Preprint of an Artikle that appeard in
"Delay Tolerant Networks: Protocols and Applications"
Editor(s): A. Vasilakos, Y. Zhang, T. Spyropoulos
ISBN: 9781439811085

Auerbach Publications, CRC Press, Taylor & Francis Group, 2011

Chapter 1

Delay-tolerant Monitoring of Mobility-assisted WSN

Abdelmajid Khelil, Faisal Karim Shaikh, Azad Ali and Neeraj Suri [1]

{khelil, fkarim, azad, suri}@informatik.tu-darmstadt.de

Dependable Embedded Systems and Software Group

Christian Reinl

reinl@sim.tu-darmstadt.de

Simulation, Systems Optimization, and Robotics Group

Darmstadt University of Technology

Hochschulstr. 10, D-64289 Darmstadt, Germany.

"You only see what you know", J.W. von Goethe

Wireless Sensor Networks (WSN) usually are composed of fragile sensor

[1]Research supported in part by DFG GRK 1362 (TUD GKMM), EC INSPIRE, EC CoMiFin, MUET and HEC

nodes equipped with short range radios. Despite its elements fragility, the system is expected to deliver dependable services. This is mainly ensured through the inherent node redundancy. However, being battery-powered the question is not when the desired dependability can not be maintained anymore but when. Therefore, frugal monitoring techniques are vital for a WSN. Global maps of the sensor field, such as residual energy maps, are of high acceptance for both system users and designers. But, the map construction can become very inefficient if it requires an extensive intervention of the resource-limited sensor nodes. In this work, we present gMAP, an extremely efficient mobility-assisted approach to construct global maps. In gMAP (a) sensor nodes do not need to process readings of other nodes and (b) require to communicate a minimal number of messages compared to the existing map-based approaches. This is achieved by opportunistically exploiting node mobility to collect data of interest, keeping sensor nodes transmit only their own readings on-demand to a mobile node in their transmission area. Our approach is designed for generalized scenarios from unstructured, semi-structured to unstructured. If node mobility is controllable, gMAP includes an integrated scalable path planning algorithm for mobile nodes.

1.1 Introduction

Wireless Sensor Networks (WSN) are seeing increasing usage in several applications such as military, rescue and surveillance scenarios. Typical for such scenarios is that mobile nodes cooperate side-by-side with stationary sensor nodes to monitor the area of interest and to support the core network operations such as data collection [58, 50]. One dilemma of WSNs is that while they are often composed of cheap and basic elements (hardware and software) they are still expected to provide reliable services such as detecting fire. Another dilemma of WSNs is that they are supposed to be long-life systems despite of the fact that they rely on a finite energy source and operate under stress in harsh environments. Consequently, monitoring the operational health (especially energy levels) of the WSN elements aids the overall WSN in providing effective services. Accordingly, utilities for network monitoring and diagnosis are required. Most of existing monitoring techniques operate on sensor node level and ignore the inherent redundant node deployment in WSNs, which wastes valuable battery and network resources. An emerging diagnostic schema entails creating WSN wide maps of interested attributes [17] called as global maps (gMap), which operate on region-level and less on node-level.

gMaps present the spatial distribution of such relevant attributes. For example the energy map (eMap) depicts the spatial distribution of residual energy of the WSN elements. Maps provide elementary utilities for the assessment of the WSN dependability. They support network designers and administrators to monitor and optimize the network in operation. An example of supporting the network management is to utilize the eMap to detect and predict important vulnerabilities such as network partitioning.

A variety of approaches for map data collection have been developed [18, 19, 20, 21, 22, 23, 24]. However, all these approaches rely on multi-hop communication and/or in-network aggregation, which overstrains the sensor nodes through the use of the limited energy and processing resources of the stationary sensor nodes. In [25], the authors demonstrated that node mobility can increase the capacity of ad hoc networks, if the mobile nodes transport the message closer to the destination instead of immediately using multi-hop communication. This comes at the cost of higher end-to-end delays for communication. Fortunately, several WSN applications and network management tasks can tolerate delays in the range of minutes, hours or even days [26, 58, 50, 53]. For instance, the energy spatial distribution usually evolves only on a relatively large time scale, therefore, the data collection for constructing and updating the eMap can last longer to increase efficiency.

In [27], we have presented gMAP, an extremely efficient mobility-assisted approach to construct global maps. In gMAP (a) sensor nodes do not need to process readings of other nodes and (b) require to communicate a minimal number of messages compared to all existing approaches. This is achieved by opportunistically exploiting node mobility to collect data of interest, keeping sensor nodes transmit only their own readings on-demand to a mobile node in their transmission area.

While [27] focused on the construction of eMaps for two extreme deployment scenarios, in this chapter we synthesize our previous results [17, 27] and extend the gMAP approach to support scenarios of generalized mobility and deployment. In particular, we present a novel path planning algorithm for mobile nodes to efficiently collect map data and its updates from the WSN. In order to determine an optimal path w.r.t. completeness of collected data and energy overhead, we solve an adapted traveling salesman problem with "flexible cities".

The remainder of this chapter is organized as follows. After the review of the state of the art in Section 1.2, Section 1.3 clarifies our system model. Section 1.4 presents our novel gMAP approach using the example of eMaps. In Section 1.5, we evaluate our approach through simulations. We conclude the paper in Section 1.6 and discuss open issues in Section 1.7.

1.2 State of the Art

Given the particular properties of WSNs such as energy-constraints and node fragility it is difficult to simply customize the existing monitoring techniques for WSNs. Subsequently, many monitoring techniques tailored for WSNs have been developed. Concerning latency of data collection, we identify two main classes of network monitoring techniques in WSN: Delay-critical and delay-tolerant. Regarding the monitoring level, we distinguish between approaches that focus on node properties and diagnose the network on node-level and those that group nodes into regions (Fig. 1.1).

A few of the time-critical techniques exploit the spatial correlation of node properties. These are known as region-centric or (map-based) monitoring techniques. They do not consider node mobility and rely on pure multi-hop communication. Delay-tolerant techniques exploit mobility to collect the data. However, they do not exploit the spatial correlation of monitoring attributes to optimize collection and presentation of the monitoring data. Our gMAP approach profits from the spatial correlation of monitoring data as well as the high data's lifetime (Fig. 1.1).

1.2.1 Node-centric Delay-critical Monitoring Techniques

Sympathy [40] is a tool for detecting and debugging node-level failures in WSN. The authors show that it is possible to detect and diagnose failures by collecting and analyzing a minimal set of metrics regarding connectivity, traffic flow and node properties at a centralized sink. Sympathy includes an algorithm that root-causes failures and localizes their sources in order to point the user to a small number of probable causes: Node, path or the sink. *Memento* [41] provides failure detection and symptom alerts. Memento provides an energy-efficient protocol to deliver state summaries and a distributed failure detector module. The authors show that distributed monitoring of a subset of well-connected neighbors using a variance-bound based failure detector is suitable for use in practice. The *Sensor Network Tomography (SNT)* [42, 43] provides techniques to compute global predicates (e.g., the total number of nodes) using in-network processing mainly the in-network aggregation.

Sympathy, Memento and SNT waste bandwidth and energy resources. In addition, the monitoring performance suffers from network perturbations. Therefore, new (non-intrusive) approaches [44, 45] have been presented to passively monitor existing traffic and deduce network health indications. Unfortunately, these approaches suffer from volatile and high inaccuracies.

1.2.2 Region-centric Delay-critical Monitoring Techniques

The region-centric (or map-based) techniques collect data from the entire network and construct a map of this data on the fly. In cartography [31], isolines (also isopleth or contour) and choropleths are the common types of maps. For the varied map types, different data collection techniques have been developed. The naive approach to collect raw data for map construction would be if each node reports its value to the sink using multi-hop communication. This is obviously inefficient. Consequently, more efficient approaches have been developed based on techniques such as in-network aggregation [18, 19, 20]. Other approaches use suppression mechanisms to reduce the number of nodes reporting their raw readings to the sink [22, 23, 24].

Aggregation-based Approaches: *eScan [18] and isobar [19]* are approaches based on polygon aggregation. First, a request for energy values is flood to all network nodes. This constructs an aggregation tree that can be used to aggregate the energy values while being reported by each node. The aggregation consists in grouping sensor readings that meet a certain criteria (being geographically adjacent and in the same value range). The outcome of the aggregation is a list of (spatial) regions. A region is a polygon that is defined by the line spanning its border nodes. At the sink the aggregation results in an energy map delivered to the user. Each sensor node propagates its position along with the energy values for aggregation purposes. Furthermore, the sensor nodes (even those that have critical residual energy level and especially those closer to the sink) are main actors in map construction, leading to higher processing and communication activities and subsequently to a serious degradation of the network lifetime and disturbance of the core functionality. *INLR [20]* is an aggregation-based approach that focuses on small scale WSNs. A sensor node sends its reading or the calculated aggregate not only to its parent in the aggregation tree but to all its neighbors that are 1-hop closer to the sink. Therefore, nodes possess a partial map and the sink the global map (choropleth). While using more than one parent increases the accuracy of the map, the efficiency is sacrificed.

Suppression-based Approaches: *Isoline [22]* is an approach based on localized isocluster aggregation. The map building is reduced to the detection of isolines. Neighboring nodes share their readings. A node compares its reading with the readings of all neighbor nodes and detects an isoline, when the readings lie in different sides of a globally defined isoline. The detection of an isoline needs to be reported to the sink by the closest neighbor to the sink. The isocluster aggregation outperforms polygon aggregation in terms of accuracy with minor energy savings. *Meng et al. [23]* motivate the use of contour (isoline) maps for efficient continuous monitoring in sensor networks. The main contribution of this paper is the design of a temporal and spatial local suppression mechanism that prohibits some nodes to report their readings. The number of saved reports highly depends on the spatial correlation between sensor readings. Sensor nodes report their readings using multi-hop routing without any in-network processing. The map is constructed on the sink using interpolation and smoothing techniques. *Iso-Map [24]* also does not rely on in-network processing. It uses a suppression mechanism to reduce the number of nodes that report their readings to the sink using multi-hop communication. This approach is very similar to that of Isoline [22]. However, nodes need to report the gradient direction of the isolines, which requires excessive processing on sensor nodes.

1.2.3 Mobility-assisted Delay-tolerant Monitoring Techniques

Mobility-assisted data collection achieve high bandwidth and energy efficiency at the cost of a high end-to-end delay. In the following we survey the techniques to collect data from the WSN in a mobility-aided fashion. Next, we review the existing approaches to control mobility.

Mobility-aided Data Collection: There exist different mobility-assisted data collection techniques in the literature. These techniques have been developed to collect user data, however, they are easy to adopt to collect monitoring data. In the following we review these techniques. For further readings we refer to the excellent survey [26].

Data Mule [58, 57, 56, 55, 54] is a mobile node that collects data from sensor nodes as it passes by. In [58], a basic theoretical analysis has been developed. Hereby, simplistic deployments and communication protocols have been considered. Sensor nodes are deployed randomly on a grid and multiple data mules move according to the random walk model. The sensor nodes buffer their data until the mule can receive it through direct communication. The mule buffers the data until it can deliver it to the sink. In [57, 56, 55, 54] the movement paths are fixed and the coverage is not guaranteed. Therefore, some sensor nodes may need to have a multi-hop path to reach the mule. Nodes not on the movement path establish a local routing tree to send data to sensor nodes on the path, which collect the data till the data mule passes by.

Message Ferrying [50] is similar to data mules, but designed for mobile ad hoc networks (MANET) and mainly to overcome partitioning. Mobile nodes act like ferries to reach disconnected sensor nodes. [50] uses controlled movement to deliver data. Ferries can either follow a predefined movement path or change their movement path on-demand. To "order" the ferry, nodes can spontaneously increase their communication range to reach a message ferry near by and inform it about communication need so that the ferry can change its path. In order to allow for differentiated end-to-end latencies in [51] the authors suggested to prioritize messages. They present a forwarding strategy for the ferry route and discuss the buffer requirements to deal with this proposal. In [52, 53], the authors improve message ferrying by introducing a power management framework. If nodes know when they are going to encounter the ferry they can sleep to conserve power.

Also [61] is designed for MANET. This work analyzes two simple data delivery schemes, namely, the direct transmission and flooding, based on (1) the likelihood that a sensor node can deliver data messages to the sink, (2) and the message fault tolerance, i.e., the probability that at least one copy of the message is delivered to the sink by other sensor nodes.

Path Planning: Path planning for coverage problems are studied since many years and many centralized and distributed approaches, for various indoor and outdoor scenarios (e.g. [5, 4]) have been proposed. Many of these approaches are basically built on studying traveling salesman problems (TSP[15]) and vehicle routing and scheduling problems [6]. For the case of unknown or only roughly estimated node positions, that have to be covered, a distributed planning on the mobile node based on its local information is more adequate [2].

A suitable approach for the problem of planing an optimal path based on the nodes' topology is using hybrid optimal control theory, what results in solving mixed-integer nonlinear programming (MINLP) in practice. By considering communication ranges and optimality w.r.t a specific physical motion dynamics of the mobile nodes [11, 15]. Therefore, an adjusted objective function describes desired features to be minimized like overlappings or path-length. This solution suits well for small scale networks but becomes very inefficient for common WSN settings. This is mainly caused by the underlying discrete structure which results in a highly combinatorial character of the entire problem.

Even the basic problem (known as traveling salesman problem with neighborhoods - TSPN [16]) of planning the shortest path is NP-hard; without considering a certain locomotion dynamic. Thus many approximating algorithms for the TSPN under mild assumptions were proposed in the last years [14, 10]. For small scale WSN scenarios (a few dozen of nodes) a solution based on TSP-path-planning has been recently presented in [59]. The TSPN-problem is only shortly discussed there without presenting a solution for larger scenarios with a dense setting of nodes. For larger WSN scenarios a problem similar to TSPN is solved in [3], but also disregarding overlaps within the transmission areas of subsequent breakpoints.

1.2.4 Our Contributions Compared to the Existing Approaches

Compared to node-centric monitoring techniques, gMAP collects data at comparable high accuracy (node level). However, gMAP outperforms these techniques w.r.t. load balancing on sensor nodes and the map-based presentation of monitoring data.

Compared to map-based techniques, our gMAP approach uses a *minimal* number of messages without sacrificing the completeness of sensor information. This provides for high efficiency with respect to both energy and bandwidth consumption. In gMAP we *decouple* the collection of the sensor values from the construction of the map, which results in minimal processing on sensor nodes reducing the energy consumption on them. Furthermore, gMAP charges all sensor nodes similarly and contributes to the desired energy balancing in WSNs. Our approach is *resilient to network partitioning*, which increases the dependability of the WSN, since monitoring tasks can continue reporting the health of the network even if critical failures/situations occur.

Data collection in gMAP is similar to that of data mule and message ferrying approaches. However, most of existing approaches have been developed for specific scenarios such as sparsely deployed and structured WSNs (data mule) or MANET (message ferrying). We focus on WSN and provide techniques that are for a generalized scenarios (from structured to unstructured).

1.3 System Model

In this work, we consider the established *mobile* Wireless Sensor Network (mWSN) model. This model is used in a variety of WSN deployments, in particular in emergency and military scenarios. The main functionality of the mWSN is implemented by a large number of stationary resource-limited *sensor nodes (SN)* that are deployed following either an arbitrary or structured spatial distribution in the area of interest. Also one dedicated stationary sink is selected as the interface to the user. Additionally, a few mobile *assist nodes (AN)* are deployed with generalized support roles such as (1) application support (e.g., additional interface to users), (2) functionality support (e.g., delay-tolerant data transport), and (3) network support (e.g., diagnosis). The mobile nodes cover a functional capability spanning robots, unmanned air vehicle (UAV), etc. Hereby, we assume that the AN is able to move to and stop at any position in the sensor field. In this paper, we consider a mWSN composed of $N_N = N - 1$ SNs, with one sink and one mobile AN.

We consider two major classes of mobility: Structured mobility, i.e., predictable & controllable, and unstructured mobility, i.e., unpredictable & uncontrollable. The AN possesses high processing, storage and energy capabilities compared to SNs. Furthermore, it has no energy limitations because it can recharge its batteries by means of on-board renewable energy resources [32] or through moving to recharging energy-stations. We assume that SNs use the batteries as a main energy source. These batteries continuously discharge following a long-running process in the range of months or even years. We consider that SNs as well as the AN knows its position. We assume that all deployed nodes are cooperating and that no misbehaving nodes may exist.

For simplicity, we consider all nodes (AN and SNs) are equipped with a conformal level of communication technology and are able to communicate if they are in each other's transmission range R. We use a CSMA/CA based MAC layer, where communication links are symmetric and bidirectional, and collisions may occur. Furthermore, we assume that network can get partitioned, i.e., some SNs may not be able to communicate with the sink. We allow for the use of duty cycles for SNs. However, we assume that the magnitude of the movement distance covered by the AN during the time period of a duty cycle is negligible and that the duty cycles scheduler assures that all SNs in the AN's transmission area eventually receive the messages sent by the AN.

1.4 gMAP: Mobility-assisted Monitoring using Global Maps

We now present our novel gMAP approach comprising new algorithms to collect samples in a mobility-assisted way and a new technique to construct maps. We use the eMaps as an example, however, our methodology is generic and can be easily adopted for other functionality maps. We refer to our gMAP approach for eMaps by eMAP.

1.4.1 Overview of Approach

The main reasoning behind the eMAP approach to construct eMaps is that battery depletion occurs over an extended period of time and it is sufficient to check the battery level at a daily or weekly basis. This shows that collection of energy-based health indications is a *delay-tolerant* process, which allows us to deploy established concepts from the delay-tolerant networking research. Accordingly, the main design principle for the eMAP approach is to exploit the mobility of nodes to transport messages and collect information in a delay-tolerant way, thus reducing the communication overhead.

We let the mobile AN scan the sensor field and collect the energy information from each node it encounters. We are using one single mobile AN for simplicity of communicating the idea whereas a real implementation can consider multiple nodes or some primary/secondary arrangements. The AN sends a short beacon, on which nodes reply with their energy value and optionally their position. We proceed progressively, by first considering a structured scenario, then a semi-structured one and finally an unstructured one. For scenarios with controllable node mobility we design an integrated path planning algorithm. For all scenario we design appropriate algorithms to collect energy information.

We also present an efficient technique for the mobile AN to locally construct an appropriate eMap from the collected energy samples. The technique is based on measuring inequalities between neighboring samples and to group similar values into a region. Therefore, we refer to our technique by regioning (Section 1.4.5).

1.4.2 Scenario Classification

In this work, we focus on three important types of scenarios that provide basic features to build realistic scenarios.

1. In a structured scenario we assume that the spatial deployment of SNs is known a-priori and that the mobility of the AN is controllable (Fig. 1.2(a)).

2. In an semi-structured scenario with an a-priori known (or reliably estimated) spatial deployment of SNs we assume that the mobility of the AN to be controllable (Fig. 1.2(b)).

3. in a scenario with an unknown topology (e.g., random spatial deployment) the mobility of the AN is assumed to be unpredictable and uncontrollable (Fig. 1.2(c)).

Our main driver for the scenario selection is the proof of concept in extreme scenarios. Furthermore, in a realistic scenario the spatial deployment of SNs can be structured or known only partially. The mobility of the AN can be either controllable or uncontrollable and may follow varied patterns.

1.4.3 Path Planning of ANs

Path planning is required for the structured and semi-structured scenarios.

Structured mWSN Scenarios

In such scenarios, the SNs are deployed according to a specific uniform scheme (e.g., on a grid). The AN knows the accurate positions of SNs and accordingly plans it's movement.

Planning an optimal tour to collect the data with a minimal number of messages means to determine a minimal number of breakpoints with a minimal overlap of their corresponding transmission areas. The knowledge of the uniformly structured spatial distribution of the SNs can be used to simplify the complex problem of the optimal AN path planing, e.g., by applying an adopted pattern. Especially, if there are repeatedly occurring parts, this structure may allow to combine solutions from a simpler path planing problem determined for this smaller parts.

Further optimizations of the tour are possible, e.g, concerning tour time, length and the energy overhead for an AN. As our main goal is the proof-of-concept, we do not further consider these optimization in this work. We also assume a simple structure for SN deployment, i.e., grid topology.

For the grid topology, the work [5] suggest a zigzag movement of nodes. Accordingly, the AN crosses the full length of the sensor field in a straight line, turns around, and then traces a new straight line path adjacent to the previous one and $2 \cdot R$ far from it. By repeating this the AN covers the entire WSN field.

Semi-structured mWSN Scenarios

In such scenarios, the AN knows the accurate positions of SNs and accordingly plans it's movement. However, nodes deployment is not structured, which complicates an intuitive path planning of the AN. In the following, we investigate the approaches to plan the movement of one controllable AN to efficiently collect network health information from SNs. Our approach consists in stepwise decomposition of the problem into different less complex subproblems: (1) Finding suitable breakpoints, (2) reducing overlaps in the communication range, and (3) planing a shortest-path-tour. Subsequently, we integrate the sub-solutions into a single path planning algorithm.

We are looking for a set of points where the AN stops and communicates with the SNs within the AN's communication range. The goal is to get a set of points, considerably fewer than the number of SNs and fewer than the number of points needed for a whole coverage of the area with communication ranges. We point out here that it is not necessarily desired to get the really smallest possible number of breakpoints, but to get a small set of breakpoints that minimizes the useless overlapping in their coverage. The reasoning behind this is that sensor nodes in coverage overlap will waste energy to listen to redundant beacons.

Fig. 1.3 depicts our proposal for path planning. First, we get candidates for a set of AN-positions by solving a nonlinear problem (NLP) and removing unnecessary points. Then, we repeat solving an approximative mixed-integer linear program (MILP) to finally get the optimal sequence TSP-solution.

STEP 1: Find a reduced number of breakpoint candidates: To avoid solving a large mixed-integer NLP we are proposing a new basic algorithm 1 for a given set of sensor positions $\mathtt{P}_S := \{(\xi^j, \eta^j) | j = 1, ..., N_N\}$. It works fine, even for the case of a verey dense spacial deployment of SNs and thus strongly overlapping transmission ranges of the SNs.

Minimizing the non-linear penalty function

$$\min_{\substack{(x^j,y^j)\\ j=1,\dots,N_B}} \varphi((x^j, y^j)) = \min_{\substack{(x^j,y^j)\\ j=1,\dots,N_B}} \sum_{i \in I_j^{BS}} \log(\|(x^i, y^i) - (x^j, y^j)\| + 1) \tag{1.1}$$

subject to the constraints $\forall k \in I_j^{BS} : \|(x^j, y^j) - (\xi^k, \eta^k)\| \leq R$ effects that closely adjacent breakpoints converge towards a common position and coincide ideally. In the context of solving a TSP for determining the shortest roundtrip for visiting all these points, we finally refer to them as “flexible cities”.

The efficiency of solving eq. (1.1) depends decisively on the number of adjacent SNs in N_N and number of breakpoints N_B. Thus it is desirable to start with a set $\mathtt{P}_B$ as small as possible. The algorithm can be adjusted by the adjacency parameter k_1.

Intermediately maximizing the penalty function effects changes in the set

Algorithm 1: Reducing breakpoint candidates

Initialize a set of breakpoints $\mathtt{P}_B := \{(x^j, y^j)|j = 1, ..., N_B\}$ that guarantees connectivity to all SN
while (number N_B of breakpoints decreases) **do**
  For each break point $(x^j, y^j) \in \mathtt{P}_B$ determine the indexes of SN lying inside the Voronoy-cell of (x^j, y^j) :
  $I_j^{BS} = \{i | j = \arg \min_{k=1,...,N_B} \{\|(x^k, y^k) - (\xi^i, \eta^i)\|\}\}$
  For each (x^j, y^j) determine the set of indexes of breakpoints in its adjacency: $I_j^{BB} = \{i | \|(x^j, y^j) - (x^i, y^i)\| \leq k_1 \cdot R\}\}$
  Minimize penalty function $\varphi((x^1, y^1), ..., (x^{N_B}, y^{N_B}))$ satisfying connectivity constraints
  Inspect for redundant transmission areas and delete breakpoints from $\mathtt{P}_B$.
  if ($\mathtt{P}_B$ could not be reduced) **then**
    Maximize (approximatively) penalty function $\varphi((x^1, y^1), ..., (x^{N_B}, y^{N_B}))$ satisfying connectivity constraints.
    Inspect for redundant transmission areas and delete breakpoints from $\mathtt{P}_B$.
  end if
end while
Return $\mathtt{P}_B^* := \mathtt{P}_B = \{(x^j, y^j)|j = 1, ..., N_B^*\}$

of adjacent SNs. This has shown up to improve local minima of N_B again. Eventually, this will result in a reduced set of N_B^* breakpoints (Fig. 1.3 (b)).

STEP 2 (optionally): Reduce overlapping between breakpoints: An overlap of breakpoints occurs if some nodes are covered by the transmission area of multiple breakpoints. The reduction of overlappings for a given number of N_B^* breakpoints can also be achieved by solving a MINLP:

$$\min \quad \sum_{(i,j) \in I_{BN}} b_{ij} \tag{1.2}$$

$$\text{subject to} \quad \forall (i,j) \in I_{BN} : \quad b_{ij} = 1 \;\Leftrightarrow\; \|(x^i, y^i) - (\xi^j, \eta^j)\| < R \tag{1.3}$$

$$\forall j \in 1, ..., N_N : \quad \sum_{i=1}^{N_B^*} b_{ij} \geq 1 \tag{1.4}$$

with $I_{BN} = \{(i,j) \in \{1,...,N_B^*\} \times \{1,...,N_N\} \,\big|\, \|(x^i,y^i)-(\xi^j,\eta^j)\| < 2\cdot R\}$ and $b_{ij} \in \{0,\,1\}$ Therefore, logical constraints like (1.3) have to be transformed to (in-)equalities by using e.g. Big-M- or Convex-Hull-formulations [12].

To increase efficiency and robustness of path planning for larger settings, we suggest a scaleable linearized standard approximation for (1.3): $(\forall (i,j) \in I_{BN})$

$$\forall k = 1,..\gamma : (x^i - \xi^j)\sin(k\frac{2\,\pi}{\gamma}) + (y^i - \eta^j)\cos(k\frac{2\,\pi}{\gamma}) \le R + (1-b_{ij})M_j \quad (1.5)$$

$$\forall k = 1,..\gamma : (x^i - \xi^j)\sin(k\frac{2\,\pi}{\gamma}) + (y^i - \eta^j)\cos(k\frac{2\,\pi}{\gamma}) > R + \hat{b}_{k,(i,j)}m_j \quad (1.6)$$

$$\sum_{k=1}^{\gamma} \hat{b}_{k,(i,j)} \le b_{i,j} + \gamma - 1\ , \quad (1.7)$$

where $M_j = max_{x^i,y^i}\{(x^i-\xi^j)\sin(k\frac{2\,\pi}{\gamma})+(y^i-\eta^j)\cos(k\frac{2\,\pi}{\gamma})-R\}$, $\hat{b}_{k,(i,j)} \in \{0,\,1\}$ and $m = max_{x^i,y^i}\{(x^i-\xi^j)\sin(k\frac{2\,\pi}{\gamma}) + (y^i-\eta^j)\cos(k\frac{2\,\pi}{\gamma}) - R\}$.

Compared to a MINLP problem, the MILP has an increased discrete structure. On the other hand, solving a MILP has some strong advantages: It depends less on the quality of solution estimations, results in a global minimum, is much more robust and more efficient so that it is also easier to handle time limits with the solver (i.e., the so far best feasible solution is returned to the user, when a given time limit is exceeded).

STEP 3: Finding the optimal path: Assuming that the AN has to stop to communicate, the costs to go from breakpoint i_1 to breakpoint i_2 are constant and independent from the sequence of breakpoints visited before i_1 and after i_2. Thus, one have to solve an Euclidean TSP. This can be efficiently achieved for hundreds of positions using an existing solver such as [9].

In [1] we are presenting an extension to a cooperative synchronized movement of multiple AN that additionally allows for inter-vehicle communication.

The Integrated Path Planning Algorithm:

As an example we now propose a scalable path planning algorithm based on solving the subproblems above. Depending on R, N_N and on the spatial distribution, the steps can be scaled by time constants $t_{max,1}$, $t_{max,2}$ and $t_{max,3}$ and by parameters in the implementation like the size of the adjacency in the set I_{NN} and I_{BN} or γ in (1.5) - (1.7).

Algorithm 2: Optimization-based path-planning for mWSN

/***** On sink in order to plan the mobile data collection ******/
/***STEP 1***/
[$\mathtt{P}_B$]=determine_breakpoints_NLP($R, \mathtt{P}_S, t_{max,1}$);
[$\mathtt{P}_B$]=remove_unnecessary_points($R, \mathtt{P}_B, \mathtt{P}_S$);
while $t \leq t_{max,2}$ AND $\mathtt{P}_B$ have been improved in last run **do**
  /***STEP 2***/
  $\mathtt{P}_B$=reduce_overlapping_MILP($R, \mathtt{P}_B, \mathtt{P}_S, t_{max,3}$);
  [$\mathtt{P}_B$]=remove_unnecessary_points($R, \mathtt{P}_B, \mathtt{P}_S$);
end while
/***STEP 3***/
solve_TSP($\mathtt{P}_B$);

Interrupting the MILP optimization after $t_{max,3}$ and then looking for unnecessary breakpoints on the so far best solution may again reduce the combinatorial character of the MILP before the minimization starts. Because of the MILPs structure here with many minima of equal value, this showed up to be more efficient than waiting until the solver gives back a proven global optimum.

We also apply Alg. 2 to the structured scenario and compare the performance of our path planning to the optimal zigzag tour.

1.4.4 Data Collection

For the structured and semi-structured scenarios, we present the following data collection algorithm. The AN performs a first *snapshot* by sending a REQ-beacon to all SNs in its transmission area using a MAC broadcast. A SN replies by sending a message containing its node-ID, location (loc) and energy level

E_{lev}. In order to reduce collisions, nodes schedule their reply for a random time t_{rand} between 0 and a maximum value T_{max}. The AN performs the subsequent snapshot after visiting the next breakpoint according to the path planning algorithm. The optimal result of the collection operation is a set of N_N elements with the following structure: {node-ID, loc, E_{lev}}.

In the following, we present our algorithm to collect energy information (Alg. 3) for unstructured scenario, where the movement of the AN is neither controllable nor predictable. If the AN performs a snapshot, moves $2 \cdot R$ away without changing the direction, and performs a second snapshot, then both snapshots are covering disjoint areas. Subsequently, we let the AN perform a second snapshot, only after moving $2 \cdot R$ from the location of the previous snapshot. The data collection completes, when the total WSN area is covered by all snapshots. We note that if the AN changes its movement direction, then the snapshots overlap and some nodes may receive redundant REQ beacons. The major concern for SNs is to minimize the number of messages to be sent or received. The AN is powerful enough to send REQ beacons frequently. However the REQ beacons are received by energy precious SNs. Therefore, we have to minimize the number of unnecessary REQ messages sent by AN. To avoid unnecessary snapshots, the AN maintains a history of snapshots $\{snapshot_{id}$, $snapshot_{loc}\}$. After moving $2 \cdot R$ from the location of the previous snapshot and before performing a second snapshot, the AN uses the history to calculate if the second snapshot has an additional coverage higher than a fixed threshold coverage $COV_{th}\%$. Only in this case the AN performs a snapshot. The value of $COV_{th}\%$ allows to investigate the trade-off between the number of redundant REQ beacons and the sampling latency. Once the AN scans the whole sensor field, the history of snapshots will be flushed and a new round will be initiated by the AN. To avoid unnecessary transmissions, SNs send information only once in a round as presented in Alg. 3.

Algorithm 3: Data Collection Algorithm for Unstructured mWSN

/***** On Assist Node (AN) ******/
var $HIST_{snapshot}$
Initiate a new round $round_{id}$ for sampling
AN: Do first snapshot: SEND REQ beacon with $round_{id}$
AN: STORE $\{snapshot_{id}, AN_{loc}\}$ in $HIST_{snapshot}$
$snapshot_{id} + +$
If AN has moved a distance of 2*R since previous snapshot do:
AN: CHECK $HIST_{snapshot}$
AN: Compute $COV_{additional}$ from current AN_{loc} and $HIST_{snapshot}$
if $COV_{additional} \geq COV_{th}$ **then**
    AN: SEND REQ beacon with $round_{id}$
    AN: STORE $\{snapshot_{id}, AN_{loc}\}$ in $HIST_{snapshot}$
    $snapshot_{id} + +$
else
    AN: Suppress REQ beacon
end if
AN: RECEIVE E_{msg}
AN: GOTO 7
/***** On Sensor Node (SN) ******/
SN: RECEIVE REQ beacon
if SN: new *round* **then**
    SN: Schedule transmission between $0 < t_{rand} < T_{max}$
    SN: SEND E_{msg} $\{ID, SN_{loc}, E_{lev}\}$
else
    SN: Suppress SEND E_{msg}
end if

1.4.5 eMap Construction

The prime goal of the map construction is to identify inequalities of energy density. Expected is an eMap that divides the sensor field into regions, which are indicators of similar energy-densities. The input of the construction algorithm is the collected residual-energy information and the output is the map's regions. The eMap is a geometrical/spatial data structure (e.g., tree) which is easy to evaluate. The construction operation has to satisfy some crucial requirements. First, it should be easy to evaluate on the AN. Second, two neighboring regions should have two "sufficiently" different energy densities. The map construction process is composed of the spatial partitioning of the sensor field (*space partition*) and the fusion of the regions of similar residual energy values (*regioning*).

For space partitioning, Voxel grid, triangulation (e.g., Voronoi or Delaunay), octree, k-d tree and BSP tree [35] can be used. All these schemes, except the Voxel grid are dependent from the input data. For this reason we select the simple Voxel grid for space partitioning. The primitive parameter to divide the sensor field is the size of smallest fragment of area, i.e., grid-cell size or the partitioning *resolution* (r). The energy density in the cell is the basis to form a region. Selecting r is a crucial decision for creating the eMap. Depending on this resolution a cell may contain more than one SN. We refer to the residual energy value of one cell by the sum of the energy values of all the nodes in that cell.

In order to merge the cells into regions (regioning), we need to ascertain if neighboring cells have similar values for residual energy. For this we need a technique to decide if two neighbor cells can be merged or not. A first possible technique is to use a metric to measure the inequality between two neighboring cells. In the literature we identify several inequality indices [36] that measure the inequality of a set: Variance, entropy coefficients, Hoover coefficient, Coulter coefficient, Gini coefficient etc. A second technique is to use global classes. In the eMAP approach we rely on the class-based technique for its simplicity and easy evaluation on ANs. Furthermore, we are investigating the suitability of other indices in ongoing work. The cells are classified into a fixed number of classes depending on their energy density. Neighboring cells are merged into the same region if they belong to the same class.

In Alg. 4 we propose the pseudocode of our regioning algorithm. This algorithm is based on searching and is inspired by the region growing algorithm for image segmentation [37]. We assign a region-ID to any cell to start regioning.

Then we check if neighbor cells can be merged with this cell. When we merge cells we assign the same region ID to them. Once the neighbor cells are checked for merge, we will repeat the process of cell merging for the neighbors that have been successfully merged to the current region. After completing regioning for the starting cell, all the other cells (which are not assigned to a region) will form regions in the same way. Hence we complete regioning for the whole WSN field.

Algorithm 4: Regioning

```
cell := structure{cellID, neighborList[], regionID=-1, energyClass}
grid := array of all cells
var currentRegionID
var currentRegion:= array of all cells with regionID=currentRegionID
/*********      regioning ()      *********/
regioning()
for each cell_i ∈ grid do
   if cell_i.cellID > −1 then
      next iteration
   end if
   cell_i.regionID=currentRegionID;
   regionMaking(cell_i)
   for each cell_j merged with the region of cell_i do
      regionMaking(cell_j)
   end for
   currentRegionID++
end for
/***** regioning with the 8 neighboring cells ******/
regionMaking(myCell)
for each neighborCell do
   if (myCell.energyClass = neighborCell.energyClass) then
      neighborCell.regionID = myCell.regionID
   end if
end for
```

We observe a trade-off between the *accuracy* and comprehensibility of the constructed map. The accuracy of the eMap depends on the accuracy of energy information collection and on the accuracy of regioning. Regioning accuracy is important to comprehend node distribution. If the model provides such a map that only the neighboring cells with the same energy-level form regions, it becomes the most accurate region map. It would be a worse map if regions consist of cells with highly different energy densities.

The selection of the number of energy classes is crucial since it allows for tuning the trade-off between map accuracy and comprehensibility. It should take into account the range of possible values and the level of inequality tolerated for regioning. Selecting a higher number of classes provides for higher map accuracy on the one hand but hardens regioning and subsequently the comprehensibility of the map on the other hand. A lower number of classes sacrifices accuracy in order to provide for a better comprehensibility. However, if the number of classes is too low, we merge cells with high difference in energy densities. Thus, regioning weakly reflects the energy spatial distribution. This results in an erroneous map. Summarizing, the number of classes should be appropriately selected to provide for the required trade-off between accuracy and comprehensibility.

1.5 Evaluation

In this section, we provide simulation-based analysis of the gMAP approach. In particular we evaluate the performance of the path planning framework, the data collection accuracy and efficiency, as well as the map construction accuracy and comprehensibility. In [27], we showed that gMAP requires to communicate a minimal number of messages compared to the existing map-based approaches.

1.5.1 Path Planning Performance

The performance of the proposed algorithm strongly depends on the spatial deployment of the SNs and particularly on the nodes local density. The more neighbors within the communication range of SNs, the more discrete structures in the neighborhood of each breakpoint occur, which arises as sets of constraints in the NLP or MILP to solve.

Obviously, it is easier to solve a coverage problem over the whole area for scenarios with a high density of SN. For scenarios with very low node density a TSP solution is recommended, followed by a minimization that shortens and smoothes the path again by moving the breakpoints ("flexible cities") within a certain neighborhood of the nodes.

For the proof of concept, the proposed algorithm in Section 1.4.3 has been designed and tested for the representative scenarios discussed in (Fig. 1.2). We solved STEP 1 using the IPOPT [8] NLP-solver on the NEOS server [7]. The MILP STEP 2 was solved by using CPLEX [13] running on a PC (Intel(R) Pentium(R) M processor 1.86GHz; 1024 MB RAM). The TSP in STEP 3 was solved with CONCORDE [9] on the same platform.

For a single AN we investigated scenarios with an increasing number of SNs but : (1) in a fixed area with an increasing density and (2) with a constant density. As time limits $t_{max,1} = \infty$, $t_{max,2} = 540$s, $t_{max,3} = 180$s and as MILP-approximation parameter $\gamma = 8$ were used. Resulting computing times for a setting with randomly distributed nodes is depicted in Fig. 1.4, a resulting path is shown in Fig. 1.2(b).

For the setting with 225 randomly distributed SNs, each node was covered by an average of 1.3 breakpoints, when the arbitrary time limit $t_{max,2} = 540$s was reached. For the harder problem with 225 nodes on a fixed grid, each node was covered by an average of 1.5 breakpoints after the same time limit $t_{max,2}$.

1.5.2 Data Collection and Map Construction Performance

We first describe our simulation settings. Then we define the evaluation metrics based on which we present our results.

Simulation Settings

We use Tossim [39] and its Tython extension for network simulations, and Matlab for map construction. Tossim is an event-driven simulation tool widely used in the WSN community. We have used the disc radio model provided by Tossim with 5 units communication range. All nodes lying in this communication range communicate without errors and have symmetric links. Although collisions may occur. We have considered 225 SNs either generated randomly in the area of 42 unit x 42 unit or in a grid topology of 15 x 15 (cell size $c = 3$ units) including sink. The AN moves either in controlled fashion (with pause time of $t_0 = 3sec$) or according to the random waypoint model using a constant speed of 1 $unit/sec$. We selected the commonly used random waypoint mobility model as its high randomness maximizes the unpredictability and assume that the mobility of the AN eventually covers the entire sensor field. SNs use $T_{max} = 500ms$ as a maximum time to schedule their replies to REQ beacons.

Nodes initially have energy values following an arbitrary distribution. In this work we use the distribution depicted in Fig. 1.5. The choice of space partitioning resolution (r) is critical. Intuitively a good choice is $D \leq r \leq R$, where D is the average distance between two neighboring SNs. For the structured as well as the unstructured scenarios (225 nodes and 42 x 42 units area), simulations with different r values showed that $r = 3$ $units$ allows for the most comprehensive eMap. This is about the average distance between two neighboring SNs in the structured scenario.

Evaluation Metrics

The performance of constructing global maps is commonly measured with respect to its completeness, efficiency and regioning accuracy.

- *Collection Completeness*: The ratio of nodes whose values are collected by the AN to the total number of SNs.

- *Collection Efficiency*: To measure collection efficiency we consider the number of energy messages per SN, i.e., the ratio of total number of energy messages sent by SNs to the number of SNs that received a REQ-beacon. We also consider the number of snapshots as overhead since it implies the reception of beacons by the SNs.

- *Region Accuracy*: To evaluate the regioning accuracy we compare the eMap constructed by the AN with the perfect eMap, i.e., the map constructed from complete energy information.

  To compute regions accuracy we either compare both maps either cell-wise or region-wise:

  (1) The Misclassification Cell Count (MCC). We count the number of grid cell in the map that have been misclassified due to incompleteness of collected data. To evaluate MCC we define reference map as the ideal map and the test map that is acquired through robot. We have one reference map for each scenario, i.e., one for structured scenario and one for unstructured scenario and we have two test maps for each scenario. The MCC can be defined by the following equation:

$$Ce = \sum_i \sum_j count(Cr_{ij} - Ct_{ij}), \tag{1.8}$$

  Ce is the total count of class cells that differ between the reference Cr and test class map Ct. The count() function returns '1' if the two cells do not belong to the same class else it returns '0'. Ce is the direct measure of correct classification of the grid cells into the classes and indirect measure of the accuracy of area and perimeter of the detected event area.

(2) Regional percentile error (RPE): RPE is the misclassified cells percentage for each region. RPE assesses the accuracy on regions level and is computed as follows. The regioning algorithm (Alg. 4) is used to detect various regions in both the original perfect map and the collected data map. Afterwards, each region from the collected map is compared to the perfect region to find out the percentile accuracy of each region.

Data Collection Performance

The results of our simulations are summarized in Table 1.1. We observe that the completeness of energy information is close to but lower than 100% in all scenarios. In structured and semi-structured scenarios, the incompleteness is due to MAC collisions. We also observe that our path planning has improves the completeness of the collected eMap, though it requires more snapshots in structured scenario. Furthermore, we observe the less number of snapshots in semi-structured scenario, which is direct result of optimizations in the proposed path planning algorithm for uniform random distribution of nodes. In unstructured scenarios, besides collisions, mobility leads to fast topology changes and therefore to additional message loss. We observe that lower COV_{th} values provide for higher completeness at the cost of higher number of snapshots. This is due to the fact that if COV_{th} is increased higher overlaps between snapshots is tolerated. After sufficient number of snapshots, the additional coverage will not be able to be higher than COV_{th} and no further snapshots are possible although some nodes have not received a REQ beacon. This results in higher efficiency (limited number of snapshots) but the completeness of collected information may suffer. The efficiency is 1.0 in all scenarios, given the fact that nodes that receive a REQ beacon from the AN respond with a single message per collection round, irrespective of the AN received it or not. The latency of gMAP is as expected in the range of minutes to hours. The latency for data

	Structured		**Semi-structured**	**Un-structured**	
	Zigzag	**TSP-Grid**	**TSP-Random**	**COV = 70%**	**COV = 90%**
Completeness [%]	94.7	98.2	95.1	88.4	81.3
Efficiency	1.0	1.0	1.0	1.0	1.0
#snapshots	25	41	20	61	43
Latency [min]	5.0	7.94	4.14	40.71	39.25

Table 1.1: Simulation results for gMAP

collection in unstructured scenarios is higher than that in structured scenarios. This is due to the fact the movement of the AN is random implying that more time is needed to cover the sensor field. Whereas in semi-structured scenario we observe the latency to be optimal corresponding to the less number of snapshots and resulting shorter path from the path planning algorithm. Generally, the simulation results confirm the gMAP efficiency and utility in the practical scenarios.

Regioning Performance

Given that energy levels are between 0 and 100%, and as it is likely to tolerate 10% difference within the single region, we use for regioning 10 classes of energy levels. Fig. 1.5(a) shows the isolines of the considered energy distribution. We use this map as a reference and compare different eMaps generated by our approach. In Fig. 1.5(b) we show the perfect choropleth eMap of the structured scenario. Obviously, choropleth is more expressive and comprehensive than the isolines.

Given the 10 possible energy class, the considered energy distribution results into an eMap of 9 regions. Fig. 1.6 shows the accuracy of each region formed from the collected data along with the overall accuracy of the eMap. Map and region accuracies are calculated according to the MCC and RPE metric, respectively. As the structured scenario has the highest level of completeness, its eMap is almost perfect and regions have a very high accuracy. Apart from 2 regions all other regions are perfectly formed. This high accuracy level proves the high performance of our path planning algorithm. In the semi-structured scenario the achieved accuracy is lower in comparison to the structured scenario since nodes are randomly deployed in the former. However, our path planning algorithm helped to collect enough data for each region to be considerably accurate with more than 90% of accuracy in each of 7 regions out of 9. From the map accuracy in the unstructured scenario, it is clear that data completeness is the lowes amon all scenarios. However, gMAP achieves 88.4% map accuracy for $COV_{th} = 70\%$ and 81.3% map accuracy for $COV_{th} = 90\%$. Though the selection of $COV_{th} = 70\%$ increases the number of snapshots, the data completeness increases as highlighted in Fig. 1.6. We observe there that all regions are more than 80% accurately formed in comparison to $COV_{th} = 90\%$ where we have a worst case situation of a region formed with just 50% accuracy.

1.6 Conclusions

We have presented gMAP, an extremely energy-efficient methodology that collects data of interest from the WSN and presents its geographical distribution as a map. Our approach is opportunistic as it exploits existing node mobility to collect data. Being mobility-assisted the collection process lasts for the time that mobile entities need to scan the whole sensor field. Therefore, data should be of high time relevance, i.e., do not change suddenly or radically in magnitude. As an example we focussed on the map of residual energy (eMap) since the battery depletion is a long-running process. Considering three representative scenarios, i.e., structured, semi-structured and unstructured, we showed the efficiency and the accuracy of our gMAP approach.

1.7 Open Issues

Though there exists a wide effort to design mobility-assisted collection of data of higher lifetime, there is still a need for further research work that we discuss in the following.

We provided an efficient, approximative and scalable path planning algorithm that performs in the minutes range on a conventional PC platform. As ANs may be much more restricted in resources, it is interesting to investigate further optimizations of our algorithm in order to sacrifice a tolerable accuracy for increased efficiency. This will be useful to achieve for different conventional platforms from laptops to PDAs to motes. An implementation of the algorithm on resource-constraint devices is a challenge for path planning algorithms. Therefore, a distributed computation of path planning on motes is an interesting open issue.

In this work, we focussed on controlling the mobility in space, which is important to provide for high data collection completeness. Further investigations of mobility control in time such as suggested in [57] will allow AN to autonomously adjust their motion to run time dynamics of the system and to the evolving tolerable data lifetime. The dynamic adaptation of mobility control in both time and space on resource-constraint ANs is a crucial research to allow for autonomous and pro-active reconfiguration.

The coordination between multiple ANs to optimize data collection is also a crucial research field that should be extended beyond the few work existing in the literature [62, 63]. In particular, it is an open issue how to fix the number of optimal AN in dependency of data lifetime, the WSN coverage area, movement properties of AN, and load balancing policies (buffer etc.). The data mule projet suggested a few of these optimizations such as identifying a set of congested nodes (with high data rates or lossy link etc.) so that the next tour is improved by adjusting the node speed for example [54]. In [60], the authors present a techniques to adapt the node speed to the varied data rates of different sensor nodes.

Another interesting adaptation issue for path planning is the adaptation to realistic network conditions at run-time. For example path planning is assuming a disc model for communication (i.e., fixed communication range), however, realistic radio propagation models while running the path planning algorithm are more appropriate. We believe that the ANs should react locally while traversing the path if communication perturbations occur.

In order to provide for continuous monitoring it is important to investigate update strategies to keep monitored status accurate. Usually, the collected samples become obsolete and need to be updated after varied collection time points in different spatial regions. A natural optimization of the path planning is to adapt it to the location of the next needed updates. We believe that AN can optimize their path if they can predict future profiles. One possibility is that ANs can weight the nodes in dependency of the urgency of the status update from the different nodes. However, we believe that for efficiency reasons ANs should weight the regions of the map and accordingly do a path planning on top of regions and then on node level.

Bibliography

[1] A. Khelil, C. Reinl, B. Ayari, F. K. Shaikh, P. Szczytowski, A. Ali, N. Suri Sensor Cooperation for a Sustainable Quality of Information *Pervasive Computing and Networking, Wiley & Sons (ACCEPTED FOR PUBLICATION)*, 2009

[2] W. Burgard, M. Moors, C. Stachniss, F.E. Schneider Coordinated multi-robot exploration, *IEEE Transactions on Robotics*,2005 ,vol. 21, pp 376-386

[3] R. Sugihara, R. K. Gupta Improving the Data Delivery Latency in Sensor Networks with Controlled Mobility *Distributed Computing in Sensor Systems*, Springer Berlin / Heidelberg, vol. 5067 of Lecture Notes in Computer Science,2008,pp. 386-399

[4] M. Bosse, N. Nourani-Vatani, and J. Roberts Coverage Algorithms for an Under-actuated Car-Like Vehicle in an Uncertain Environment *ICRA*, 2007.

[5] C. L. Bloebaum, and Ulrich Faigle Coverage Path Planning: The Boustrophedon Cellular Decomposition In *International Conference on Field and Service Robotics*, 1997.

[6] Marius M. Solomon Algorithms for the Vehicle Routing ans Scheduling Problems with Time Window Constraints *Operations Research*, 1987, 35(2), pp. 254-265

[7] http://neos.mcs.anl.gov

[8] A. Wächter and L. T. Biegler On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming In *Mathematical Programming* 106(1), pages 25-57, 2006

[9] David Applegate, Robert E. Bixby, Vaek Chvtal, and William J. Cook. Concorde tsp-solver. Technical report, http://www.tsp.gatech.edu/concorde.html, 2006.

[10] Khaled Elbassioni, Aleksei V. Fishkin, and Ren Sitters. On approximating the TSP with intersecting neighborhoods. In *Algorithms and Computation*, volume 4288 of *Lecture Notes in Computer Science*, pages 213–222. Springer Berlin / Heidelberg, 2006.

[11] Markus Glocker, Christian Reinl, and Oskar von Stryk. Optimal task allocation and dynamic trajectory planning for multi-vehicle systems using nonlinear hybrid optimal control. In *Proc. 1st IFAC-Symposium on Multi-vehicle Systems*, 2006.

[12] I.E. Grossmann and S. Lee. Generalized convex disjunctive programming: Nonlinear convex hull relaxation. *Computational Optimization and Applications*, 26(1):83–100, 2003.

[13] ILOG. *ILOG CPLEX 11.0, User's Manual*, 2007.

[14] Joseph S. B. Mitchell. A ptas for tsp with neighborhoods among fat regions in the plane. In *SODA '07: Proceedings of the eighteenth annual ACM-*

SIAM symposium on Discrete algorithms, pages 11–18, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[15] C. Reinl and O. von Stryk. Optimal control of multi-vehicle systems under communication constraints using mixed-integer linear programming. In *Proceedings of the. First International Conference on Robot Communication and Coordination (RoboComm)*, 2007. ICTS.

[16] Bo Yuan, M. Orlowska, and S. Sadiq. On the optimal robot routing problem in wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering*,

[17] A. Khelil, F. K. Shaikh, B. Ayari, and N. Suri. MWM: A map-based world model for event-driven wireless sensor networks. In *The 2nd ACM International Conference on Autonomic Computing and Communication Systems (AUTONOMICS)*, 2008.

[18] Y. Zhao, R. Govindan, and D. Estrin. Residual energy scan for monitoring sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC'02)*, 2002.

[19] Joseph M. Hellerstein, Wei Hong, Samuel Madden, and Kyle Stanek. Beyond Average: Toward Sophisticated Sensing with Queries. In *Second International Workshop on Information Processing in Sensor Networks (IPSN)*, page 553, 2003.

[20] Wenwei Xue, Qiong Luo, Lei Chen, and Yunhao Liu. Contour Map Matching For Event Detection in Sensor Networks. In *ACM SIGMOD*, 2006.

[21] R.A.F. Mini, M.d.V. Machado, A.A.F. Loureiro, and B. Nath. Prediction-based energy map for wireless sensor networks. *Elsevier Ad-hoc Networks Journal (special issue on Ad Hoc Networking for Pervasive Systems)*, 2005.

[22] I. Solis and K. Obraczka. Isolines: energy-efficient mapping in sensor networks. In *10th IEEE Symposium on Computers and Communications (ISCC)*, 2005.

[23] Xiaoqiao Meng, Thyaga Nandagopal, Li Li, and Songwu Lu. Contour maps: Monitoring and diagnosis in sensor networks. *Computer Networks, 50(15)*, 2006.

[24] Yunhao Liu and Mo Li. Iso-Map: Energy-Efficient Contour Mapping in Wireless Sensor Networks. In *The 27th International Conference on Distributed Computing Systems (ICDCS)*, 2007.

[25] M. Grossglauser and D. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking, 10(4)*, 2002.

[26] Y. Wang, H. Dang, and H. Wu. A survey on analytic studies of delay-tolerant mobile sensor networks. *Journal of Wireless Communications and Mobile Computing (WCMC) SI on Disruption Tolerant Networking for Mobile or Sensor Networks, 7(10)*, 2007.

[27] A. Khelil, F. K. Shaikh, A. Ali, and N. Suri. gmap: Efficient construction of global maps for mobility-assisted wireless sensor networks. In *Conference on Wireless On demand Network Systems and Services (WONS)*, 2009.

[28] Yonggang Jerry Zhao, Ramesh Govindan, and Deborah Estrin. Sensor Network Tomography: Monitoring Wireless Sensor Networks. In *Student Research Poster, ACM SIGCOMM*, 2001.

[29] Raquel A. F. Mini, A. A. F. Loureiro, and Badri Nath. The distinctive design characteristic of a wireless sensor network: the energy map. *Computer Communications, 27(10)*, 2004.

[30] E. Souto, R. Gomes, D. Sadok, and J. Kelner. Sampling Energy Consumption in Wireless Sensor Networks. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2006.

[31] A.H. Robinson, J.L. Morrison, P.C. Muehrcke, A.J. Kimerling, and S.C. Guptill. *Elements of Cartography.* John Wiley & Sons, New York, 1995. 6th Edition.

[32] Prabal Dutta, Jonathan Hui, Jaein Jeong, Sukun Kim, Cory Sharp, Jay Taneja, Gilman Tolle, Kamin Whitehouse, and David Culler. Trio: Enabling Sustainable and Scalable Outdoor Wireless Sensor Network Deployments. In *The Fifth International Conference on Information Processing in Sensor Networks (IPSN)*, 2006.

[33] S. M. LaValle. *Planning Algorithms.* Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.

[34] L. Lima and J. Barros. Random Walks on Sensor Networks. In *the 5th International Syposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, 2007.

[35] J. Nievergelt and P. Widmayer. Spatial data structures: Concepts and design choices. In *Handbook of Computational Geometry, Elsevier Science Publishers*, 2000.

[36] Boris A. Portnov and Daniel Felsenstein. *Regional Disparities in Small Countries.* Springer, Berlin Heidelberg, 2005. ISBN 978-3-540-24303-8.

[37] L.G. Shapiro and G.C. Stockman. *Computer Vision.* Prentice-Hall, Upper Saddle River, NJ, 2001. 978-0130307965.

[38] B. Krishnamachari, D. Estrin, and S. Wicker. Modelling Data-Centric Routing in Wireless Sensor Networks. In *USC Computer Engineering TR CENG 02-14*, 2002.

[39] P. Levis, N. Lee, M. Welsh, and D. Culler. Tossim: accurate and scalable simulation of entire tinyos applications. In *Proc. of SenSys*, 2003.

[40] N. Ramanathan, K. Chang, R. Kapur, L. Girod, E. Kohler, D. Estrin. Sympathy for the sensor network debugger. In *Proc. of SenSys*, 2005.

[41] S. Rost, H. Balakrishnan. Memento: a health monitoring system for wireless sensor networks. In *Proc. of IEEE SECON*, 2006.

[42] Jerry Zhao, Ramesh Govindan, and Deborah Estrin. Tomography: Monitoring Wireless Sensor Networks. In *In Student Research Poster, ACM SIGCOMM*, 2001.

[43] Jerry Zhao, Ramesh Govindan, and Deborah Estrin. Computing Aggregates for Monitoring Wireless Sensor Networks. In *In Proceedings of The First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, 2003.

[44] M. Woehrle, C. Plessl, R. Lim, J. Beutel, L. Thiele. EvAnT: Analysis and Checking of Event Traces for Wireless Sensor Networks. In *In IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC)*, 2008.

[45] Matthias Ringwald, Kay Römer, and Andrea Vitaletti. Passive inspection of sensor networks. In *In Proc. of DCOSS*, 2007.

[46] Guoliang Xing, Tian Wang, Zhihui Xie, Weijia Jia. Rendezvous Planning in Mobility-Assisted Wireless Sensor Networks. In *Proc. 28th IEEE International Real-Time Systems Symposium (RTSS)*, 2007.

[47] Guoliang Xing, Tian Wang, Zhihui Xie, Weijia Jia. Rendezvous Planning in Wireless Sensor Networks with Mobile Elements. In *IEEE Trans. Mob. Comput. 7(12)*, 2008.

[48] Guoliang Xing, Tian Wang, Weijia Jia, Minming Li. Rendezvous design algorithms for wireless sensor networks with a mobile base station. In *Proc. MobiHoc*, 2008.

[49] Guoliang Xing, Jianping Wang, Ke Shen, Qingfeng Huang, Xiaohua Jia, Hing Cheung So. Mobility-Assisted Spatiotemporal Detection in Wireless Sensor Networks. In *Proc. ICDCS*, 2008.

[50] Wenrui Zhao, Mostafa Ammar, and Ellen Zegura. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In *Proc. of ACM Mobihoc*, 2004.

[51] Mooi Choo Chuah, Peng Yang. A Message Ferrying Scheme with Differentiated Services. In *Proc. of MILCOM*, 2005.

[52] Hyewon Jun, Wenrui Zhao, Mostafa H. Ammar, Ellen W. Zegura, Chungki Lee. Trading Latency for Energy in Wireless Ad Hoc Networks Using Message Ferrying. In *Proc. Third IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW)*, 2005.

[53] Hyewon Jun, Mostafa Ammar, and Ellen Zegura. Power Management in Delay Tolerant Networks: A Framework and Knowledge-Based Mechanisms. In *IEEE SECON*, 2005.

[54] Arun A Somasundara, Aman Kansal, David Jea, Deborah Estrin, Mani B Srivastava. Controllably Mobile Infrastructure for Low Energy Embedded Networks. In *IEEE Transactions on Mobile Computing*, 2006.

[55] Arun A Somasundara, Aditya Ramamoorthy, Mani B Srivastava. Mobile Element Scheduling for Efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines. In *Proc. IEEE Real Time Systems Symposium (RTSS)*, 2004.

[56] A. Kansal, M. Rahimi, W.J. Kaiser, M. Srivastava, G.J. Pottie, D. Estrin. Controlled Mobility for Sustainable Wireless Networks. In *Proc. IEEE SECON*, 2004.

[57] A. Kansal, A. Somasundara, D. Jea, M. Srivastava, and D. Estrin. Intelligent fluid infrastructure for embedded networks. In *Proc. of MobiSys*, 2004.

[58] R.C. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: modeling a three-tier architecture for sparse sensor networks. In *Proc. of the First International Workshop on Sensor Network Protocols and Applications (SNPA)*, 2003.

[59] O. Tekdas, V. Isler, J.H. Lim and A. Terzis. Using Mobile Robots to Harvest Data from Sensor Fields. In *Proc. IEEE Wireless Communications*, 2009.

[60] Y. Gu, D. Bozdag, E. Ekici, F. Ozguner, C.-G. Lee. Partitioning-Based Mobile Element Scheduling in Wireless Sensor Networks. In *Proc. IEEE SECON*, 2005.

[61] Y. Wang and H. Wu. DFT-MSN: The Delay/Fault-Tolerant Mobile Sensor Network for Pervasive Information Gathering. In *Proc. of INFOCOM*, 2006.

[62] W. Cheng, M. Li, K. Liu, Y. Liu, X.-Y. Li, and X. Liao. Sweep Coverage with Mobile Sensors. In *Proceedings of the 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2008.

[63] D. Jea, A. Somasundara and M. Srivastava. Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *Proc. DCOSS*, 2005.

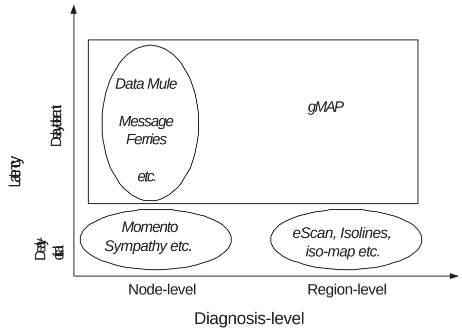
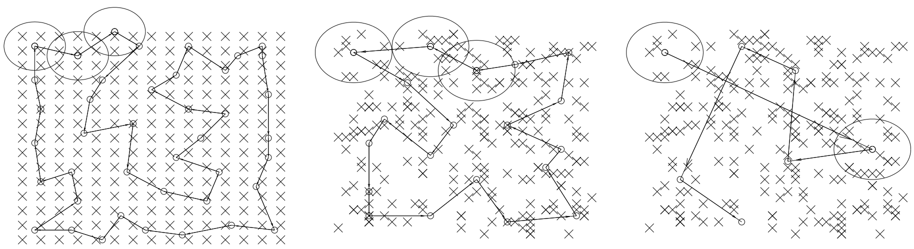


Figure 1.1: State of the art in problem space

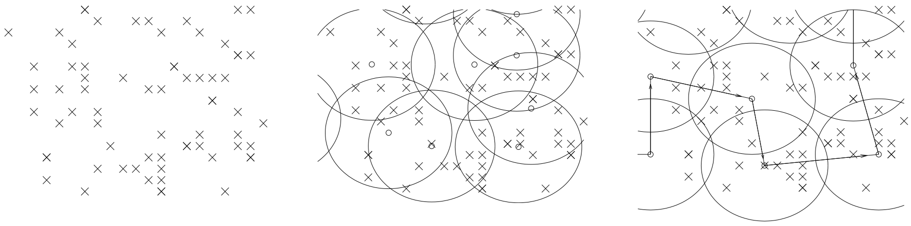


(a) Structured (grid deployment of SNs with controllable AN)

(b) Semi-structured (random SNs deployment with controllable AN)

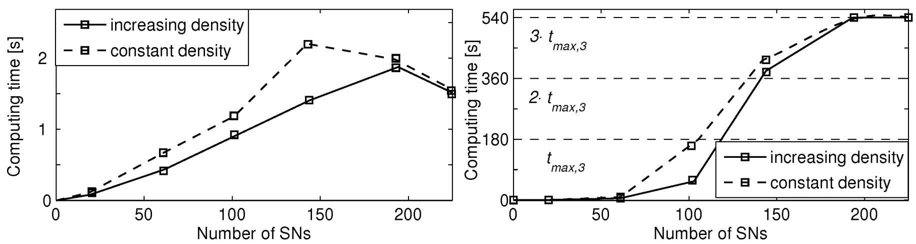
(c) Unstructured (unknown topology and unpredictable mobility)

Figure 1.2: Basic scenarios (nodes are represented by $\times$ and breakpoints by $\circ$), with results (in (a) and (b)) from the path planing algorithm as detailed in Section 1.4.3



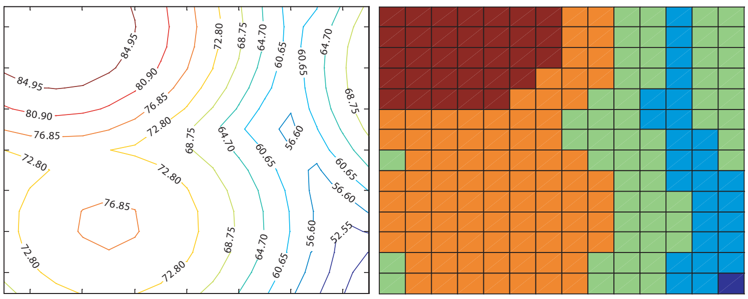
(a) SN positions (b) Breakpoint candidates (c) Overlap reduction and TSP-path

Figure 1.3: Visualization of the basic steps of our path planning



(a) NLP (get breakpoint candidates) (b) MILP (reduce overlaps)

Figure 1.4: Time to solve the NLP (STEP 1) and the approximating MILP (STEP 2)



(a) Perfect isoline eMap (b) Perfect choropleth eMap

Figure 1.5: Perfect maps

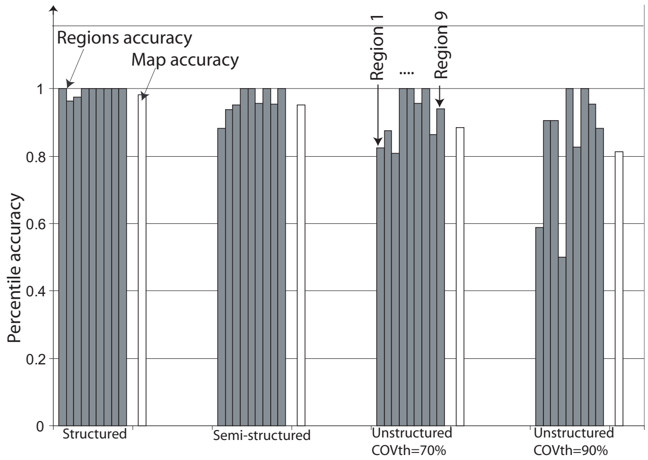


Figure 1.6: Regioning accuracy