. Bin, and S. Arunabha. Relay node placement in large scale wireless sensor networks. *Computer Communications*, 29(4):490–501, February 2006.

[21] Jennifer Johnson, Elizabeth Basha, and Carrick Detweiler. Charge selection algorithms for maximizing sensor network life with uav-based limited wireless recharging. In *Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on*, pages 159–164. IEEE, 2013.

[22] Joseph Kirk. Fixed start/end point multiple traveling salesmen problem - genetic algorithm (matlab code), May 2014.

[23] Andr Kurs, Aristeidis Karalis, and Robert Moffatt. Wireless power transfer via strongly coupled magnetic resonances. *Science*, 317(5843):83–86, June 2007.

[24] Stuart Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[25] Neil Mathew, Stephen L Smith, and Steven Lake Waslander. A graph-based approach to multi-robot rendezvous for recharging in persistent tasks. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3497–3502. IEEE, 2013.

[26] Yash Mulgaonkar. *Automated recharging for persistence missions with multiple micro aerial vehicles*. PhD thesis, University of Pennsylvania, 2012.

[27] Chulsung Park and P.H. Chou. Ambimax: Autonomous energy harvesting platform for multi-supply wireless sensor nodes. In *IEEE Sensor and Ad Hoc Communications and Networks, SECON 2006*, Reston, VA, September 2006.

[28] Y. Shi, L. Xie, Y. T. Hou, W. Lou, and H. D. Sherali. On renewable sensor networks with wireless energy transfer: The multi-node case. In *Proc. of IEEE International Conference on Computer Communications, INFOCOM 2011*, Shanghai, China, April 2011.

[29] Victor Shnayder, Mark Hempstead, Bor rong Chen, Geoff Werner Allen, and Matt Welsh. Simulating the power consumption of large-scale sensor network applications. In *ACM Sensys*, Baltimore, MD, November 2004.

[30] Stephen L Smith, Mac Schwager, and Daniela Rus. Persistent robotic tasks: Monitoring and sweeping in changing environments. *Robotics, IEEE Transactions on*, 28(2):410–426, 2012.

[31] Daniel E Soltero, Mac Schwager, and Daniela Rus. Decentralized path planning for coverage tasks using gradient descent adaptive control. *The International Journal of Robotics Research*, page 0278364913497241, 2013.

[32] Srikanth Kandanuru Venkata Sudarshan and Aaron T. Becker. "Decentralized Path Planning For Coverage using Gradient Descent Adaptive Control." MATLAB Central File Exchange, February 2015.

[33] Kart A Swieringa, Clarence B Hanson, Johnhenri R Richardson, Jonathan D White, Zahid Hasan, Elizabeth Qian, and Anouck Girard. Autonomous battery swapping system for small-scale helicopters. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3335–3340. IEEE, 2010.

[34] Tuna Toksoz, Joshua Redding, Matthew Michini, Bernard Michini, Jonathan P How, M Vavrina, and John Vian. Automated battery swap and recharge to enable persistent uav missions. In *AIAA Infotech@ Aerospace Conference*, 2011.

[35] L. Xie, Y. Shi, and Y. Hou. Making sensor networks immortal: An energy-renewal approach with wireless power transfer. *IEEE/ACM Transactions on Networking*, 20(6):1748–1761, December 2012.

[36] Liguang Xie, Yi Shi, Y Thomas Hou, and A Lou. Wireless power transfer and applications to sensor networks. *Wireless Communications, IEEE*, 20(4):140–145, 2013.

[37] Peng Yang, Zi Li, Wensheng Zhang, and D. Qiao. Prolonging sensor network lifetime through wireless charging. In *IEEE Real-Time Systems Symposium, RTSS 2010*, San Diego, CA, December 2010.