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 together with robot control of unmanned vehicles (UVs) to simultaneously select sensors to recharge, cut transmissions from long to short-distances, collect sensed information, and replenish WSN's energy. Different from prior work focusing solely on energy replenishment, this research replenishes sensors' energy, significantly reduce WSN's energy consumption, and efficiently deliver sensor data to the sink. This paper focuses on fundamental challenges associated with sustainable WSN development by jointly using wireless power transfer technology and controlling UVs. Energy costs due to power transmission are less than the UV's transportation costs. This is true for WSNs spread over large geographic areas, terrain with obstacles, or where transportation costs are high, such as subsea or aerial UVs. This task focuses on algorithms that make such WSNs sustainable by focusing on path-planning and trajectory optimization.**

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

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], [20], [23], [35]. Although advances in processing and computing designs can endow sensors with a multitude of sensing modalities (temperature, pressure, light, magnetometer, infrared, etc.), the crawling development of battery technology imposes harsh energy constraints on battery-powered sensors and the sustainable working of WSNs. In WSNs, the majority of energy is consumed by (i) wireless transmission of perceived data [21], [34], [35], and (ii) long-distance multi-hop transmissions from source sensors to the sink. Research efforts to address WSN energy concerns have focused on energy conservation [8], environmental energy harvesting [15], [33] and incremental sensor deployment [43]. 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 [13]. 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*" [12].

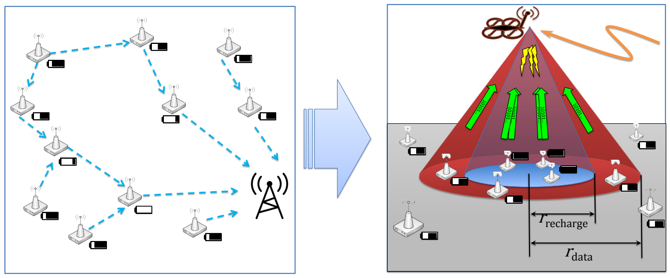
Fortunately, recent breakthroughs in the area of wireless power transfer technologies (e.g. inductive coupling, magnetic resonant, and RF energy harvesting) [28] provide promising alternatives for deploying such WSNs. *Magnetic resonant wireless power transfer* [28] has the ability to wirelessly transfer electric power from the energy storage device to the receiving device efficiently within medium range (e.g., 40% 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 [41].

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.

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 [14]. Servicing a WSN is closely related to coverage problems, recent work



Traditional WSN — Dynamic path-planning

Fig. 1: Evolution from traditional wireless sensor networks (WSNs) to servicing WSN with UV(s). We present techniques that uses unmanned vehicles (UVs) to gather aggregated data and recharge sensors using one or more vehicles, and design energy-optimal control policies for the UVs.

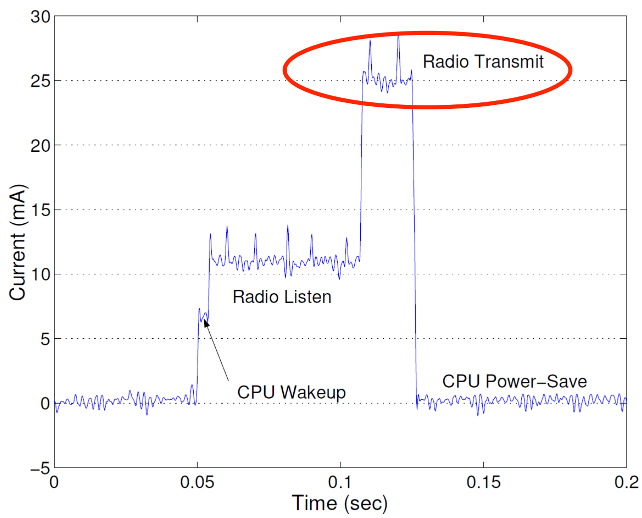


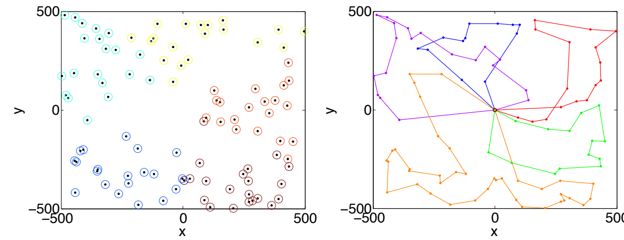
Fig. 2: Power usage in a wireless sensor node is dominated by transmission costs and listening costs. Figure modified from [26].

includes methods for optimizing speed along given routes [36], and techniques to continually improve existing routes [37]. 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 [22], to minimizing overall delay while sharing bandwidth [19], to having a set schedule and opportunistically deviating from it [18]

Using unmanned aerial vehicles to recharge other robots or sensor nodes has focused on physical design, which includes direct contact, such as swapping batteries [39], [40] or direct recharge [32], wireless resonant coupling [17], [24], [25], and electromagnetic radiation [42], and algorithmic improvements using graph theory [31], linear programming [36], and gradient descent optimization [37].

III. Overview

The previous work use 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



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

Fig. 3: Using the motion tracking system, we can abstract the problem of localizing our sensor nodes and focus on implementing path-planning algorithms. (Right) screenshots from a mTSP solver aided by our heuristic.

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 could be written as $K_{dist} = \frac{E_{\text{movement}} * \text{path\_length}}{E_{\text{recharge}}|\text{nodes}|}$. 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].

This paper's goal is to explore *tipping points* in recharging WSNs. We will focus on three classes of tipping points: travel vs. recharging (III-A), local density of WSN nodes (III-A), static vs. dynamic loading(III-C).

The following subtasks delineate when each indicates a required change in algorithm, investigate a specific swarm-robotics technique for each, and implement solutions. Together these techniques will increase sustainability of WSNs. Solutions for these subtasks will be verified using the quadcopter testbed described in III-C1.

A. Travel vs. Recharging: mTSP for path-planning

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 labelling our sensor nodes as cities, the solution to the traveling salesman problem gives the shortest length path. This problem is NP-hard, 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]).

UVs have a limited energy budget. As the number of nodes grows, more UVs are needed. One solution is to require all UVs to return to the sink to recharge and return data. This formulation is called the *multiple-Traveling Salesman Problem* (mTSP) [7], but does not try to balance the workload between UVs. A good heuristic can increase TSP solver performance. In our numerical simulations, priming an open-source genetic algorithm solver [27] by sorting the nodes by angle from the sink and dividing the sorted list equally between the UVs decreased path costs by 20%. Figure 8 shows results from our simulation with 100 nodes and 5 UVs.

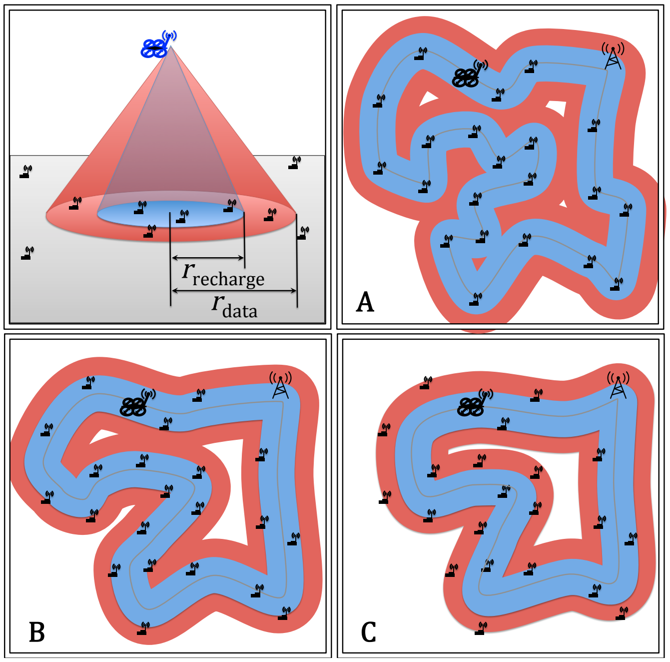


Fig. 4: A UV has an associated recharging footprint and a data-transfer footprint, which can often be modeled as disks of radius $r_{recharge}$ and r_{data}. Path **A** visits each node, but path **B** is shorter because it is designed merely to recharge all nodes. Path **C** is the least tortuous because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

B. Using local density to optimize path speed

Data aggregation and recharging shares similarities with persistent robotic tasks, such as cleaning, mowing, observing, and patrolling [5], [9]–[11], [26], [30], [36], [37]. A key tipping point in these problems is if a UV can service multiple clients simultaneously. 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. We represent the fraction of sensors that are clustered as $K_{footprint} = \sum_{j=i+1}^{N} \sum_{i=1}^{N} ||p_i - p_j||_2 \leq r_{footprint}$. Here, p_i is the position of the ith node, $i \neq j$.

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 to take paths *between* one or more sensors. In Fig. 4, path **B** is designed to transfer data from all nodes, and the optimal solution is often to weave between clusters of nodes. The third regime is when many nodes are close enough for recharging.

C. Static vs. dynamic loading: gradient descent on path-planning

The above algorithms assumed a static WSN, but often sensor data transmission is dependent on transient phenomena. For example, a swarm of subsea sensors may track a school

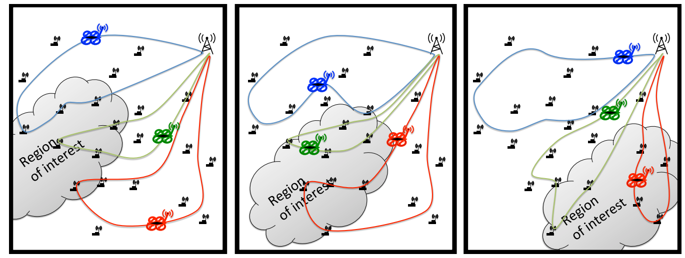


Fig. 5: The grey cloud represents a time-varying region of interest. Allowing the UVs to dynamically modify their routes in a distributed manner enables a robust response to changing conditions while maintaining service.

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.

For this subtask we design local optimization techniques to iteratively adapt the paths of UVs. A schematic of our the adaptive control law is shown in Fig. 5. Recent research has focused on local optimization techniques that gradually improve the paths followed by robots during persistent tasks [37]. These techniques are amenable to WSN. The base technique is a variant of Lloyd's algorithm [6], [29]. Each path is represented by a finite number of waypoints, and these waypoints are both attracted to the centroid of all sensor nodes within their Voronoi cell, and attracted to their neighboring waypoints. Our MATLAB implementation is available at mathworks.com/matlabcentral/fileexchange/49863 [38].

1) Hardware implementations

Hardware implementations generate confidence in solutions, often uncover assumptions in our theory, and are excellent for teaching. Our initial implementations will use an existing fleet of eight industrial mobile robot bases (ERA), shown in Fig. **??**. These robots are ideal for demonstrating WSN servicing because they are simple, have a long battery life, and are highly expandable. Each robot fits within a 40cm cube, and can be accurately tracked with our OptiTrack motion capture system, allowing us to quickly focus on the algorithms.

IV. Algorithm

Let us consider $\mathbf{N}(r = 1...\mathbf{N})$ UVs servicing the Wireless Sensor Network in a convex, bounded area $\mathbf{Q} \subset \mathbf{R^2}$ and let $\mathbf{p}_i^r$ be the position of the $\mathbf{i}^{th}(\mathbf{i} \in (1...n(r)))$ waypoint of the $\mathbf{r}^{th}$ UV. Servicing includes recharging the nodes and collecting a part of the data that the nodes are about to transmit to the sink there by reducing the power expenditure in the sensor nodes. The algorithm help in formulating an optimal path to visit the sensor nodes in the WSN depending on the interesting regions. Waypoints are a set of points that define the path for each UV $(\mathbf{1} : \mathbf{r})$, the UV travels in a straight line in between two neighboring waypoints. At each step, we compute the Voronoi partition$(\mathbf{V_i^r})$ defined by the waypoints, with one partition assigned to each waypoint. We define an interesting function $\phi(\mathbf{q})$ that commands the usefulness of a location on the map for servicing the WSN $(\mathbf{q} \in Q|\phi(\mathbf{q}) > 0)$.

$$\mathbf{H} = \sum_{r=1}^{\mathbf{N}} \sum_{i=1}^{\mathbf{n(r)}} \int_{V_i^r} \frac{\mathbf{W_s}}{2} \|q - p_i^r\|^2 \phi(\mathbf{q}) d\mathbf{q} \quad + \sum_{r=1}^{\mathbf{N}} \sum_{i=1}^{\mathbf{n(r)}} \frac{\mathbf{W_n}}{2} \|p_i^r - p_{i+1}^r\|^2 \tag{1}$$

The above mentioned equation is the cost-function of the algorithm we aim at minimizing the value. The first part of the equation indicates sensing in regions far away from the interesting regions is costly and the second part indicates having neighboring points far away is also costly. This help is producing a concise path that mostly travels only on the interesting region there by minimizing the cost of travel. $\mathbf{W_s}, \mathbf{W_n}$ are positive scalar constants that are used to weight the sensing and neighbor distance respectively depending on the experimental setup.

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

$$M_i^r = \int_{V_i^r} \phi(\mathbf{q}) d\mathbf{q}, \mathbf{L}_i^r = \int_{V_i^r} \mathbf{q}\phi(\mathbf{q}) d\mathbf{q}, \quad \mathbf{C}_i^r = \frac{\mathbf{L}_i^r}{M_i^r} \tag{2}$$

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

$$\mathbf{u}_i^r = \frac{K_i^r (M_i^r \mathbf{e}_i^r + \boldsymbol{\alpha}_i^r)}{\beta_i^r} \tag{3}$$

Here, $\mathbf{K_i^r}$ is a positive definite matrix (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. This tries to move the waypoint towards the interesting region there by 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 binds the neighboring waypoints together in order to obtain a closed path. $\boldsymbol{\beta}_i^r = \mathbf{M}_i^r + 2\mathbf{W}_n$ normalizes the weight distribution between servicing interesting regions and staying close to neighbors.

The control is then applied to each waypoint and it's position is updated:

$$\mathbf{p}_i^r(k) = \mathbf{p}_i^r(k-1) + \mathbf{u}_i^r \tag{4}$$

It is important to have an initial path that fills the map so that the robots can identify all the sensory regions, which helps in forming an optimal path. A space filling algorithm needs to be implemented to fill the map. 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 [16].

The above algorithm results in a sub-optimal path that cannot be improved any further since we have hit a local minimum. To improvise the path i.e. to minimize the cost function we input the location of the waypoints obtained after

---

Algorithm 1 IC (Interesting Closed) path controller for the i^{th} waypoint $\mathbf{p}_i^r$ in robot r's path in a known environment (from [36], implemented at [38]).

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 $\mathbf{C}_i$ according to (2)
  Obtain neighbor waypoint locations $\mathbf{p}_{i-1}^r$ and $\mathbf{p}_{i+1}^r$
  Compute $\mathbf{u}_i^r$ according to (3)
  Update $\mathbf{p}_i^r$ according to (4)

---

running the above algorithm for $\mathbf{n}$ iterations into a **mTSP** (multiple-Traveling Salesman Problem) solver. The mTSP is an NP(Non-Deterministic Polynomial)-Hard problem it cannot be solved in polynomial time by a computer. A salesman wants to travel to every city in the world and come back to the city he started, the problem is we have to find the shortest route for his travel. In [] based on heuristics you get an approximate solution, it depends on the execution time of the algorithm. The waypoints position after $\mathbf{n}$ iteration is given to the mTSP solver which considers these to be city locations and formulates a path depending on the execution time. This straightens the loops in the path, reduces the cost function and snaps out of the local minimum obtained after the executing the first algorithm. These algorithms are repeated in succession to obtain a minimal value of cost function depending on the computational time available.

V. Results

A. One UV

A single UV system was simulated using MATLAB in Fig. 6. [38].The initial path at iteration 0 was set to be a space-filling Hilbert's curve, in order to identify the location of the sensor nodes. The UV follows this initial path there by getting access to the whole map and identifying the interesting points. A waypoint at $[0, 0]$ is stationary, it acts as a sink for the UV to recharge and to unload the data collected while servicing the sensor nodes. For the first 100 iterations the Algorithm with Gradient Descent is used for simulation. This uses the Voronoi diagram for moving the waypoints to locations where there is high sensory information and uses the Gradient Descent for optimization. The path achieved after 100 iterations is a local-optimum and iterating further does not decrease the cost function. This sub-optimal solution obtained is not impressive. Thus to optimize our path further and to snap out of 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 deceased 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 converges asymptotically. The cost function has also been plotted which is strictly decreasing reassures the significance of the Algorithms used.

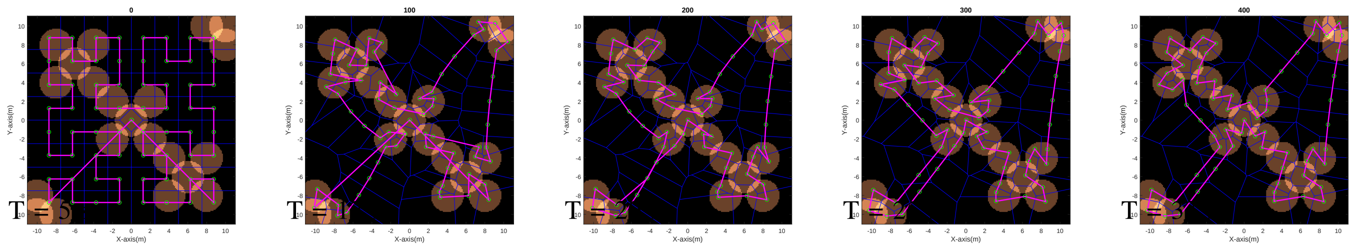


Fig. 6: Simulation of the Algorithm that uses Gradient Descent 1 and the mTSP was plotted with one UV. 1.)Iteration 0-Hilbert's Space Filling Curve.2.)Iteration 100- Algorithm with Gradient Descent(first).3.)Iteration 200- mTSP Solver(first).4.)Iteration 300-Algorithm with Gradient Descent(second).5.)Iteration 400- mTSP Solver(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 overlaying density plot represents the sensor node's interesting region.

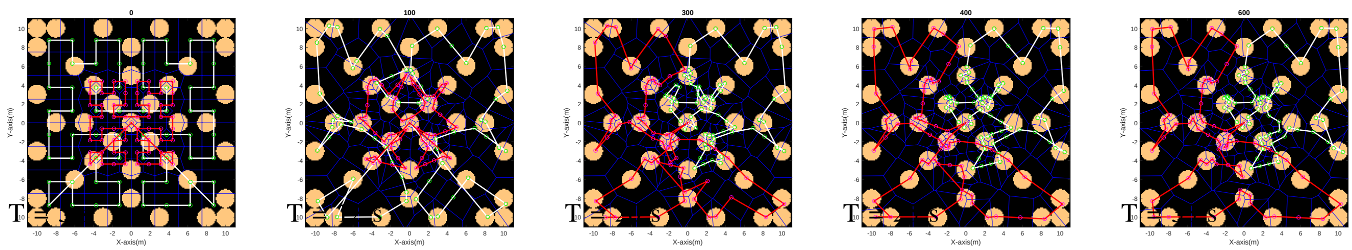
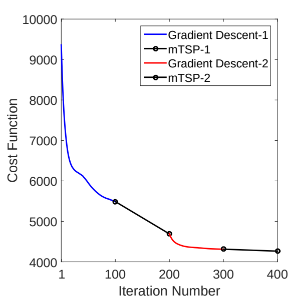
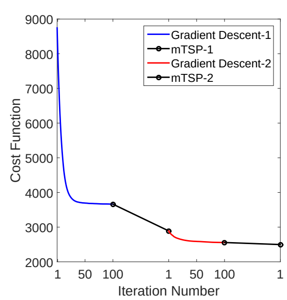


Fig. 7: Simulation of Algorithm with Gradient Descent 1 and the mTSP was plotted with two UVs. 1.)Iteration 0-Hilbert's Space Filling Curve.2.)Iteration 100- Algorithm with Gradient Descent(first).3.)Iteration 200- mTSP Solver(first).4.)Iteration 300-Algorithm with Gradient Descent(second).5.)Iteration 400- mTSP Solver(second).6.)Cost-function is plotted. The waypoints are indicated by a set of linked red and yellow ○ markers, the associated Voronoi diagram is in blue, the white and cyan lines represent the path the UVs follows for servicing the sensor nodes and the overlaying density plot represents the sensor node's interesting region.



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



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

Fig. 8: Cost function indicates a decreasing trend approving the optimization algorithm.

B. Multiple UVs

A two UV system was simulated on MATLAB. As shown in 7, UVs service different sets of nodes in the WSN. A multi-UV system is apt in a practical sense since 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 for the same reasons stated. A waypoint at $[0, 0]$ is stationary, it acts as a sink for both the UVs to recharge and to unload the 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

An optimized path-planning algorithm was simulated to service a WSN. The path constructed is adaptive to the sensor node locations. Future work should extend our simulation to handle non-stationary sensor nodes, improve convergence rate, and use our mTSP code to escape local minimal. We are in the process of implementing the algorithm on mobile-robots, with eventual implementation with a set of quadcopters, as shown in Fig. **??**.

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] Howie Choset. Coverage for robotics - a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):113–126, October 2001.

[10] Nikolaus Correll, Nikos Arechiga, Adrienne Bolger, Mario Bollini, Benjamin Charrow, Adam Clayton, Felipe Dominguez, Kenneth Donahue, Samuel Dyar, Luke Johnson, et al. Building a distributed robot garden. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1509–1516. IEEE, 2009.

[11] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *Robotics and Automation, IEEE Transactions on*, 20(2):243–255, April 2004.

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

[13] 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.

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

[15] 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.

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

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

[25] Sunghun Jung, Taesup Lee, Tamzidul Mina, and Kartik B Ariyur. Inductive or magnetic recharging for small uavs. Technical report, SAE Technical Paper, 2012.

[26] Nikolaos MP Kakalis and Yiannis Ventikos. Robotic swarm concept for efficient oil spill confrontation. *Journal of hazardous materials*, 154(1):880–887, 2008.

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

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

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

[30] Douglas MacKenzie, Tucker Balch, et al. Making a clean sweep: Behavior based vacuuming. In *AAAI Fall Symposium, Instantiating Real-World Agents*, mar 1996.

[31] 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.

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

[33] 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.

[34] Y. Shi and Y. T. Hou. Optimal base station placement in wireless sensor networks. *ACM Transactions on Sensor Networks*, 5(4), 2009.

[35] 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.

[36] 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.

[37] 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.

[38] Srikanth Kandanuru Venkata Sudarshan and Aaron T. Becker. "Decentralized Path Planning For Coverage using Gradient Descent Adaptive Control." MATLAB Central File Exchange, February 2014.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.