# Energy Management in Solar Powered Wireless Sensor Networks

by

Benjamin Gaudette

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved January 2012 by the
Graduate Supervisory Committee:

Sarma Vrudhula, Chair
Aviral Shrivastava
Arunabha Sen

ARIZONA STATE UNIVERSITY

May 2012

## ABSTRACT

The use of energy-harvesting in a wireless sensor network (WSN) is essential for situations where it is either difficult or not cost effective to access the network's nodes to replace the batteries. In this paper, the problems involved in controlling an active sensor network that is powered both by batteries and solar energy are investigated. The objective is to develop control strategies to maximize the quality of coverage (QoC), which is defined as the minimum number of targets that must be covered and reported over a 24 hour period. Assuming a time varying solar profile, the problem is to optimally control the sensing range of each sensor so as to maximize the QoC while maintaining connectivity throughout the network. Implicit in the solution is the dynamic allocation of solar energy during the day to sensing and to recharging the battery so that a minimum coverage is guaranteed even during the night, when only the batteries can supply energy to the sensors. This problem turns out to be a non-linear optimal control problem of high complexity. Based on novel and useful observations, a method is presented to solve it as a series of quasiconvex (unimodal) optimization problems which not only ensures a maximum QoC, but also maintains connectivity throughout the network. The runtime of the proposed solution is 60X less than a naive but optimal method which is based on dynamic programming, while the peak error of the solution is less than 8%. Unlike the dynamic programming method, the proposed method is scalable to large networks consisting of hundreds of sensors and targets. The solution method enables a designer to explore the optimal configuration of network design. This paper offers many insights in the design of energy-harvesting networks, which result in minimum network setup cost through determination of optimal configuration of number of sensors, sensing beam width, and the sampling time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

# Chapter 1

# INTRODUCTION

Advances in microelectronics have made it possible to produce very low cost and low power active sensors. Consequently the deployment of a large network of active sensors over a large geographical area is now feasible and can be used for a variety of purposes such as environmental and structural monitoring or area surveillance [4].

A desirable feature of modern sensor networks is zero reliance on existing physical infrastructure such as power lines and network cables, but instead use chemical batteries for power and RF chips for wireless communication. This introduces a number of challenging problems in the management of such a network. A substantial body of research has been conducted in the area of low-power wireless sensor network (WSN) management at the physical, networking, and application layers [4].

Regardless of how energy efficient a battery powered sensor network is made, eventually the network will fail due to the limited power resource; and sensor nodes will either have to be replaced or repaired manually. This can be a costly procedure if the network is in a difficult to access area. A solution to this problem is to use energy-harvesting in conjunction with rechargeable batteries. This will reduce the cost by requiring smaller batteries for some measure of performance, or equivalently improve the performance for the same cost. Some work has gone into hybridizing sensor networks to use power from both a rechargeable battery and a renewable energy source. Currently, the most promising form of renewable energy is solar. Photovoltaic panels, more commonly known as solar panels, are becoming cheaper to manufacture and are capable of providing greater energy density than ever before thus allowing for more harvested energy from smaller, cheaper panels. These advances have made solar energy viable, and profitable [5].

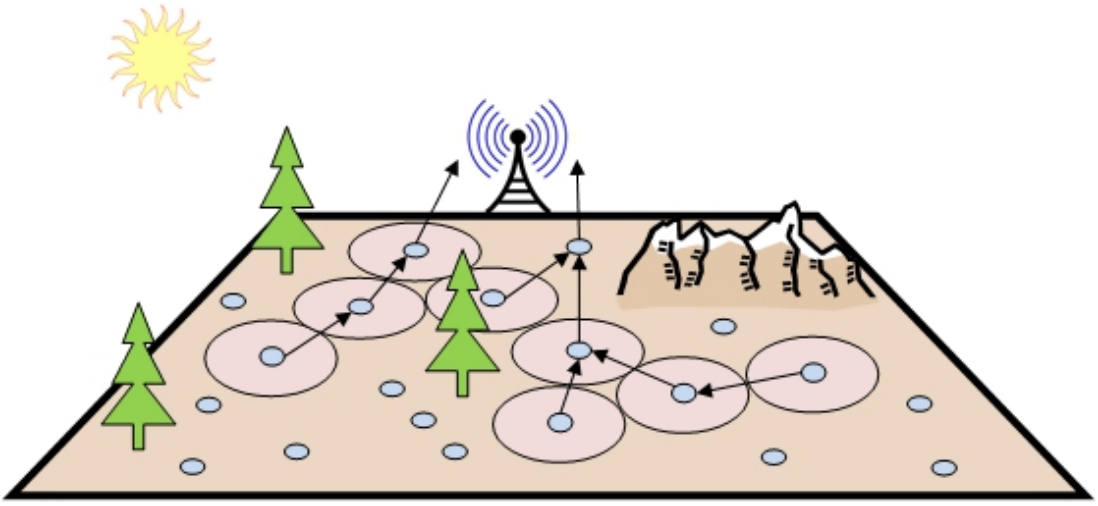


Figure 1.1: Sensor nodes scattered in a sensing region.

With these benefits of solar energy comes many new challenges. Energy-harvesting adds a degree of uncertainty to the task of managing the sensors (adjusting sensor radii, sampling intervals, maintaining connectivity, etc.) due to the unpredictability of the solar profile (cloud cover, shadows of buildings, routing paths, etc.). Thus the basic and important problem of guaranteeing a minimum coverage of targets becomes an even more important problem with the energy-harvesting sensors [6]. Also an optimal scheduling policy allows a designer to scale the battery, sensor cover region, and the solar panel size for each sensor node appropriately to minimize the network startup cost, while ensuring a minimum quality of coverage (QoC).

## 1.1 Sensor Network Architecture

A sensor network is comprised of numerous sensor nodes deployed over a geographic region such as in Figure 1.1. Thanks to the relatively cheap nature of individual sensor nodes, the number of nodes can be in the hundreds or more. Each sensor node is capable of monitoring a subset of the network region and producing data based on the state of the environment. This data is then routed to a centralized base station via

some given network protocol. It is then up to the base station to make the appropriate response to the data. The network architecture can vary greatly based on the operating environment, sensor/target location, and the physical hardware of the sensor nodes. Each of these factors is briefly described below.

### *Operating Environment*

A major draw to wireless sensor networks is the minimal existing infrastructure needed for deployment. For this reason sensor networks are often deployed in remote or hostile geographical regions or regions in which human presence can obscure observations. Examples of such regions are battlefields, oceans, volatile volcanic regions, and animal habitats. Furthermore, for solar-powered WSN's, the environment can play a huge role in solar energy harvesting. Basic geographical location can alter solar flux along with humidity, cloud cover, and any obstructions (e.g. trees, buildings) that may be present. The vast number of applications and operating environments play an important role in the sensor hardware and network design.

### *Node Deployment*

A critical factor to the performance and operations of a sensor network is the deployment layout; that is the locations of the sensors and the targets as well as the regions that each sensor is capable of monitoring. Examples of deployment types are statically placed by hand, or randomly deployed by an aerial vehicle. In addition to the initial deployment, networks must be able to adapt to changing geographic topology sensor node failures, and additional sensor node deployments. For these reasons it is important for sensor network algorithms to be flexible to changes and self-organizing under any deployment conditions and for large numbers of sensor nodes.

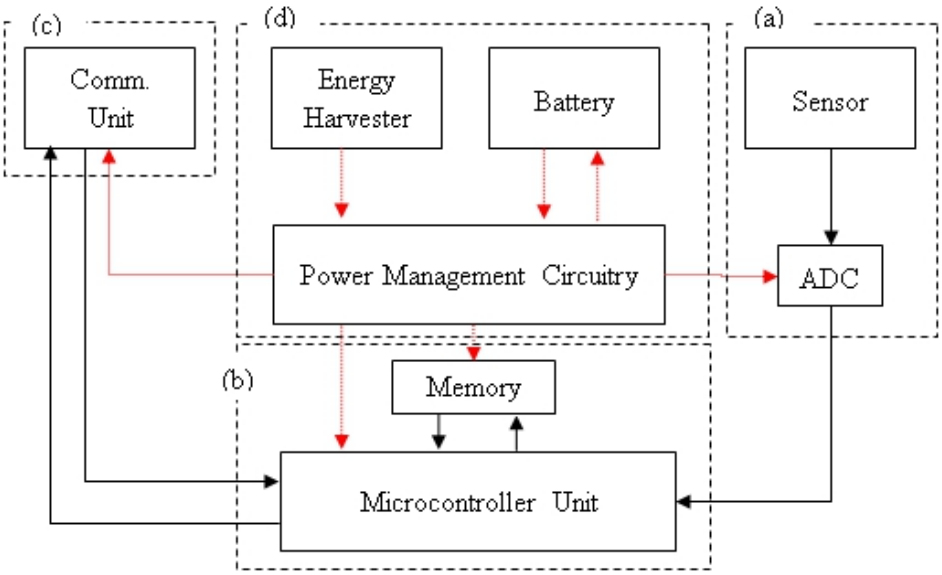



Figure 1.2: Components of a sensor node.

*Hardware Constraints*

A sensor node is a highly modulized unit which is customized for the specific sensor network application. Figure 1.2 illustrates a general sensor node design such that data communication is shown with solid black lines while energy flow is given by dotted red lines. Key elements of a sensor node are (a) a sensing unit or an array of sensing units, (b) a processing unit, (c) a communication unit, and (d) a power unit with possible energy harvesting hardware. These elements are detailed in Figure 1.2 by dashed boxes with the appropriate alphabetic labels.

The sensing unit is typically an analog sensor combined with an analog to digital converter (ADC). Sensors vary widely by applications. In general there are two types of sensors - active and passive. Active sensors are sensors which interact with the environment via electromagnetic/sonic waves or physical interactions. For this reason, active sensors typically have much higher energy consumption than passive sensors.

Typically the field of vision of such sensors is narrow (less than 30 degrees) and referred to as narrow beam sensors. Examples of active sensors include ultrasonic and radar distancing sensors. On the other hand, passive sensors are sensors which receive power from the phenomena that is being monitored. The only power necessary for such sensors is the amplification circuitry to view the signals. In many cases, passive sensors are omni-directional. Acoustic monitors are an example of passive sensors.

Regardless of the type of sensors, the node uses a communication unit such as a radio frequency (RF) IC to communicate sensed data with other sensor nodes and with a centralized base station. The communication unit may also provide sensor location when needed either directly or through a location finding process [7]. The power unit is typically a chemical battery which may be restored by the use of energy harvesting units. Additional circuitry may be required to ensure the battery receives and produces the correct voltages and currents. Ideally the power unit should also provide the processing unit with the state of the battery and the energy harvesting information. The processing unit is responsible for controlling all of the individual units and ensuring the network application is achieved.

## 1.2 Prior Related Works

The *Operational Range Assignment Problem* for solar powered sensor networks is a general version of the target cover problem, which takes advantage of predictable renewable energy sources [8]. The simplest form of the operational range assignment problem is the cover problem where there are only two ranges for sensing - 0 (off) or $r$ (on). One of the most intuitive definitions of the cover problem was provided by Klee and solved by O'Rourke [9]. Klee referred to this as the Art Gallery Problem and it asks: given a floor plan of an art gallery, how many stationary guards are needed to

monitor every exhibit in the art gallery if all guards have a known fixed field of vision. This was optimally solved in 2D space [9].

The cover problem has been greatly extended and modified since the Art Gallery Problem. In sensor networks, the sensor nodes or "guards" already have fixed locations just like the targets or "art exhibits". Therefore, the objective is to find a subset of these nodes such that all targets are covered. In the case where all sensors have a fixed sensing radius, full cover verification can be determined efficiently through the use of techniques like binary decision diagrams [10] and perimeter cover methods [11].

When considering the sensor nodes' limited energy supply, the cover problem extends to the cover life time problem. The goal of this problem is to extend the time in which all targets are covered for the maximum duration. This is typically addressed by minimizing total energy consumption [12, 13] or prolonging the weakest node's life time [10].

One can further constrain the cover problem by considering connectivity. In traditional battery powered sensor networks a great deal of research has been conducted in the area of network connectivity. Liu et. al. [14] considers both the coverage and connectivity problem in wireless sensor networks. They provide a randomized scheduling algorithm which provides probabilistic levels of QoC. Connectivity is then ensured by turning on additional sensor nodes until all nodes are connected to the sink. Zhang et. al. [15] proposed an alternative solution to the connectivity and coverage problem called Optimal Geographical Density Control (OGDC). The authors proved that any sensing network with full area coverage would be connected if the radio range of the sensors was twice that of the sensing range assuming all sensors used the same sensing range; their solution assumes this is a network property. Their distributed solution constructs a cover schedule via a series of request messages and volunteering to determine

which sensors are on and off. Simulations show that the algorithm outperforms other similar algorithms.

In [8], the life time problem was extended further by introducing multiple discrete sensing ranges that each sensor may choose from at any given time. This paper refers to this problem as the *operational range assignment problem*. The authors have shown this problem to be NP-Complete and have provided several heuristics for approximate solutions. One approximate solution used a centralized linear programming (LP) solution, which finds a series of valid covers that maximize the lifetime of the network. The second approximate solution used a greedy heuristic, in which covers were constructed by increasing a sensor's sensing radius sequentially until maximum number of targets is covered. The results produced by this method were inferior to the LP solution, because the constructed covers required a wide variation in sensing ranges among the sensor nodes. Although this result is valid for a linear sensor power model, in realistic scenarios, where sensor power consumption is at least quadratic with sensor radii, the large sensing radii will quickly deplete the sensors resources.

The above solution was improved with the addition of fuzzy sensor location knowledge in [16]. The proposed solution used a distributed approach to solve the operational range assignment problem similar to the greedy method used in [8]. In this method, the first phase consists of sensors increasing their radii in the order of available battery life until all targets are covered. In the second phase, sensors' radii are decreased, while ensuring that full coverage is still met. Although this method increased the network lifetimes substantially, it does not include the possibility of energy harvesting sources.

There has been some focus on increasing network lifetime through message routing in solar powered networks. Niyato et. al. [17] explored the unpredictability of

energy harvesting via solar radiation. With the use of Markovian models and game theory, cooperation between sensors was better established to minimize the losses in message passing, and thus, increasing the overall energy-efficiency of the network. In [18], a set of routing protocols for the battery powered sensor networks were introduced, whose job is to prevent passing messages through areas of the network which have reduced solar energy. A gradient was formed at each sensor node, which determined the subsequent path of a message leading to the destined receiver. Results show that there were considerable energy savings in shifting the burden away from resource limited nodes.

## 1.3 Our Contribution

To summarize, the following are the key contributions of this paper:

1. This thesis introduces the concept of optimal scheduling of sensor radii in a solar powered network that maximizes the minimum cover over the operational time period of 24 hours. Furthermore, this work considers network connectivity as a constraint of the cover problem; a critical necessity to sensor networks overlooked by many when considering the cover problem.

2. The radii scheduling problem is formulated as a nonlinear optimal control problem while the connectivity problem is expressed as a linear control problem. The intersection of these two problems defines the feasible search space for the overall problem. Based on certain useful characteristics of the problem, a near optimal approximate method is described. The problem is solved as a binary search of quasi-convex optimization problems solved over all time intervals. The search is over the minimum cover. The proposed solution outperforms the naive DP ap-

proach by a factor of **60** in computation speed, while maintaining the accuracy of solution to within **8%** of the optimal solution.

3. Several design space exploration experiments are described that offer new insights in the design and deployment of sensor networks that employ energy-harvesting.

## 1.4 Notation and Terminology

For consistency, terms and notations used in this document will be as follows: when referring to matrices and vectors, **bold** script will be used. Additionally, when an individual element of a set is referenced, subscript will be used. Descriptive tags are denoted by superscripts.

### *Covering Terms*

- **S** - set of all sensor nodes, including their locational information. The size of this set is denoted by $N$.
- **T** - set of all targets, including their locational information. The size of this set is denoted by $M$.
- $K$ - number of discrete time intervals dividing the total operational time.
- $\tau$ - length of each time interval.
- $\mathbf{r}(t)$ - vector or radii corresponding to the elements of **S** at a given time, $t$. $0 \leq r \leq r^{max}$.
- $\theta_n$ - angular direction narrow-beam sensor $n$ is facing. All $\theta_n$ should be taken with respect to the same reference vector.

- $\theta_{nm}$ - angular direction target $m$ is in respect to sensor $n$. All $\theta_{nm}$ should be taken with respect to the same reference vector.

- $\phi_n$ - beam width or beam angle of sensor $n$.

- $\zeta(\mathbf{r}(t))$ - cover function. Given a vector of radii, $\mathbf{r}(t)$, this returns the number of elements covered in $T$.

- $\zeta^{min}$ - minimum number of targets covered at any point in time during the operation time.

*Battery and Power Terms*

- $\mathbf{B}(t)$ - vector of residual battery energy corresponding to the elements of $\mathbf{S}$ at time $t$. $0 \leq B_n(t) \leq B^{max}$.

- $\mathbf{B^{init}}$ - initial residual battery energy of the $N$ sensors.

- $\mathbf{B^{min}}$ - minimum required battery level, such that $\mathbf{B}(t) \geq \mathbf{B^{min}} \; \forall \, t$.

- $\mathbf{P^{sol}}(t)$ - vector of harvested solar power for each of the $N$ sensor nodes at time $t$.

- $\mathbf{P^{sen}}(\mathbf{r}(t))$ - power consumed by the $N$ sensors at time $t$ given a the vector of radii, $\mathbf{r}(t)$.

*Connectivity Terms*

- $\mathbf{S^{on}}(t)$ - subset of $\mathbf{S}$. This set contains all sensors such that $r_n(t) > 0$.

- $\mathbf{S^{base}}(t)$ - contains all base stations for the network. In other words, all locations in which the elements of $\mathbf{S}$ may deliver sensed data.

- $\mathbf{x}(t)$ - $N \times N$ matrix where each element, $x_{n_1 n_2}$, corresponds to the amount of data (in messages) that sensor $n_1$ sends to $n_2$.

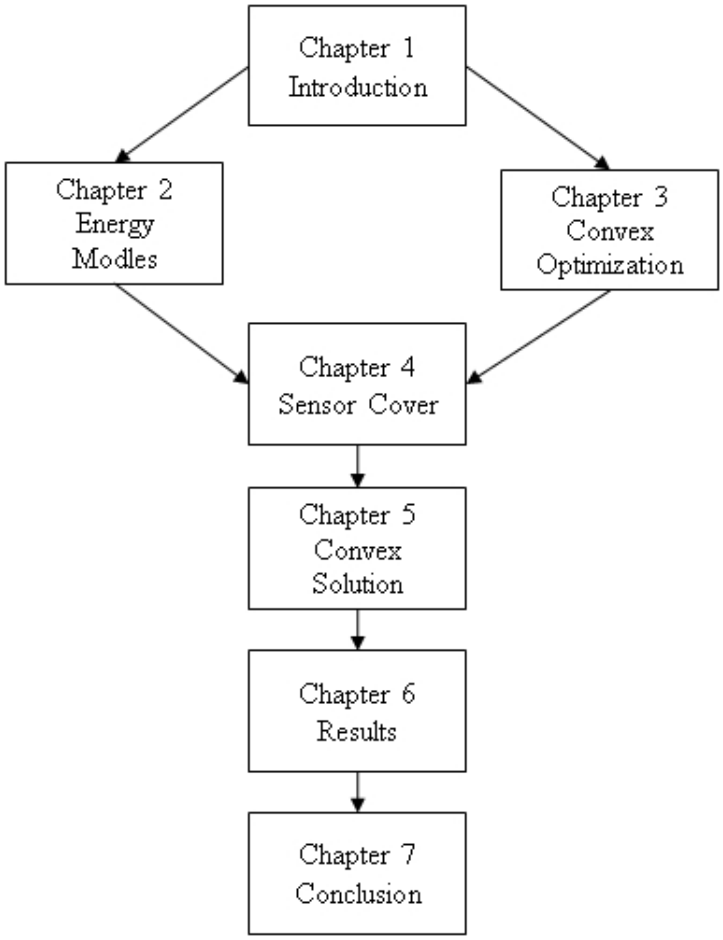



Figure 1.3: Thesis structure.

- $D_n(t)$ - amount of data sensor $n$ must transmit at time t, due to covering targets. Effectively this is dictated by the number of messages.
- $\mathbf{E^{RX}}(t)$ - $N \times N$ matrix, where each element, $E^{RX}_{n_1 n_2}$, corresponds to the amount of energy $n_2$ needed to receive one message of data from $n_1$.
- $\mathbf{E^{TX}}(t)$ - $N \times N$ matrix, where each element, $E^{TX}_{n_1 n_2}$, corresponds to the amount of energy $n_1$ needed to transmit one message of data to $n_2$.
- $\mathbf{E^{com}}(t)$ - total amount energy spent from communications.

## 1.5 Content Outline

Figure 1.3 shows the report's flow. This introduction gave basic background information on solar-powered WSN's and several versions of the target cover problem. Furthermore, motivation for this work was presented. The required background knowledge is presented in Chapters 2 and 3. In Chapter 2, the energy models used for the sensor nodes and the solar energy are presented and justified. In Chapter 3 a brief introduction to optimal control theory and convex optimization is given. With these two chapters, enough information is known to understand the operational range assignment problem for solar powered WSN's formally defined in Chapter 4 and the presented solution is in Chapter 5. Simulation results are discussed in Chapter 6. Finally results and contributions are summarized in Chapter 7.

# Chapter 2

# SENSOR MODELS

As discussed in Chapter 1, sensor nodes are composed of several power hungry components such as active sensors, radios, and microcontroller units. Furthermore, even with energy harvesting systems, nodes only have a finite amount of energy resources for these components at any given point in time. Hence, to maximize the quality of cover of the network, accurate estimations of energy resources and demands is critical. In this chapter, the energy and power models used in this work are presented along with the formal mathematical representation of the cover model.

## 2.1 Solar Profile

**Definition 2.1.1.** *A solar profile, $P_n^{sol}(t)$, is the available solar power via the solar panel throughout the operating time period for sensor node n.*

In practice, this profile is a stochastic process, however, the theoretical maximum solar power profile (also called the ideal solar profile) may be modeled with

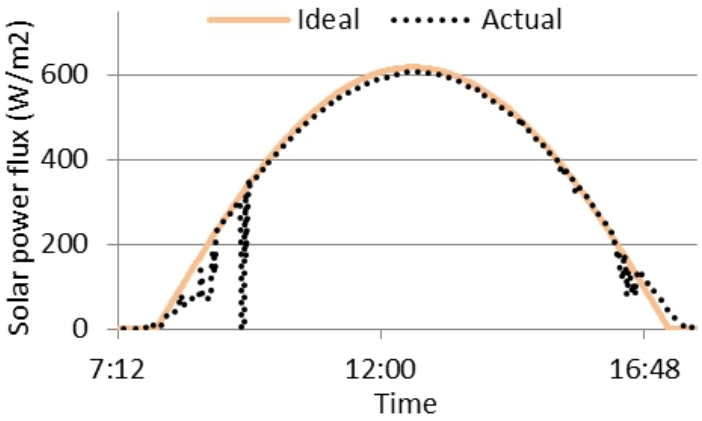



Figure 2.1: Ideal and actual [1] solar power profiles observed in Phoenix, Arizona on January 8, 2011

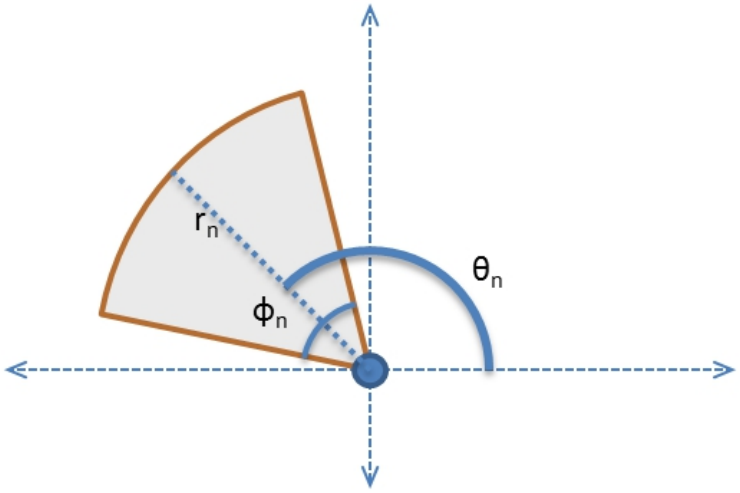



Figure 2.2: The sensing region of an active, narrow beam sensor.

the knowledge of the Sun's and the Earth's physical properties (speed, rotation, shape, and so on). This thesis uses the models from [19] and [20] to model the ideal solar radiation. Figure 2.1 shows the ideal and actual solar profiles for January 16, 2011 in Phoenix, Arizona [1]. Because the profile may be any process (random or otherwise), it is important to keep energy harvesting algorithms flexible enough to handle large changes in power supply. The efficiency of most modern solar panels is found be between 10-15% [21], and the same will be assumed in this work.

## 2.2 Sensor Power

**Definition 2.2.1.** *The power consumed by sensor n to monitor a distance r, $P_n^{sen}(r)$, is a direct function of the sensing distance.*

The sensor system assumed in this work consists of active, narrow-beam sensors such as an ultrasonic sensor or a radar. The area in which a narrow-beam sensor is capable of monitoring can be expressed as a cone assuming uniform sensing distance along the beam-width $\phi$ (also referred to as beam angle). Figure 2.2 illustrates the

area which may be covered. One can note that as $\phi$ approaches $2\pi$; the narrow beam sensor becomes a omni-directional sensor. In this work we assume that $\phi$ is fixed and is given; however, the sensing radius is variable with exponentially increasing cost to the residual battery life.

Narrow-beam sensors work by transmitting data in the form of waves to detect the presence or the absence of a target. If the target is present, the sensor will receive either a response from the target, or the remnants of the reflected/scattered data the sensor originally sent [22]. The relationship between the transmission power and the received power is given by the Friis transmission equation [23],

$$\frac{P_r}{P_t} = G_r G_t \left(\frac{\lambda}{4\pi r}\right)^{\alpha}. \tag{2.1}$$

$P_r$ and $P_t$ represent the power of the signal at the receiver and the transmitter respectively. Likewise, $G_r$ and $G_t$ represent the respective gain factors. $\lambda$ represents the signal wavelength, $\alpha$ is the degradation exponent between 2 and 5 that is experimentally determined, and $r$ is the distance between the transmitter and the receiver. Note, that one may need to double the value of $r$ if the sensor only receives remnants of the original signal. This is owed to the fact that the signal must make not only reach the target, but also return to the sensor node. The above equation may be rearranged to find the optimal transmission power for a given transmission distance and minimum receiving power, as shown below

$$P_n^{sen}(r) = \left(\frac{P_r^{min}}{G_r G_t}\right)\left(\frac{4\pi r}{\lambda}\right)^{\alpha}. \tag{2.2}$$

Alternatively, for dish based radar systems, the following equation is equivalent to (2.2).

$$P_n^{sen}(r) = \frac{P_r^{min}(4\pi)^2 r^4}{G_r G_t A_r \sigma F^4}. \tag{2.3}$$

$A_r$ is the area of the receiver's dish, $F$ is the propagation factor (F = 1 in a vacuum) and $\sigma$ is the scattering coefficient of the target. For simplicity, we ignore radio in-

terference between simultaneously active sensors, assuming that such interference is managed by the underlying MAC layer (e.g., through an appropriate TDMA or FDMA mechanism). Channel access protocols for interference mitigation in wireless sensor networks (WSN) are readily available.

The sensor's power can now be expressed simply in the form shown in (2.4) where $\alpha$ and $\beta$ are based on the properties of the sensor and $\mu$ is the average power requirement of all other system components, except the communication unit. These values are found through experimentation.

$$P^{sen}(r) = \beta r^{\alpha} + \mu. \tag{2.4}$$

The constant $\mu$ in (2.4) can be used to account for energy consumption related to channel access, data processing (e.g., fusion), and inter-sensor communications.

## 2.3 Radio Frequency Communication

RF communications and power requirements have been studied extensively. One may take consideration to link budget, phase noise, start-up time, data rate, channel fading and channel interference when determine an apt energy model for RF communications [24]. These effects are ultimately all dependent on the communication protocol used by the sensor network and since this work does not try to create such an extensive protocol, all effects can be considered constant.

Furthermore, if the location of sensors are fixed and known along with message size and communication speeds, we may construct an $N \times N$ matrix, $E^{TX}$, corresponding the transmission energy requirements to send a single message of data. Once again we may use Friis law (2.1) to find the transmission power required to send from one node to another. Figure 2.3 summarizes the results from [25] which experimented with Texas Instruments' CC2500 [26] to find the range capabilities of their RF chip. This

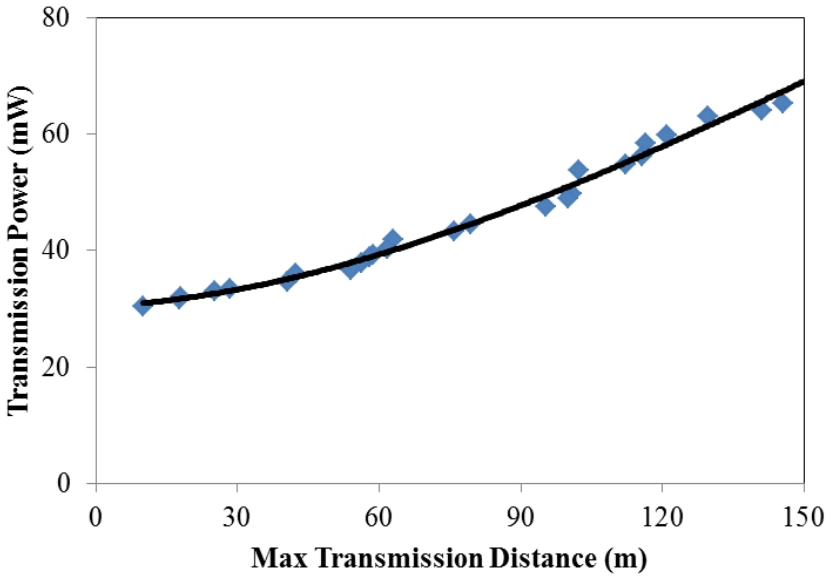



Figure 2.3: Transmission power vs. Range for TI's CC2500 RF IC.

data conforms to (2.4) if $\alpha = 3$ and $\beta = 4.176 \cdot 10^{-10} W/m^3$. A similar matrix to $E^{TX}$ can be constructed for the energy requirements to receive each message, $E^{RX}$. The total energy expended on RF communications may be expressed as the linear model

$$E_n^{com} = \sum_{i=1}^{N} E_{ni}^{TX} \chi_{ni} + \sum_{i=1}^{N} E_{in}^{RX} \chi_{in}, \tag{2.5}$$

where $\chi_i j$ is the number of messages sensor $i$ sends to sensor $j$.

## 2.4 Battery Model

The battery model used in this work is a simple linear battery model (linear charging and discharging) with no energy loss or leakage. However, we will show that our solution can easily accommodate the more realistic models that account for *rate dependent capacity* and temperature dependence [27, 28, 29] in latter sections. For short term operations, such as the 24 hour target monitoring application addressed in this paper, the benefits of using a more complicated battery model are negligible and would only serve to increase the time complexity of our solution. It will be shown in later sections

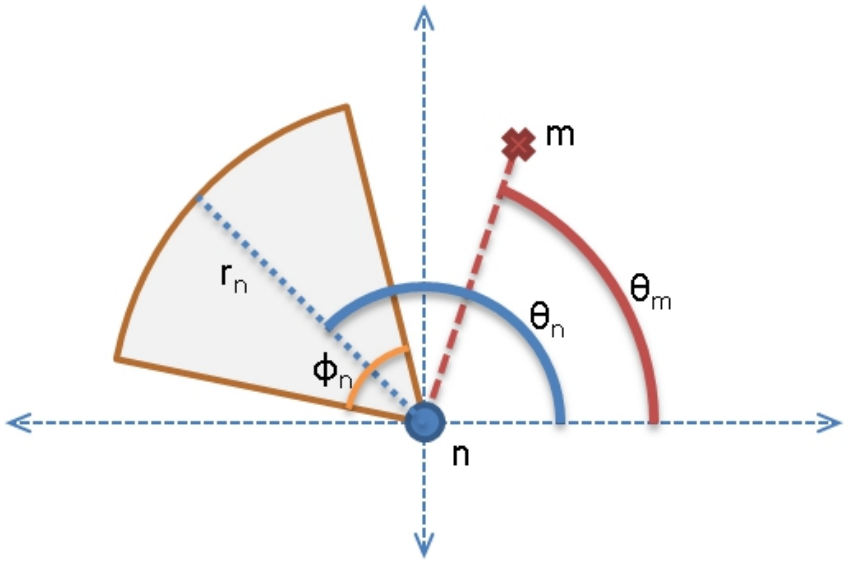



Figure 2.4: An active sensor and relevant terms.

that the proposed solution can work for any battery model with monotonic charging and discharging profiles. The energy in the battery at time $t$ for sensor $n$ is given by

$$B_n(r_n,t) = B_n(0) + \int_0^t (P^{sol}(z) - P_n^{sen}(r_n(z)))\, dz - E_n^{com}(t), \tag{2.6}$$

where $E_n^{com}(t)$ is the energy spent routing messages upto time $t$.

## 2.5 The Cover Function

In Chapter 1, a level of cover, $\zeta$, was mentioned. This section will formally define $\zeta(\mathbf{r}(t))$, the cover function. A cover model defines the percentage or the total coverage of all targets for a given vector of sensor radii. It depends on the location of sensor nodes and targets. Equation (2.7) defines the coverage of a single target $m$ by a single omni-directional, active sensor $n$.

$$\zeta(r_n,n,m) = \begin{cases} 0, & \text{for } r_n < 2 \times d_{nm} \\ 1, & \text{for } r_n \geq 2 \times d_{nm} \end{cases} \tag{2.7}$$

where $d_{nm}$ is the distance between sensor $n$ and target $m$. Note that the constant factor 2 comes from the sensor being an active sensor (e.g. radar), as the sensing signal travels to the target and then is reflected back to the sensor node. For the more general case of narrow beam sensors, the cover function is defined by

$$\zeta(r_n, n, m) = \begin{cases} 0, & \text{for } r_n < 2 \times d_{nm} \text{ or } \theta_m \notin [\theta_n \pm (\phi_n/2)] \\ 1, & \text{for } r_n \geq 2 \times d_{nm} \text{ and } \theta_m \in [\theta_n \pm (\phi_n/2)] \end{cases} \tag{2.8}$$

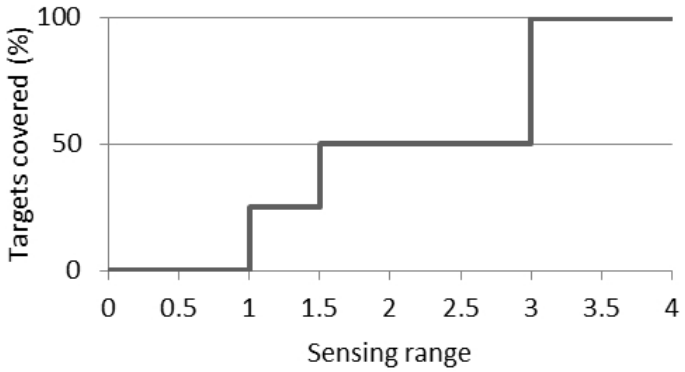
where $\theta_n$ is the angular direction that sensor $n$ is facing, and $\theta_m$ is the angular direction target $m$ is in respect to $n$ with the same reference vector used to find $\theta_n$. $\phi_n$ represents the beam width in radians of sensor $n$. $\theta_m$, $\theta_n$, and $\phi_n$ are all expressed in radians between 0 and $2\pi$. Note, care must be taken to ensure calculations fall in this range. An example of these values are shown in Figure 2.4 where angles are taken in reference to the positive x-axis.

The cover function for the entire network is defined as the total number of targets in $T$ covered by the sensors in $S$ with radii $\mathbf{r}$, is given by

$$\zeta(\mathbf{r}) = \sum_{m \in T} \max_{n \in S} (\zeta(r_n, n, m)). \tag{2.9}$$

Figure 2.5 (b) shows the cover function $\zeta$ for the 1-D sensor network shown in Figure 2.5 (a) assuming omni-directional sensors are used. Notice the discrete nature of the cover function with respect to sensor radius. This is because the cover function is defined as the number of targets covered, and for certain sensing ranges, it is possible not to cover any new targets.


(a) 1-D sensor network. The sensor location is denoted by a square and target locations by diamonds.




(b) The corresponding cover function.

Figure 2.5: 1-D sensor network and its cover function.

Chapter 3

# OPTIMAL CONTROL THEORY AND CONVEX OPTIMIZATION

In this chapter, a brief introduction to optimal control theory and convex optimization is provided. We define basic information concerning optimal control theory and various solutions to some optimization problems including dynamic programming, convex optimization, and linear programming.

## 3.1 Introduction

In many cases, a system design requires an optimization of a given objective to provide the best quality of service or a minimal cost to the operator. This type of system design is achievable with optimal control theory [2]. Optimal control theory is an extension of calculus of variations with foundations made by Richard Bellman and Pontryagin in the 1960's. In optimal control theory, one wishes to achieve mathematical optimization by deriving an optimal control policy. Optimal control can be defined by

$$\mathbf{u}^*(t) = \mathbf{f}(\mathbf{x}(t), t). \tag{3.1}$$

That is the optimal control u* is a collection of time varying differential equations each a function of the state, x(t), and time, t. Much like the optimal control, the state evolves with time and is defined by another collection of differential equations. This optimal control may be found by solving

$$\min J = h(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t)dt \tag{3.2}$$

$$s.t.\ \dot{\mathbf{x}}(t) = a(\mathbf{x}(t), \mathbf{u}(t), t) \tag{3.3}$$

$$b(\mathbf{x}(t), \mathbf{u}(t), t) \leq 0, \tag{3.4}$$

$$c(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) = 0, \tag{3.5}$$

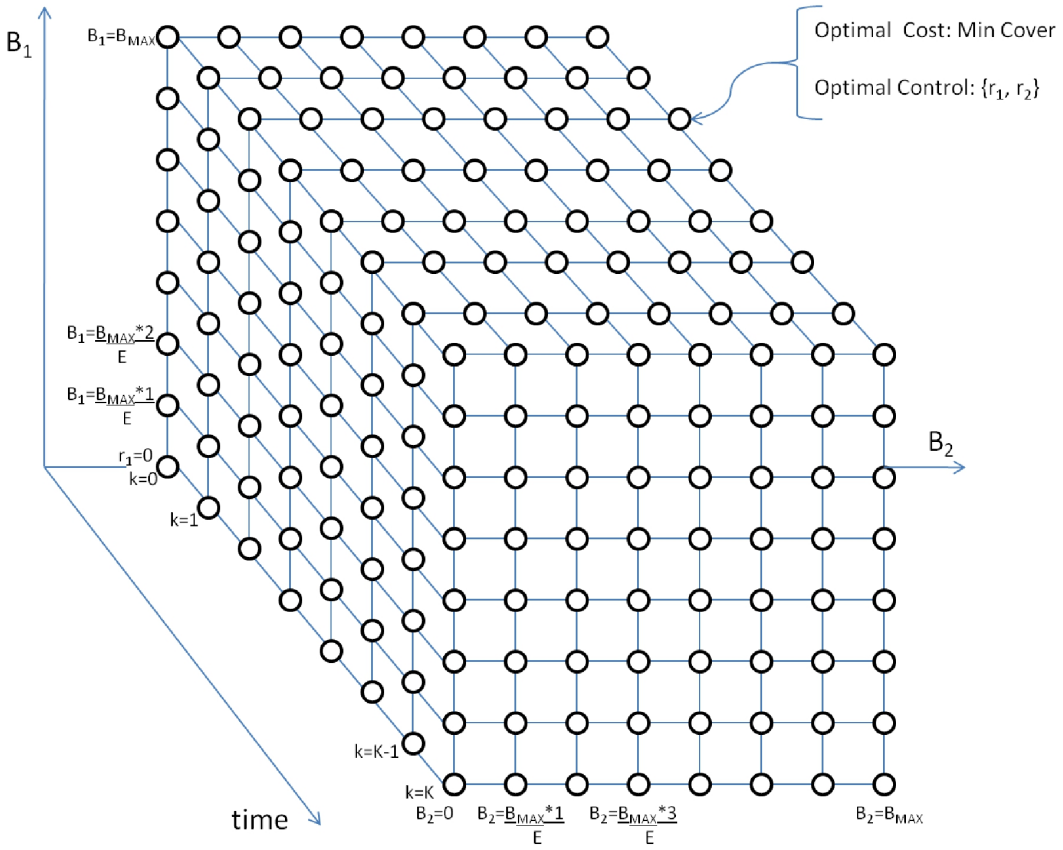



Figure 3.1: The search space used in dynamic programming applied to the QoC problem [2].

where (3.3) is the state transition model, (3.4) is the path constrains, and (3.5) is the boundary conditions. In the general case, this type of optimization is very difficult to achieve for higher dimensional problems, therefore it is important to have efficient methods of solving optimization problems for special cases. This chapter explores various solutions to special types of optimal control problems beginning with the most general solution, dynamic programming, and progressing to more restrictive, specialized solutions like convex optimization and linear programming.

## 3.2 Dynamic Programming

Optimal control theory is largely based on the principle of optimality, that is, an optimal control policy has the property that regardless the current state and past decisions; the remaining control may form an optimal policy with regard to the current state. Equivalently this is saying that optimal control problems have optimal substructures. These concepts are the basis for the Hamilton-Jacobi-Bellman equation

$$J = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(\tau), u(\tau), \tau) d\tau, \tag{3.6}$$

where J is the total cost, and h and g are specified functions which produce a cost given the current state, time and control.

Dynamic programming (DP) takes this concept and applies it to computational science [2]. First, the states, the time, and the controls are discretized if not already so. It can be shown that when given enough levels of discretization, the approximations will approach their continuous solutions. Following this, a DP solution first calculates and stores $h$ from (3.6) for each possible state at time $t_f$. Following this, DP works backwards from time $t_f$ to $t_0$ calculating and storing

$$J_k^* = \min_{u(k)} \{g(x(k), u(k)) + J_{k+1}^*(a(x(k), u(k)))\}, \forall x(k), \tag{3.7}$$

where $a$ is a function that when given the current state and a control vector, will determine the state in the following time instance, $k+1$. In the DP methodology, both the optimal cost, $J_k^*$ and the corresponding optimal control, $u^*(k)$ must be stored. This process is illustrated by Figure 3.1, where the states correspond to discrete levels of residual battery energy and controls to sensing radii.

It should be noted however, that the general DP solution is pseudo-polynomial in complexity meaning that the input is exponential in length. Given that time is discreatized to $K$ instances, $\mathbf{x}$ is discretized to $X$ instances, and $\mathbf{u}$ is discretized to $U$ instances,

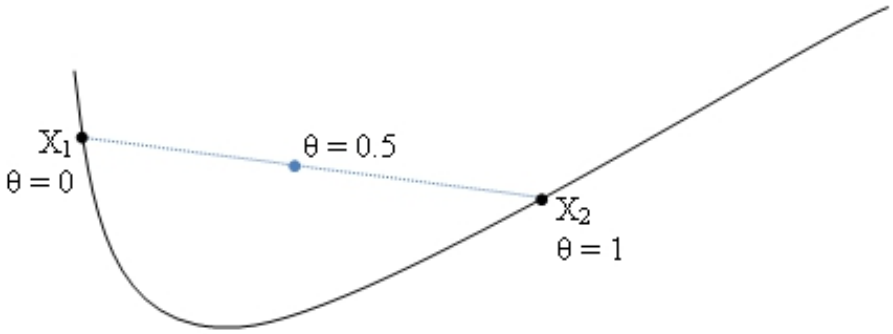



Figure 3.2: A convex function [3].

then $t$ can be shown that the run time of the general DP algorithm is $O(KX^{|\mathbf{x}|}U^{|\mathbf{u}|})$. For many instances, this is perfectly acceptable (such as in the shortest path problem); however, for the sensor cover problem where the size of $x$ and $u$ may be in the hundreds or thousands, this is simply not an efficient method.

## 3.3 Convex Optimization

### *Convex Optimization Problems*

A special case of optimization is when your optimization problem is dictated only by a set of convex functions. A function, $f : X \to \mathbf{R}$ defined on a convex set $X$, is said to be convex if given any two points, $x_1, x_2 \in X$ and any $\theta \in [0,1]$ [3].

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta) f(x_2) \tag{3.8}$$

Figure 3.2 illustrates this meaning. To this end, we can formally define a convex optimization problem to be in standard form

$$min_x \; f(x) \tag{3.9}$$

$$s.t. \; \mathbf{g}(x) \leq 0, \tag{3.10}$$

$$\mathbf{h}(x) = 0, \tag{3.11}$$

if $f$ and $\mathbf{g}$ are a convex functions, $x$ belongs to a convex set, and $h$ is an affine function. A similar approach may be taken for maximization over concave functions.

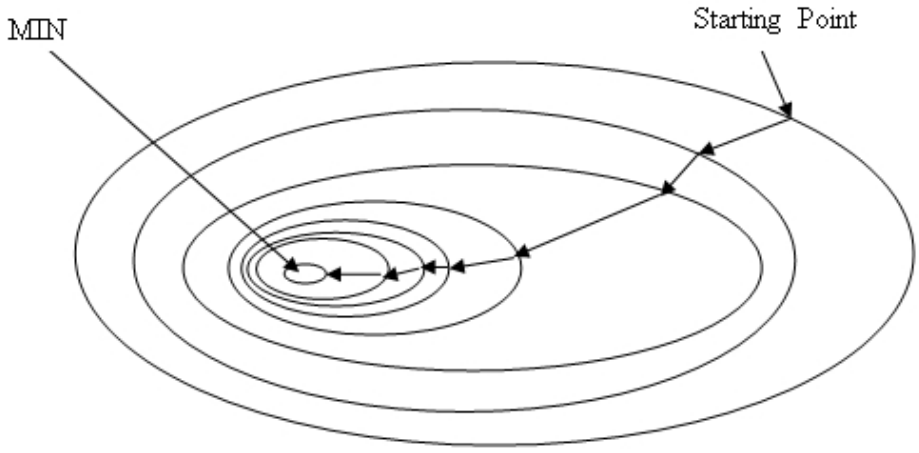



Figure 3.3: The iterative process to find the minimum value of a convex hull [3].

**Data**: $f$ - objective function, $g$ - inequality constraints, $h$ - equality constraints, $x_0$ - starting point, $E$ - stopping criteria
**Result**: min value of $f$

```
1  x := x0;
2  while all e ∈ E is false do
3      if x is unfeasible then
4          Δx := the steepest descent of g and h to feasible region;
5      else
6          Δx := the steepest decent of f;
7          Choose a step size t > 0;
8      end
9      x := x + tΔx.
10 end
```

**Algorithm 1:** Gradient search algorithm for Convex Optimization [3].

*Convex Optimization Algorithms*

Several algorithms exist to solve the general convex optimization problem including interior-point method [30], cutting-plane methods [31], and active-set [32], however a competitively viable and intuitive method is based on gradient search. Gradient search methods are a relatively simple algorithm as outlined in Algorithm 1. The stopping criteria is defined by the user. Typically this is a collection of requirements including

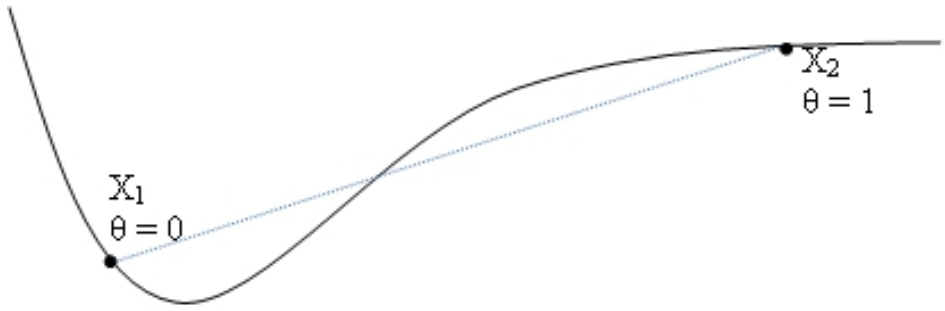



Figure 3.4: A quasi-convex function [3].

tolerances to the object minimizations, step size, iteration numbers, and so on. This process is shown in Figure 3.3. For a proof of convergence for this method please see [3]. It should be noted that there are no polynomial time bounded algorithms for convex optimization.

## 3.4 Quasiconvex Optimization

A weaker definition of convex functions is quasi-convex functions. A function, $f : X \to \mathbf{R}$ defined on a convex set $X$, is said to be quasi-convex if given any two points, $x_1, x_2 \in X$ and any $\theta \in [0,1]$.

$$f(\theta x_1 + (1-\theta)x_2) \leq max(f(x_1), f(x_2)) \tag{3.12}$$

An example of a quasi-convex function is given in Figure 3.4. Simialarly, quasiconcave functions may be defined as a function, $f : X \to \mathbf{R}$ defined on a convex set $X$, is said to be quasi-concave if given any two points, $x_1, x_2 \in X$ and any $\theta \in [0,1]$.

$$f(\theta x_1 + (1-\theta)x_2) \geq min(f(x_1), f(x_2)) \tag{3.13}$$

One should note, however, that these definition still allows the gradient search method described in 3.3 to be used since the function will always have a unique minimum (or maximum if the function is quasiconcave). A useful observation is that any monotonic function (either increasing or decreasing) is quasi-linear. A function is quasi-linear if

it is quasi-convex and quasi-concave. We will now show that any monotonic function, $f(t)$, is quasiconvex if the **dom** $f$ is also convex. If a function is monotonically increasing it is given that (3.14) is true and that $\nabla f(x)$ is positive. It follows that (3.15) must also be true which is the first-order condition for quasiconvexity [3].

$$f(x) \leq f(y) \Rightarrow x \leq y \text{ s.t. } x, y \in \textbf{dom } f \tag{3.14}$$

$$\Rightarrow \nabla f(x)^T (y - x) \leq 0 \tag{3.15}$$

This result is critical to the validity of our solution in Chapter 5 since many of our functions are monotonic (cover, power, battery).

## 3.5 Linear Programming

The final subcategory of optimization, introduced in this paper is linear programming. Linear programming is a special case of convex optimization in which all functions are linear in nature. We may therefore describe a linear programming problem as

$$\min_x \mathbf{F} \cdot \mathbf{x} \tag{3.16}$$

$$s.t.\ \mathbf{G} \cdot \mathbf{x} \leq \mathbf{b}, \tag{3.17}$$

$$\mathbf{H} \cdot \mathbf{x} = \mathbf{b_{eq}}, \tag{3.18}$$

One of the original methods to solve linear programming problems is the Simplex method [33]. Since then, highly efficient algorithms have been established such as Yinyu Ye's method [34], which has runtime complexity of $O(n^3)$, where $n$ is the number of variables. It should be noted that linear programs are far easier to solve than the general convex optimization problem and can be applied to many graph problems such as routing table construction.

Chapter 4

# QUALITY OF SENSOR COVER PROBLEM

In this chapter, the quality of cover for solar powered sensor networks is introduced. Given a network region $R$ (see Figure 4.1), with $M$ targets at known locations, $N$ sensors with adjustable sensing radius $r$ and solar profile $P_n^{sol}(t)$, the quality of cover problem aims to find radii schedule such that the minimum number of targets covered at any point during the operating period is maximized. This objective will be referred to as the Quality of Cover or QoC for short. As discussed in the previous chapter, the size of the problem grows exponentially with $N$ even for a discrete number of sensing ranges. We present the formal definition of the adjustable range quality of cover problem as used in this work. Next we define the cover function as well as examine how to approximate

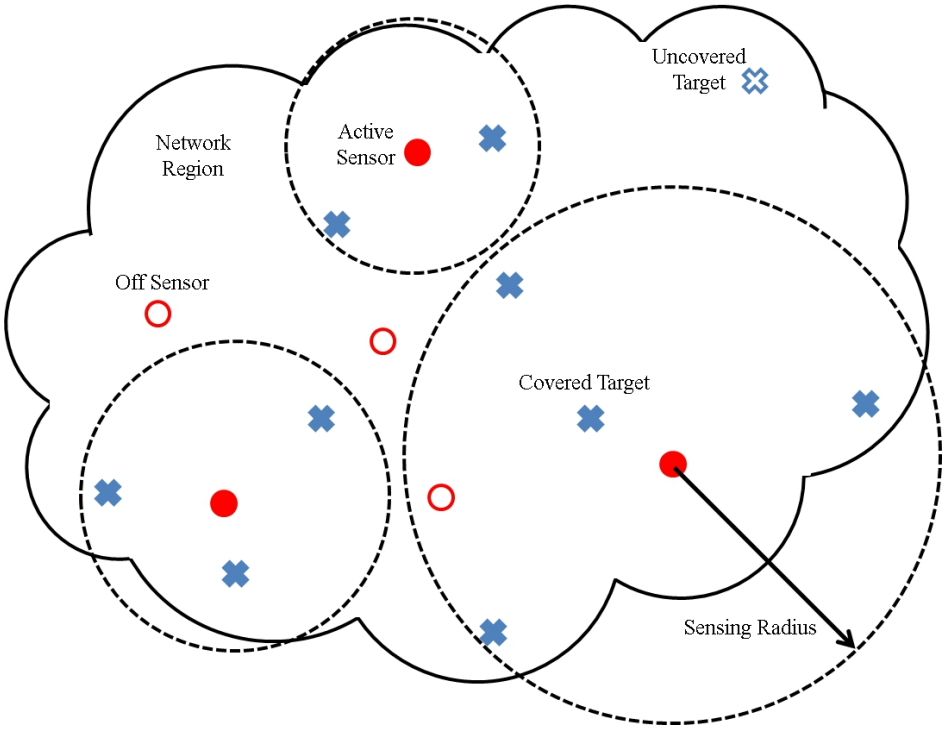



Figure 4.1: A Sensor network.

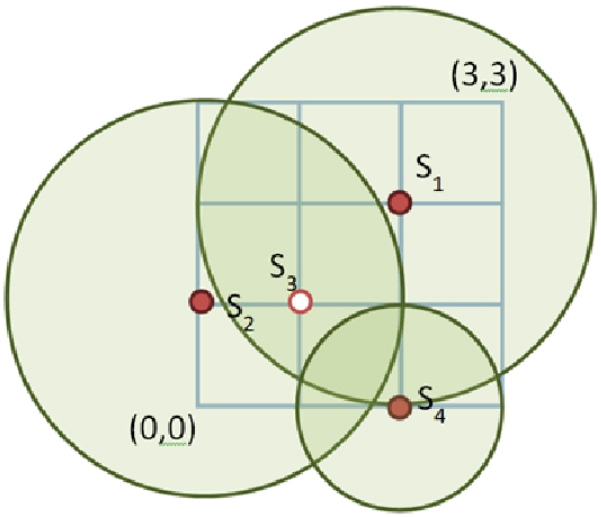



Figure 4.2: An acceptable 100% cover. Each grid point is a target.

the cover function as a convex function. Finally we make note of the importance of sensor node connectivity to the target cover problem.

## 4.1 Operational Range Assignment Quality of Cover Problem

The operational range assignment problem is defined as follows: given a set of sensor nodes and a set of targets, find a subset of sensor nodes and their corresponding sensing radii, which maximizes the quality of coverage (QoC), for the entire operational time of the sensors. The QoC is defined as the minimum cover $\zeta^{min}$ achievable at any point in time. In the example shown in Figure 4.2, a valid cover for one time instance is found with the sensor ranges $S_1 = 2$, $S_2 = 2$, $S_3 = 0$ and $S_4 = 1$, assuming $\zeta^{min} = 16$. The problem description demands such covers be computed for entire operational time, while maximizing $\zeta^{min}$.

Assuming connectivity costs are negligible, the corresponding formulation is given by:

$$\max_{\mathbf{r}(t)} \quad \min_{0 \le t \le 24 \text{ hrs}} \quad \zeta(\mathbf{r}(t)) \tag{4.1}$$

$$s.t. \quad \mathbf{B}(\mathbf{r},t) = \mathbf{B}(0) + \int_0^t (P^{sol}(z) - \mathbf{P}^{sen}(\mathbf{r}(z)))\, dz \tag{4.2}$$

$$\mathbf{B}(t) \geq \mathbf{B^{min}},\ \forall t \in [0, 24 \text{ hours}] \tag{4.3}$$

$$P_n^{sen}(r_n) = \beta r_n^{\alpha} + \mu,\ \forall n \in S. \tag{4.4}$$

In the above formulation, the objective (4.1) is to maximize the minimum level of coverage experienced at any time during a day. The constraint (4.3) prevents the battery of any sensor from reducing below a specified threshold (defined by the network designer), while (4.2) describes the state transition model of the battery energy. A linear battery model is assumed with no energy loss for simplicity and to ensure convexity in the state transition model. The variable of optimization is the vector of time varying sensor radii $\mathbf{r}(t)$. The above formulation falls under the realm of nonlinear optimal control problems [2]. Note that the sensor radii are assumed to be continuous, but in practice, the radii are discrete. Hence, the radii obtained from the solution have to be discretized corresponding to the discrete set of ranges.

Nonlinear optimal control problems of the above type are typically hard to solve, computationally expensive problems. The most common solution techniques are the dynamic programming (DP) and the Hamilton-Jacobi-Bellman methods [2]. To solve this problem using dynamic programming, the entire duration of execution (the 24 hour period) should be partitioned into $K$ time intervals to approximate the continuous nature of the problem. Furthermore, discretization of the states and controls are needed. Let $Q_b$ denote the number of states for battery energy and $Q_r$ be the number of discrete states for sensor radii. The run time complexity of the DP solution would be

$O(K(Q_b)^N(Q_r)^N)$ for an $N$ sensor network, since the solution requires examining all possible controls at every possible state. For this reason fast, near optimal solutions are needed.

## 4.2 Connectivity

**Definition 4.2.1.** *A sensor $n_1$ and sensor $n_2$ have a link if $n_1$ has enough energy supply to transmit all required data for a distance $d_{n_1 n_2}$ to $n_2$, and $n_2$ has enough energy supply to receive all required data. These energy demands are dictated by power models such as Friis transmission equation (2.1) and the per message transmission times.*

**Definition 4.2.2.** *A sensor $n_1$ has connectivity with another sensor $n_2$ only if there exists an ordered subset of sensors, $p$ such that elements $s_i$ and $s_{i+1}$ have a link and $p_0 = n_1,\ p_e = n_2$, where $p_e$ is the last element of $p$.*

In the prior sections, communication costs were assumed negligible; however, in certain applications, this may not be the case. For example long range communication devices require exuberant amounts of power to transmit data.

In the centralized version of the target cover problem previously described, all sensed data must be returned to a centralized base station. Combined with potentially large communication energy costs, these reasons makes network connectivity a critical requirement for the cover problem which should not be overlooked.

When considering connectivity, one must first know the sinks and the sources of data. For the target cover problem, the sources of data are the sensors which are actively monitoring targets, while the sinks are the base stations which manage and react to the sensed data. We are therefore able to define connectivity if the following is known:

1. The energy to transmit and receive data across links must be known. This is established by power models such as Friis equation and the transmission times. We represents these values in the $N \times N$ matrices, $\mathbf{E^{TX}}$ and $\mathbf{E^{RX}}$ respectively.

2. Given a vector of radii, the associated cover model, and the design parameters of messages size, we are able to find $\chi$ which represents the per message routing table for the network.

The second point is of most importance for defining the connectivity problem. Knowing the vector of radii provides several key pieces of information described below.

1. The QoC it generates for the current time, k.

2. The sources of data which must be delivered to the base station. This set of sensors is referred to as $\mathbf{S^{on}}(k)$.

3. The amount of data from each source (a design parameter). Let this be a vector of length N called D.

4. Partial battery levels at time $t$ after solar and sensing powers are factored, $\mathbf{B}'(t) = \mathbf{B}(t) + t\mathbf{P^{sol}}(\mathbf{t}) - t\mathbf{P^{sen}}(t, \mathbf{r}(t))$.

These pieces of information are sufficient to determine if connectivity can be established.

We are now able to define connectivity in sensor networks as any $\chi$ which that satisfies the conditions,

$$\sum_{j=1}^{N} \chi_{ji}(k) - \sum_{j=1}^{N} \chi_{ij}(k) = 0, \ \forall i \in S - \{S^{on} \cup S^{base}\}, \tag{4.5}$$

$$\sum_{j=1}^{N} \chi_{ji}(k) - \sum_{j=1}^{N} \chi_{ij}(k) = -D_i, \ \forall i \in S^{on}, \tag{4.6}$$

$$\sum_{j=1}^{N} \chi_{ji}(k) - \sum_{j=1}^{N} \chi_{ij}(k) = D_i, \ \forall i \in S^{base}, \tag{4.7}$$

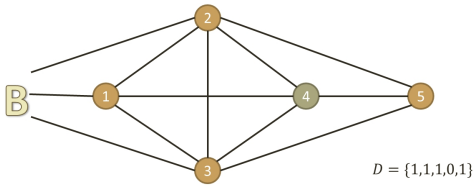
$$E_n^{com} = \sum_{i=1}^{N} E_{ni}^{TX} \chi_{ni} + \sum_{i=1}^{N} E_{in}^{RX} \chi_{in}, \tag{4.8}$$

$$\mathbf{B}(t+t_d) = \mathbf{B}'(k) - \mathbf{E}^{\mathbf{COM}}(k), \tag{4.9}$$

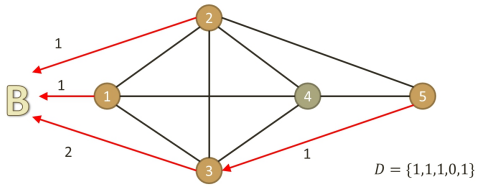
$$\mathbf{B}(t+t_d) \geq \mathbf{B}^{\mathbf{min}} \tag{4.10}$$

where $\chi(k)$ is the per message routing table for the network at time $k$ and $\mathbf{E}^{\mathbf{COM}}(t)$ is the total energy expended from communications at time $t$. We define the time required to transmit these messages from source to sink as $t_d$. Equations (4.5)–(4.7) are standard flow conservation constraints ensuring all data from sources reach the base stations (the sinks), while (4.8)–(4.10) are the state transition requirements. Care must be taken to force any elements of $\chi$ to assume the value 0 if it does not correspond to an actually link in the network. Since these conditions ensure all data is delivered to the base station and no sensor exerts more energy than it is capable of outputting, we can conclude that $\chi$ is a valid routing table and connectivity is achieved. It should be noted, that in the above formulation, messages may be split and the overhead to do so must be negligible; otherwise, the $\chi$ must consist of integer values only.

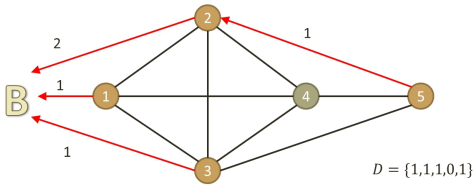
Consider the example shown in Figure 4.3 (a). A valid routing scheme is shown in Figure 4.3 (b) where the elements of $\chi$ are presented above the links. It is clear that all sources of data from $\mathbf{S^{on}}$ are routed to the base station while the residual battery lives



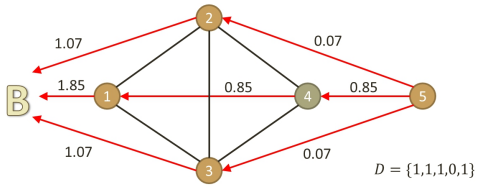

(a) A sensor network with partial battery levels, links, and given sources/sinks.




(b) An example of a valid connectivity.




(c) An example of an invalid connectivity.




(d) An example of an (in)valid connectivity.

Figure 4.3: An example of finding connectivity. The base station (B) is given as the first entry in each of the matrices.

are kept above zero unlike in Figure 4.3 (c). Furthermore, no messages are fragmented between links such as in Figure 4.3 (d) which may cause an invalid choice of $\chi$.

## 4.3 Summary

For completion, the full mathematical problem formulation is given below for the Quality of Cover Problem with Connectivity Constraints,

$$\max_{\mathbf{r}(t)} \quad \min_{0\leq t\leq 24 \text{ hrs}} \quad \zeta(\mathbf{r}(t)) \tag{4.11}$$

$$s.t. \quad \mathbf{B}(\mathbf{r},t) = \mathbf{B}(0) + \int_0^t \left(P^{sol}(z) - \mathbf{P}^{sen}(\mathbf{r}(z))\right) dz - \mathbf{E}^{\mathbf{COM}}(z) \tag{4.12}$$

$$\mathbf{B}(t) \geq \mathbf{B}^{\mathbf{min}}, \ \forall t \in [0, 24 \text{ hours}], \tag{4.13}$$

$$P_n^{sen}(r_n) = \beta r_n^{\alpha} + \mu, \ \forall n \in S, \tag{4.14}$$

$$\sum_{j=1}^{N} \chi_{ji}(t) - \sum_{j=1}^{N} \chi_{ij}(t) = 0, \ \forall i \in S - \{S^{on} \cup S^{base}\}, \tag{4.15}$$

$$\sum_{j=1}^{N} \chi_{ji}(t) - \sum_{j=1}^{N} \chi_{ij}(t) = -D_i, \ \forall i \in S^{on}, \tag{4.16}$$

$$\sum_{j=1}^{N} \chi_{ji}(t) - \sum_{j=1}^{N} \chi_{ij}(t) = D_i, \ \forall i \in S^{base}, \tag{4.17}$$

$$E_n^{com} = \sum_{i=1}^{N} E_{ni}^{TX} \chi_{ni} + \sum_{i=1}^{N} E_{in}^{RX} \chi_{in}. \tag{4.18}$$

# Chapter 5

# CENTRALIZED QUASICONVEX COVER ALGORITHM

In this chapter, we present our quasi-convex optimization-based solution to the quality of cover problem defined in Chapter 4. We will begin by making a few key observations regarding the problem formulation which will serve as a foundation for our solution. Following this we provide the reader with our approximation for the cover model. We then present our algorithm for solving cover and how we are able to maintain an associated connectivity.

## 5.1 Solution Overview

We first make few observations regarding the problem formulation (4.11)–(4.18):

1. The objective function in (4.1) for a fixed time is a discrete quasilinear function since it is monotonically increasing (weak form of convexity, also a quasiconvex function).

2. The conditions (4.12)–(4.18) can be easily shown to be convex since all equations are either linear or monotonic.

3. The connectivity constraints (4.15)–(4.17) are clearly linear. Furthermore, the control variable $\chi$ may be determined if $\mathbf{r}$ is given. This is due to the fact all conditions relating to $\mathbf{r}$ do not effect $\chi$ except the state transition model in which $\chi$ and $\mathbf{r}$ effect the state in a mutually exclusive manner. For this reason, our determination of $\mathbf{r}$ may be viewed as the process of guessing an $\mathbf{r}$, determining an $\chi$, and checking conditional constraints without any loss of correctness to the optimal $\mathbf{r}$ and $\chi$ if the optimal $\mathbf{r}$ was guessed. This process allows us to alleviate

some of the time complexity of convex optimization by introducing significantly faster LP methods into our solution.

4. The resulting optimized overall minimum cover $\zeta$ has to be a constant over all time instants. This is because, the goal is to maximize the minimum cover over the entire duration of operation. There is no added benefit of having a higher cover over the minimum cover for any duration of time, as it does not help in maximizing the objective.

It is the last observation that is key in the transformation of the formulation. Since $\zeta$ is lower bounded by a constant (say $\zeta^{min}$), we can now solve the formulation (4.11)–(4.18) as a quasiconvex optimization problem for a specified time to achieve $\zeta^{min}$ coverage. This is repeated for all $K$ intervals to ascertain if $\zeta^{min}$ coverage is attainable. If not, $\zeta^{min}$ is lowered, or if yes, $\zeta^{min}$ is increased to determine the next maximum $\zeta^{min}$. This search process of determining optimal $\zeta^{min}$ can be done through a binary search technique. Note that in the above mentioned convex optimization problem, there is no real objective, as the objective is a constant $\zeta^{min}$. This gives rise to multiple solutions. In order to avoid this, we attempt to provide the network with the greatest chance of reaching $\zeta^{min}$ by selecting the objective function

$$(1/N)\sum_{n\in N} B_n(k+1) + \min_{n\in N} B_n(k+1) \tag{5.1}$$

and maximizing over it. This objective states that we wish to maximize the tradeoff between average battery life in the network, and the minimum battery life in the network. This objective can be rationalized by considering that the weakest node in the network is most likely to cause a failure, thus we should remove some of its burden in hopes that it may recharge its energy. The average battery life portion of the objective ensures all nodes with battery greater than the weakest node cover the targets in an efficient

manner. Algorithm 2 outlines the problem transformation discussed in this section and as Figure 5.2 illustrates.

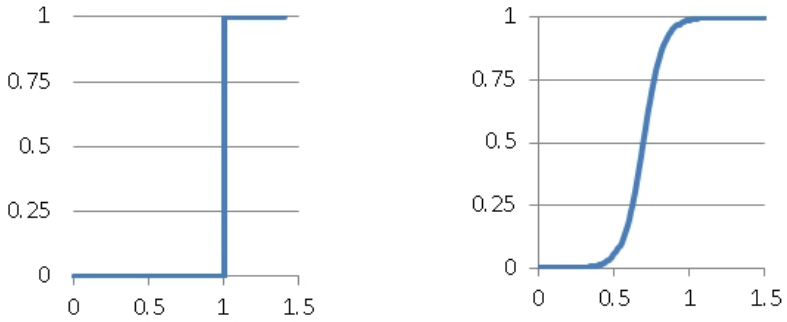
**Input**: $\mathbf{B}(0), P_{sol}, S, N, T, M$
**Output**: optimal radii schedule $\mathbf{r}^*$

1 **begin**
2  $max = M$; $min = \zeta^{min,*} = 0$;
3  **while** $min \leq max$ **do**
4  *failed* = false;
5  $\zeta^{min} = \lfloor \frac{max-min}{2} \rfloor$;
6  **for** $k$ = *1 to K* **do**
7  Solve the quasiconvex optimization problem to find a minimum energy cover s.t. $\zeta(\mathbf{r}(k)) \geq \zeta^{min}$ and $\mathbf{B}(k) \geq \mathbf{B^{min}}$ as described in Sections 5.1 and 5.2;
8  **if** *No cover found* **then**
9  *failed* = true; break;
10  **end**
11  Update $\mathbf{B}(k)$;
12  **end**
13  **if** *failed* **then**
14  $max = \zeta^{min} - 1$;
15  **else**
16  $min = \zeta^{min} + 1$;
17  **if** $\zeta^{min} > \zeta^{min,*}$ **then**
18  $\zeta^{min,*} = \zeta^{min}$; $\mathbf{r}^* = \mathbf{r}$;
19  **end**
20  **end**
21  **end**
22 **end**

**Algorithm 2:** Solution outline to the operational range assignment problem without connectivity

## 5.2 Finding an Optimal Cover

In this section, we discuss the details of the quasiconvex optimization formulation for a $k^{th}$ interval, ($k \in K$), mentioned in the previous section (Step 7 of Algorithm 2). But first we must address the discrete nature of the cover function by creating a continous approximation. This is necessary as the gradient-based solution techniques for convex




(a) Heaviside Step function (b) Logistics function

Figure 5.1: Comparison of the Heaviside Step function to the Logistics function.

programming require the objective and all constraints to be differentiable with few zero valued gradients. Towards this, we approximate the discrete jumps (considered as a *Heaviside step*) in the cover function (see Figure 5.1 (a)) with a *Logistics* function as shown in Figure 5.1 (b). The corresponding equations are:

$$\zeta(\mathbf{r}(k)) = \sum_{m \in T} \max_{n \in N} (L(r_n/d_{nm})), \tag{5.2}$$

$$L(x) = \frac{1}{1+e^{-c \cdot x + \delta}}, \ \forall x \in (0,1), \tag{5.3}$$

$$\delta = \ln(1/\varepsilon - 1) + c. \tag{5.4}$$

$L$ is the logistics function. Since $x$ is bounded in (5.3), $r_n$ in (5.2) is normalized by $d_{nm}$. $c$ and $\varepsilon$ are quality factors for controlling how closely the logistics function resembles the Heaviside step function by adjusting the slope and the horizontal positions of the function. Increasing these values provides a better approximation, but at the cost of increased solution time of the optimizer. We see that the cover function is now differentiable while maintaining its monotonic nature, hence it is now a quasi-convex function according to section 3.4.

With this, the quasiconvex formulation for a $k^{th}$ interval is given by:

$$\max_{\mathbf{r}(k)} \quad (1/N)\sum_{n\in N} Bn(k+1) + \min_{n\in N} B_n(k+1) \tag{5.5}$$

$$s.t. \quad \mathbf{B}(k) = \mathbf{B}(k-1) + \int_{k-1}^{k} (P^{sol}(z) - \mathbf{P}^{sen}(\mathbf{r}(z)))\, dz \tag{5.6}$$

$$\mathbf{B}(k) \geq \mathbf{B}^{\mathbf{min}}, \tag{5.7}$$

$$\zeta(\mathbf{r}(k)) = \sum_{1}^{M} \max_{n\in N} (L(r_n/d_{nm})), \tag{5.8}$$

$$\zeta(\mathbf{r}(k)) \geq \zeta^{min}, \tag{5.9}$$

$$L(x) = \frac{1}{1+e^{-c\cdot x+\delta}}, \quad \delta = \ln(1/\varepsilon - 1) + c, \tag{5.10}$$

$$P_n^{sen}(r_n(k)) = \beta r_n^{\alpha}(k) + \mu,\ \forall n \in S. \tag{5.11}$$

Ignoring connectivity for the time being, the objective (5.5) is to minimize the aggregate power spent by the network while ensuring that the weakest nodes are not overburdened. Constraint (5.9) requires that the minimum cover be above a specific level. The rest of the constraints are same as in the formulation (4.12)–(4.14), but for a $k^{th}$ interval. It is easy to show that the above equations are convex, except for (5.8) and (5.10). Logistic functions are monotonic functions. Hence they are also quasiconvex functions [3]. Since `sum`, `max` and `min` are convex functions in this context, (5.8) is a quasiconvex function. This makes the above formulation a quasiconvex optimization problem and may be solved as such. It should be noted that we have not yet introduced the connectivity constraints. This will be addressed in the next section.

## 5.3 Maintaining Connectivity

This section will now introduce the constraint of maintaining connectivity within the network. In order to successfully route all data through the network to a centralized base station we must first define which sensor nodes are sources of data. We define a sensor node to be a source of data for the $k$th interval if it is using a sensing radius

greater than 0; in other words, the set of all data sources for time k is equivalent to $\mathbf{S^{on}}$ at time k.

For this reason we must first know the sensing radii of the network before a valid routing scheme can be found, thus connectivity may be viewed as a subproblem to the target cover problem.

$$\mathbf{S^{on}}(t) = \mathbf{S} \cap \{\mathbf{r}(t) > 0\}. \tag{5.12}$$

Assuming that $S^{on}$ is known, the process of routing data from the sources to the sink is equivalent to the min-cost flow problem with any given objective function [33]. We will maintain the objective function (5.1) used for finding the cover for the same reasons given in Section 5.1 and because the function is linear. We can now state that a valid connectivity is found as the solution to the linear program

$$\max_{\chi(k)} \quad (1/N) \sum_{n \in N} B_n(k+1) + \min_{n \in N} B_n(k+1) \tag{5.13}$$

$$s.t. \quad \chi_{ij}(k) \geq 0 \ \forall i, j \in S, \forall t \tag{5.14}$$

$$\sum_{j=1}^{N} \chi_{ji}(t) - \sum_{j=1}^{N} \chi_{ij}(t) = 0, \ \forall i \in S - \{S^{on} \cup S^{base}\}, \tag{5.15}$$

$$\sum_{j=1}^{N} \chi_{ji}(t) - \sum_{j=1}^{N} \chi_{ij}(t) = -D_i, \ \forall i \in S^{on}, \tag{5.16}$$

$$\sum_{j=1}^{N} \chi_{ji}(t) - \sum_{j=1}^{N} \chi_{ij}(t) = D_i, \ \forall i \in S^{base}, \tag{5.17}$$

$$E_n^{com} = \sum_{i=1}^{N} E_{ni}^{TX} \chi_{ni} + \sum_{i=1}^{N} E_{in}^{RX} \chi_{in}, \tag{5.18}$$

$$\mathbf{B}(k) = \mathbf{B}(k-1) + \int_{k-1}^{k} (P^{sol}(z) - \mathbf{P}^{sen}(\mathbf{r}(z))), \ dz - \mathbf{E^{COM}}(k) \tag{5.19}$$

$$\mathbf{B}(t) \geq \mathbf{B^{min}}, \ \forall t. \tag{5.20}$$

where $D_i$ is the amount of data (messages) source $i$ must send to the base station and $\chi$ is the $N \times N$ matrix corresponding to the routing paths we are trying to find (note that $\chi_{ij}$ is the amount of data sent from node i to node j). Constraints (5.15–5.17) simply insure that all transmitted data is delivered to the base station. This is essentially a modified min cost flow problem where the node capacities are dictated by the residual battery levels as seen in (5.20).

To solve such problems, any linear program solver will work. We have discussed in Chapter 3 that a $O(n^3)$ LP solver exists and therefore could solve this problem in $O(N^6)$ since there are at most $N^2$ possible edges in the network. Alternatively, mean cycle canceling algorithms exist and can provide better runtime complexity [33].

As discussed in Chapter 4, $\chi$ may have to be integer valued if the overhead of message splitting/merging is not negligible. In this case there is no guarantee the LP to produce integer results since total unimodularity is not always possible. In this case, IP methods are required such as cutting-plane and sequential fixing [33].

## 5.4 The Complete Algorithm

We now present the complete algorithm to solve the operational range assignment problem of solar powered WSN's while insuring connectivity. Figure 5.2 shows the flow of the algorithm and will be referenced for this section.

As rationalized in the prior sections, the problem time component of the problem is discretized and thus $\zeta(t)$ now appears as a constraint, rather than the objective. For this reason we must perform a search over the possible $\zeta^{min}$ to find the one which is optimal. We accomplish this by performing binary search as seen in Boxes 1, 8, 11, and 12 of Figure 5.2. Box 1 simply gives the starting point for the binary search while Boxes 8 and 11 update the value of $\zeta^{min}$ to check depending on whether a valid

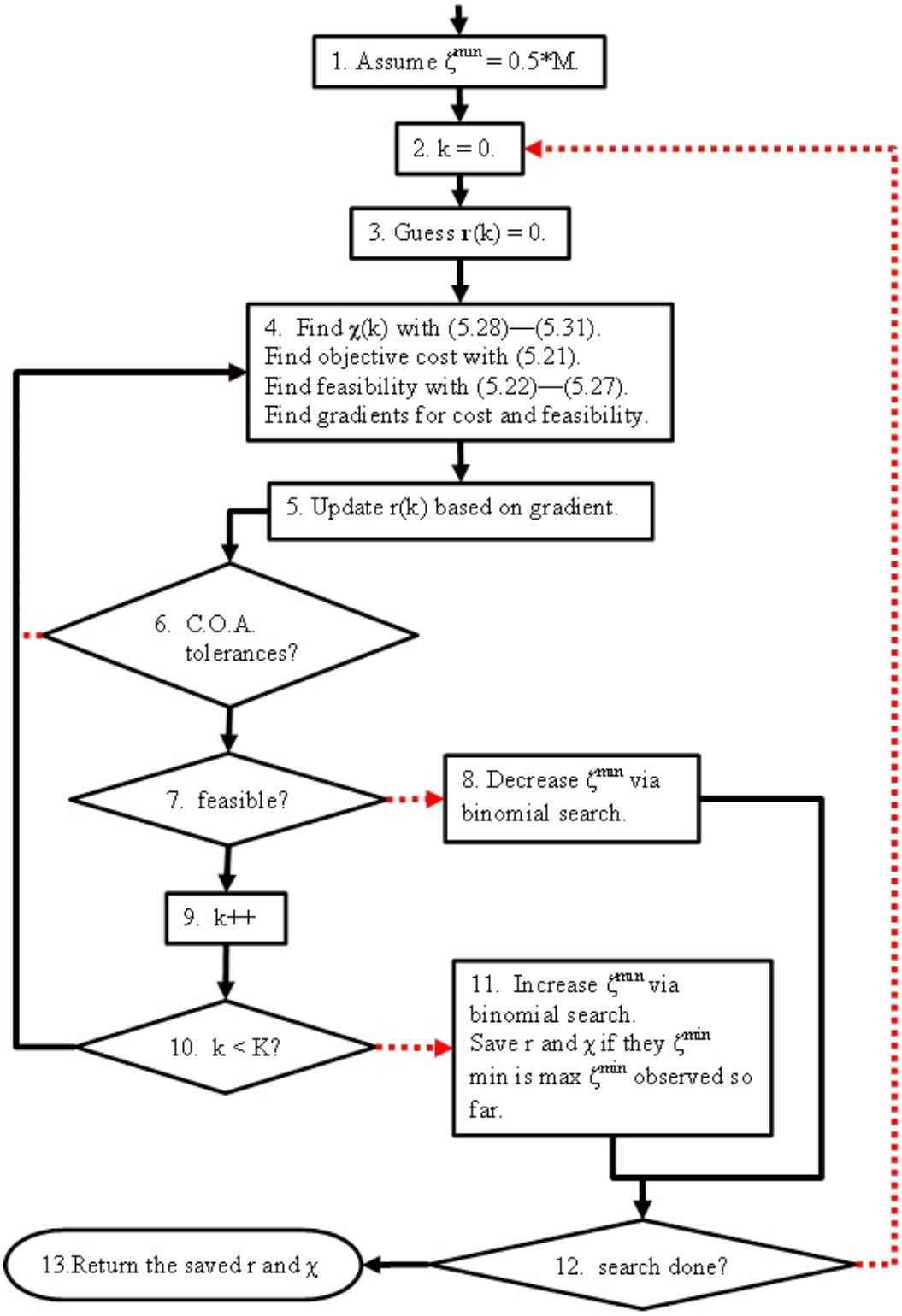



Figure 5.2: The proposed algorithm to maximize QoC.

cover and connectivity schedule was found for the current assumed $\zeta^{min}$. The algorithm concludes when the binary search completes as shown in Box 12.

The convex optimization process for each time interval $k$ of a given $\zeta^{min}$ occurs in Boxes 3 through 7. For completeness the entire problem formulation is given below.

$$\min_{\mathbf{r}(k)} \quad (1/N)\sum_{n\in N} Bn(k+1) + \min_{n\in N} B_n(k+1) \tag{5.21}$$

$$s.t. \quad \mathbf{B}(k) = \mathbf{B}(k-1) + \int_{k-1}^{k} (P^{sol}(z)) - \mathbf{P}^{sen}(\mathbf{r}(z))\, dz - \mathbf{E}^{\mathbf{COM}}(k) \tag{5.22}$$

$$\mathbf{B}(k) \geq \mathbf{B}^{\mathbf{min}} \tag{5.23}$$

$$\zeta(\mathbf{r}(k)) = \sum_{1}^{M} \max_{n\in N} (L(r_n/d_{nm})) \tag{5.24}$$

$$\zeta(\mathbf{r}(k)) \geq \zeta^{min} \tag{5.25}$$

$$L(x) = \frac{1}{1+e^{-c\cdot x+\delta}}, \quad \delta = \ln(1/\varepsilon - 1) + c \tag{5.26}$$

$$P_n^{sen}(r_n(k)) = \beta r_n^{\alpha}(k) + \mu,\ \forall n \in \mathbf{S}, \tag{5.27}$$

$$\sum_{j=1}^{N} \chi_{ji}(t) - \sum_{j=1}^{N} \chi_{ij}(t) = 0,\ \forall i \in \mathbf{S} - \{\mathbf{S}^{\mathbf{on}} \cup \mathbf{S}^{\mathbf{base}}\}, \tag{5.28}$$

$$\sum_{j=1}^{N} \chi_{ji}(t) - \sum_{j=1}^{N} \chi_{ij}(t) = -D_i,\ \forall i \in \mathbf{S}^{\mathbf{on}}, \tag{5.29}$$

$$\sum_{j=1}^{N} \chi_{ji}(t) - \sum_{j=1}^{N} \chi_{ij}(t) = D_i,\ \forall i \in \mathbf{S}^{\mathbf{base}}, \tag{5.30}$$

$$E_n^{com} = \sum_{i=1}^{N} E_{ni}^{TX} \chi_{ni} + \sum_{i=1}^{N} E_{in}^{RX} \chi_{in}. \tag{5.31}$$

An initial starting point for the convex optimization process is needed and shown in Box 3. In Box 4 of Figure 5.2, given a vector of sensing radii, we must now compute $\mathbf{S}^{\mathbf{on}}$ and determine if a valid $\chi$ exists using the LP presented in Section 5.3. We are then able to calculate the objective function since all energy consumption values are known. Given all of this, we must now determine whether our proposed $\mathbf{r}$ and $\chi$ is valid by making sure that the corresponding cover is greater than $\zeta^{min}$ and that no battery has

been depleted beyond its limits. The last step of Box 4 is to determine the gradient and feasibility gradient to determine our next proposed value of **r** in Box 5. In Box 6, the tolerance conditions must be checked to determine whether to continue with the convex optimization process. These tolerances are from the convex optimization algorithm (C.O.A) which allow the algorithm to stop after a certain number of iterations, or when the rate of change of the objective function diminishes below a threshold. Once the process terminates, Box 7 simply determines whether the found **r** and $\chi$ are valid.

The remaining iterative search steps over the discretized time intervals are given in Boxes 2, 9, and 10. If the convex optimization process ever fails before $k = K$, then we must consider that the assumed $\zeta^{min}$ is impossible to achieve and therefore we stop the iterative search, otherwise we need to start a new iterative search for a new $\zeta^{min}$. When all searching is done, the **r** and $\chi$ corresponding to the max assumed $\zeta^{min}$ is returned in Box 13.

# Chapter 6

# EXPERIMENTAL RESULTS

## 6.1 Simulation Setup

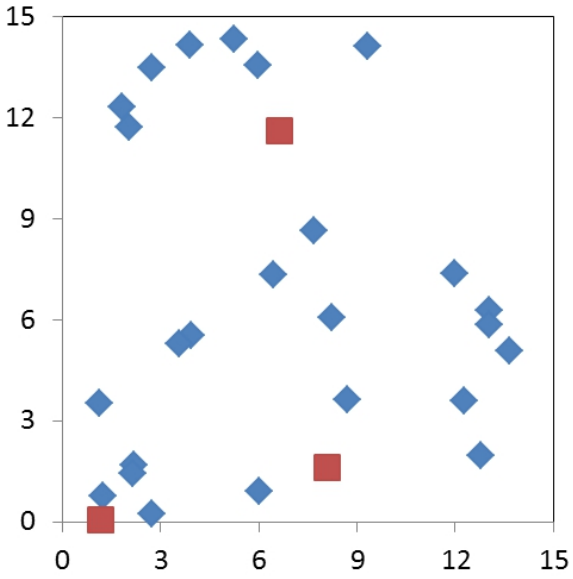
We experimentally verify the proposed solution for the operational range assignment problem by simulating a stationary network with sensor nodes and targets in various location configurations. Common sensor network parameter values are summarized in Table 6.1. Unless otherwise noted, these values are used in all the subsequent experiments. We assume that all sensor nodes are homogeneous and the solar profile depicted in Figure 2.1 is used in the experiments unless noted otherwise. In order to highlight the benefits of our technique, networks were configured in a manner such that a 100% cover is not achievable in trivially small networks ($N < 100$). In most experiments, an omni-directional, active sensor is used. The base station is always located at point $0, 0$.

## 6.2 Time Plots of Sample Scheduling of the Proposed Algorithm

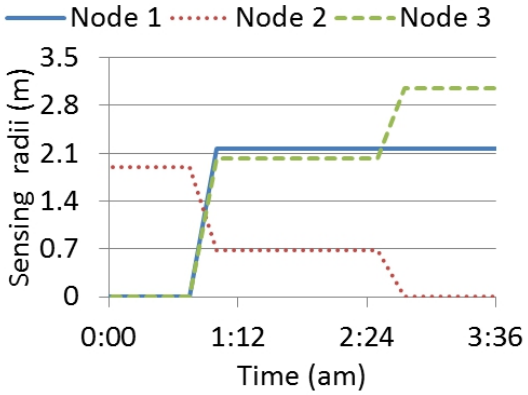
Figure 6.1 shows the plots of scheduling radii of sensors according to the proposed algorithm, the resulting battery charge, and the QoC over a course of 24 hours. The network is configured with 3 sensors and 25 targets arranged in a random pattern as shown in Figure 6.1 (a). The initial battery level for each sensor is set to 11 J. In the interest of clarity, only 3 hours of the radii schedule is shown. We observe that the algorithm switches between various covers with different radii to ensure that the

Table 6.1: Common Network Parameter Values

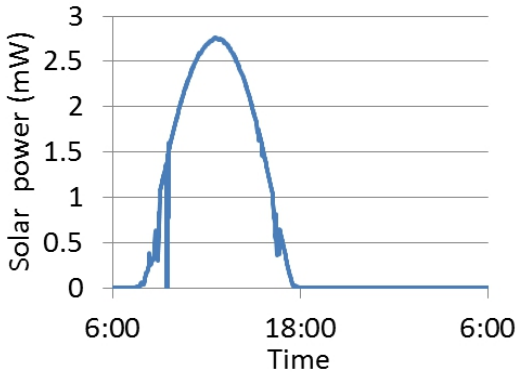
| Param | Value | Param | Value | Param | Value |
|---|---|---|---|---|---|
| Sensor: $r^{max}$ | 10 m | $B^{ini}$ | 0.72 kJ | $B^{max}$ | 4.32 kJ |
| Sensor: $\alpha$ | 4 | Radio: $\alpha$ | 3.14 | Panel Size | 5 cm$^2$ |
| Sensor: $\beta$ | $2.31 \times 10^{-8}$ | Radio: $\beta$ | 0.0002 | Area | 15 m$^2$ |
| Sensor: $\mu$ | 0 | Radio: $\mu$ | 0.003 | | |



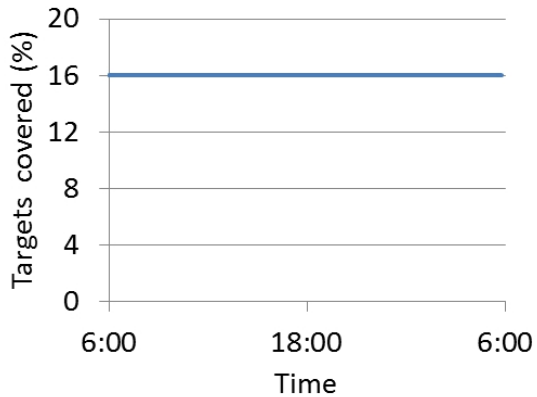

(a) Sensor (squares) and target (diamonds) locations.



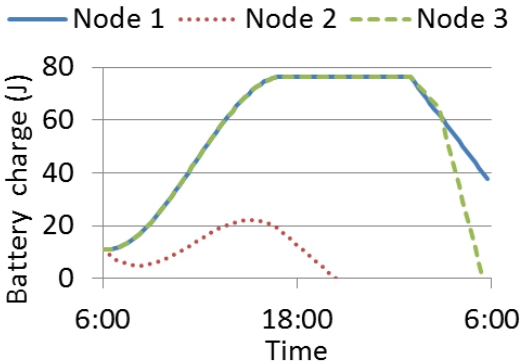

(b) Sensor radii




(c) Solar profile




(d) QoC




(e) Node battery charge

Figure 6.1: Time plots of scheduling for the proposed algorithm.

energy of no battery is reduced to zero, while attaining the maximum possible QoC. This can be observed at time 2:30 am when the battery of Node 2 depletes completely and Node 3 increases its sensing radius to satisfy the minimum QoC. Figure 6.1 (e) demonstrates that the battery primarily charges during hours of sunlight, and deeply discharges at night as expected. The QoC of this schedule is kept constant at 4 targets (16%). Because of the trivially small nature of this network, connectivity was purely a one hop (i.e. all sensors reported directly to the base station).

## 6.3 Run Time Analysis

To verify the practicality of the proposed solution, the run time of the proposed solution is compared with a dynamic programming (DP) solution. Dynamic programming is one of the few known methods to solve optimal control problems; however, it requires all continuous variables (controls, states, time, etc.) to be discretized [2]. Increasing the number of quantized values for each of these variables will increase the accuracy of the result, but will require a longer execution time as explained in Chapter 3. The proposed solution assumes continuous control over sensing ranges. To avoid unfair bias, the proposed algorithm is modified to choose the nearest discrete range from a set of radii used in the DP solution. This is also required for practical implementation of the proposed algorithm. All parameter values were based off of existing sensor node hardware, TI's ez430-RF2500 wireless sensor node system [35] along with commercially available radars. All simulations were run on a single core of a Dell workstation with an 2.93 GHz Intel Core i7 and 8 GB of RAM.

In this experiment, the sensors and targets were equally distributed over the area of operations (see Figure 6.3 (a)). The solar profile used in the experiment is shown in Figure 2.1. The maximum number of sensor nodes was limited to five in this experiment, due to the enormous time complexity of the DP solution. The battery charge was

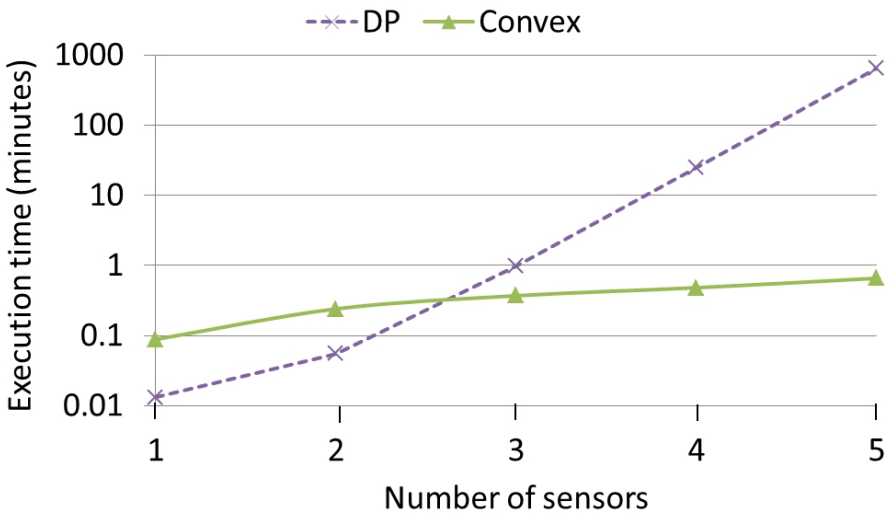
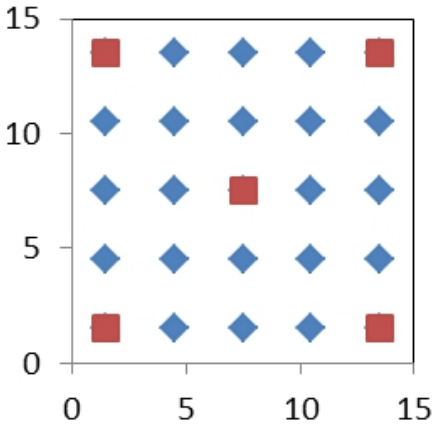


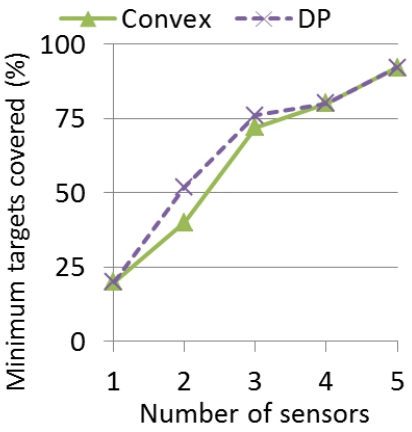

Figure 6.2: Execution time vs. number of sensors.

quantized with 7 states and the number of sensing ranges was kept constant at 6 for the DP solution. Furthermore, connectivity constraints where removed to reduce the time complexity of the DP solution.

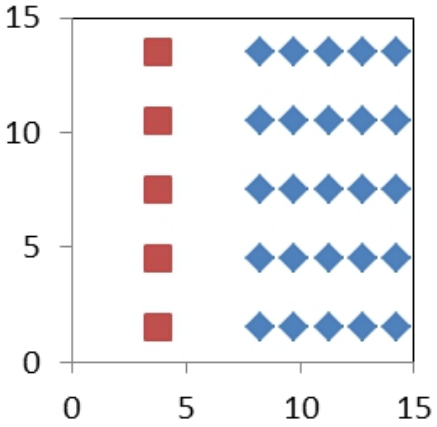
The results of this experiment are shown in Figure 6.2. Note that the scale of the y-axis is logarithmic. We observe that the proposed solution has a large speedup compared with the DP solution as the number of senor nodes increase. Even for 4 sensor nodes, the quasiconvex optimizer finds a solution **60** times faster than the DP procedure. This demonstrates that the proposed algorithm is of greater practical use than the DP solution for large networks. This also aids in design space exploration of large networks in reasonable time as seen from the subsequent experiments. Table 6.2 demonstrates that the proposed solution can handle large networks and produce solutions in reasonable time. Note that, these run times are for a single-core processor. With the help of parallelization, it is possible to reduce the run times greatly.



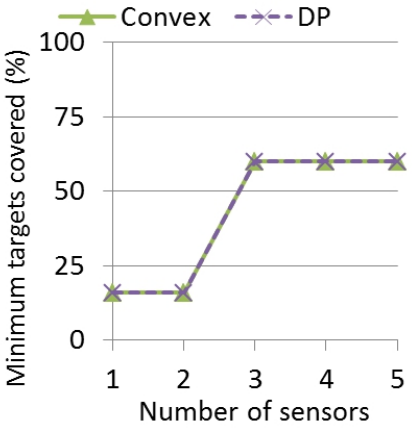

(a) Symmetric sensor-target locations



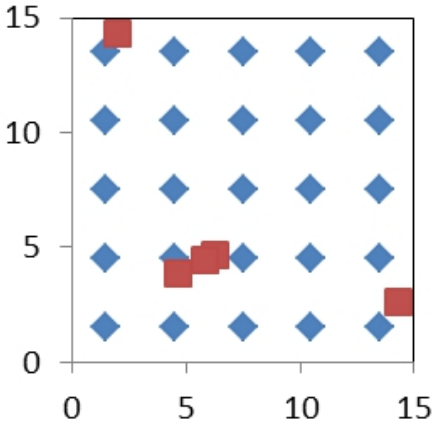

(b) QoC – Symmetric locations



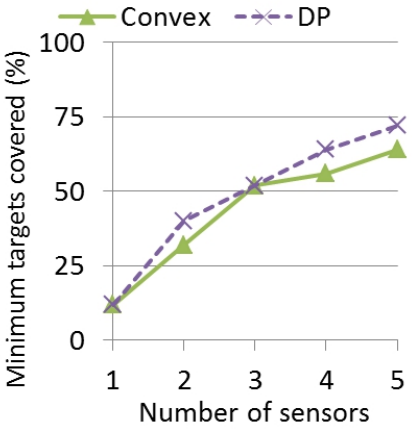

(c) Separated sensor-target locations




(d) QoC – Separated locations




(e) Random sensor locations




(f) QoC – Random locations

Figure 6.3: Various layout scenarios and the resulting QoC.

Table 6.2: Runtime for Convex Solution without Connectivity

| $N$ | 1 | 10 | 20 | 40 | 50 | 100 |
|---|---|---|---|---|---|---|
| Runtime | 3.87 s | 100 s | 309 s | 21.2 min | 33.2 min | 2.12 hr |

## 6.4 Accuracy of the Proposed Solution vs. Dynamic Programming

The overall accuracy of the proposed solution against the naive DP solution outlined in Section 6.3 is examined here. We consider three different sensor-target location configurations for this experiment as shown in Figure 6.3. These configurations represent the diverse scenarios for a sensor network tasked with surveillance duty. Figure 6.3 (a) illustrates the case where all targets and sensors are equally distributed across the operation area for optimal area coverage. Figure 6.3 (c) shows the case where all targets and sensors are equally distributed across the operations area, but the sensors and the targets are separated from one another. This scenario represents the scenario of an enemy territory surveillance. From an energy standpoint, this is one of the worst possible scenarios, since this requires larger sensing radii. Finally, Figure 6.3 (e) shows the case where sensors are randomly distributed. This is the most common scenario in sensor networks tasked with monitoring inaccessible areas. For these experiments, the number of battery states was increased for the DP procedure to increase its accuracy. All other parameters remained the same as in Section 6.3.

The results shown in Figures. 6.3 (b), 6.3 (d) and 6.3 (f) show our proposed algorithm can cope with several possible network scenarios and provide near-optimal results. In the symmetric configuration, our proposed solution tends to provide near identical results to the DP solution as $N$ increases. This is due to the fact that as the $N$ increases, the mean distance from any sensor to target decreases, thus allowing for smaller radii to be used. These smaller radii use less energy and thus the effects of any

non-ideal radii selections are reduced. For the separated configuration, results were identical between the two methodologies. This may be attributed to the fact that there are far fewer valid covers for this configuration as opposed to the symmetric configuration. Finally, the random configuration tends to display varying levels of error at each value of $N$. This is to be expected when the sensor nodes are randomly positioned creating vastly different network topologies. Over the results of all experiments, a peak of **8%** QoC loss was seen.

## 6.5 Impact of Connectivity on QoC and Execution Time

The goal of this experiment is to highlight the importance of considering connectivity and data routing alongside QoC. For this experiment a $21 \times 21$ grid of targets was placed over a 200 $m^2$ area. The number of sensors varied and their locations were randomly chosen. We also increased set the $\alpha$ and $\beta$ variables for data transmission to be equivalent to those of the sensor.

Two routing schemes are considered. The first is a simple one hop routing scheme in which all active sensors simply transmit the data directly to the base station. The second is our version presented in Chapter 5 which uses linear programming integrated into quasiconvex optimization techniques refered to as multi-hop routing. The results are shown in Figure 6.4 and Figure 6.5.

As one would expect from a nonlinear power model, a one-hop routing scheme is inferior to multi-hop routing in regards to energy. This is shown in Figure 6.4 where our proposed solution out performs the naive one-hop method every time in respect to QoC. It should be noted, however, as the sensor density increases, the QoC degradation associated with the single-hop routing scheme diminishes. Furthermore, the impact on QoC is lessen when energy transmission costs become less significant in comparison to the sensor covering costs.

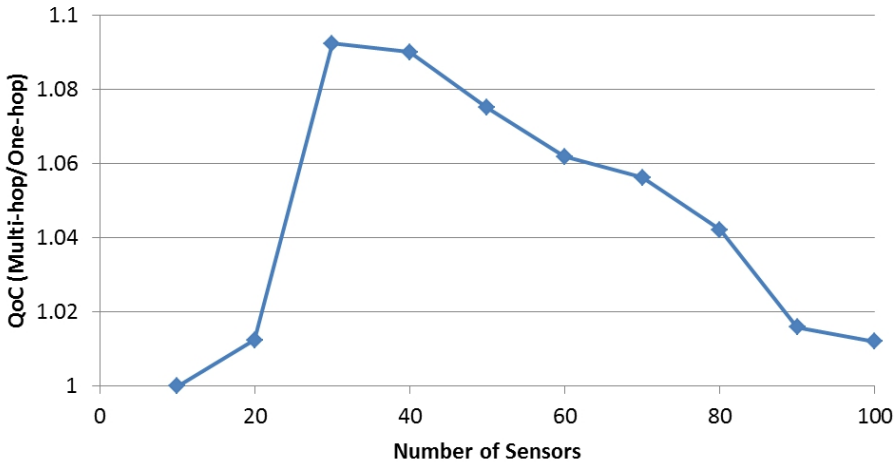



Figure 6.4: Effect of Connectivity on QoC.

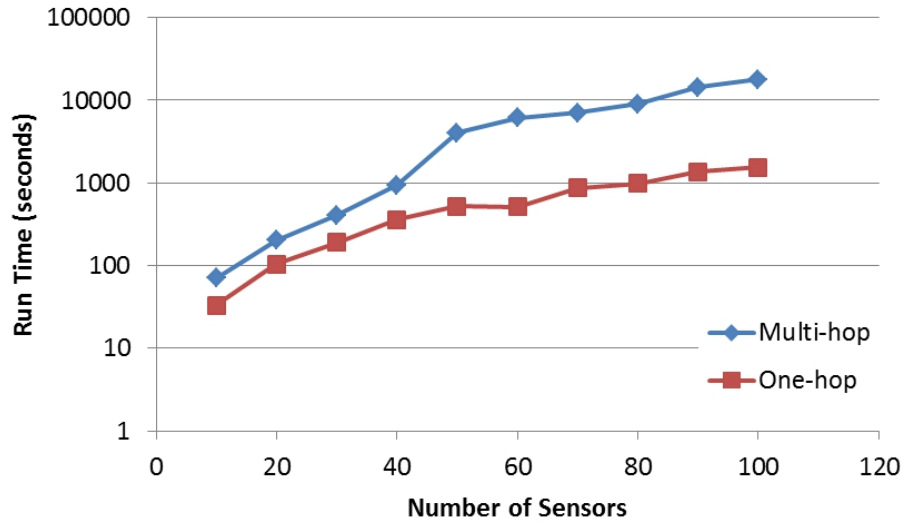



Figure 6.5: Effect of Connectivity on Run Time.

The time analysis is shown in Figure 6.5. Since the LP for connectivity must be solved for every guess of $\mathbf{r}$, it is run a considerable amount of times depending on the convex solver used. For this reason execution time suffers greatly.

Introducing the connectivity constraints to the problem can improve QoC; however, it comes at the cost of increased execution time. As such, designers need to take careful note of the network size, power costs, and node density.

## 6.6 Investigation of Objective Functions

The next experiment quickly examines the objective function chosen for our quasiconvex solution. As mentioned earlier, the objective function in our reformulation of the problem can be considered as a dummy function. As such we may consider other possible objectives. The first objective (6.1) is simply to minimize the energy consumed by the network, or equivalently, maximizing the total residual battery life of the network. The second objective (6.2) is to maximize the minimum residual battery among all nodes in the network. The third objective (6.3) is meant to balance the other two objectives. The same parameters from Section 6.5 are used. Results are shown in Figure 6.6. To help highlight the results only the difference in QoC between the alternative two objective functions and the objective function proposed in Chapter 5 is given.

$$\max_{\mathbf{r}(k),\chi(k)} \quad \sum_{n\in S} B_n(k) \qquad \text{(Sum)} \tag{6.1}$$

$$\max_{\mathbf{r}(k),\chi(k)} \quad \min_{n\in S} B_n(k) \qquad \text{(Min)} \tag{6.2}$$

$$\max_{\mathbf{r}(k),\chi(k)} \quad (1/N)\sum_{n\in S} B_n(k) + \min_{n\in S} B_n(k) \qquad \text{(Balanced)} \tag{6.3}$$

The difficulty of this problem become apparent when observing the results. No one objective function was able to consistently outperform the other two among all values of N; however, our proposed objective function does outperform the other two on average giving validity to our reasoning.

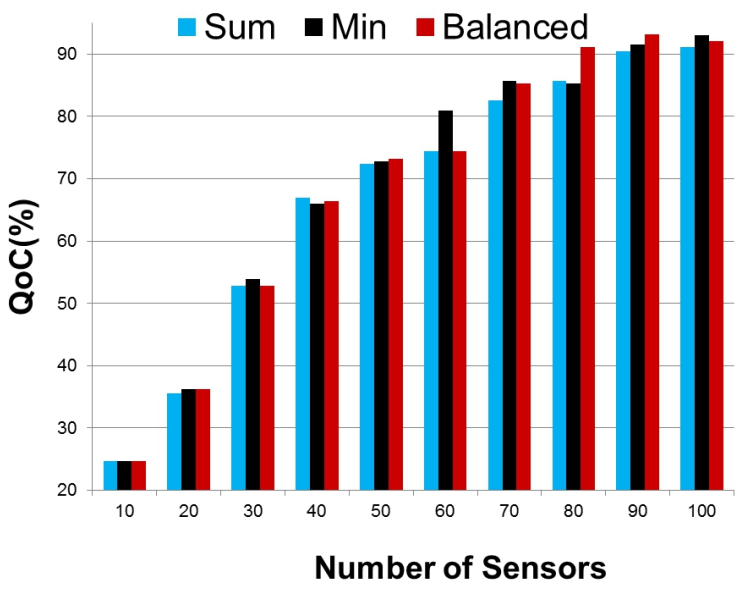



Figure 6.6: Effect of various Objective Functions on QoC

## 6.7 Effect of Number of Sensors on the Network Setup Cost

In this experiment, we study the effect of the number of sensors on the total cost of setup of a sensor network to maintain a specified QoC. The results of this experiment offer network designers with the information to trade-off number of sensors to minimize the initial setup cost of a sensor network. This is also useful to maintain energy neutral operations in a network [36]. Energy neutral operation requires minimum sizing of the battery and the solar panel to reduce the operation cost, while guaranteeing a minimum QoC for a given network. On the other hand, increasing sensor nodes increases the network cost.

For the experiment, a large number of targets (1024) are distributed over a 200 m $\times$ 200 m area in the same fashion as in Section 6.8. The requirement is that 100% of the targets must be covered. To calculate the cost of creation of a sensor network, we assume that each sensor node costs \$20 [37], solar panel costs \$2 per

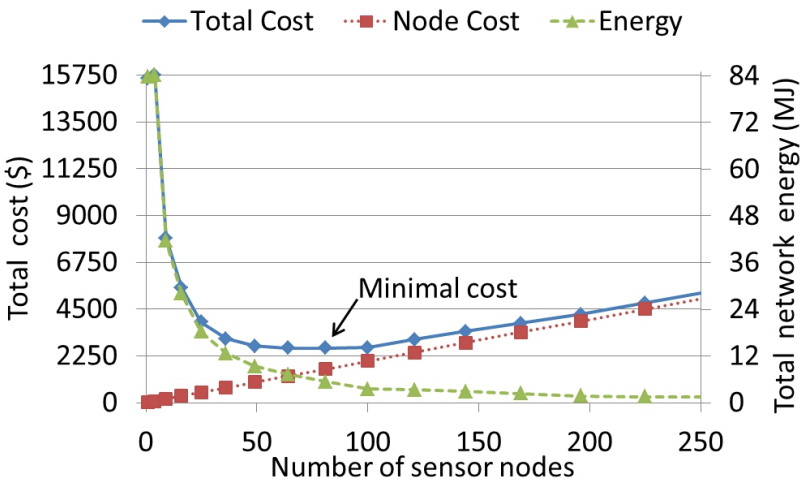



Figure 6.7: Effect of number of sensor nodes on the network setup cost

Watt [38], and batteries cost $0.47 per Watt-hour [39]. With the assumption of 10 hours of sun light per day, the results shown in Figure 6.7 were generated.

Contradicting intuition, increasing the number of sensor nodes does not increase the total network energy. This is because, with more sensors, the radii can be smaller to achieve the same coverage, and the energy cost decreases quadratically with the radius. Since the total network energy decreases, smaller batteries and smaller solar panels are sufficient to maintain the original specified cover, thus decreasing the total network setup cost. However, an increase in the number of sensor nodes adds to the total cost of network setup. Thus, there exists a unique configuration of number of sensor nodes that minimizes the overall cost of network setup. This is shown in Figure 6.7. The plot shows a sharp initial reduction in the network energy, due to the non-linear relation between the network energy and the sensor radii, and the fact that the effective sensor radii decreases quadratically with the increase in number of nodes.

## 6.8 Effect of Sampling Time on QoC

The sampling time is described as the time between two consecutive sensings of an object. In many non-critical applications, targets do not require continuous monitoring and for many sensor nodes continuous monitoring is impossible. For this reason, sampling time offers network designers a unique control variable to extend the lifetime of their networks at the expense of periods of zero cover in traditional battery powered networks. In the context of solar powered networks, the benefit is an increase in QoC. In this experiment, the effect of reducing the sampling time is investigated. Sensors and targets are arranged in a square grid-like pattern, with equal spacing. The number of sensors was kept constant at 100 and the number of targets was kept constant at 900, deployed over a 200 m $\times$ 200 m area. The sampling time was varied between 1% and 100% of the total operation time (the 24 hour period). We also make a slight modification to our definition of cover to be the number of targets covered averaged over the operation time.

Figure 6.8 shows the result. As expected, QoC increases as the sampling time increases. This is because, increasing sampling time allows the sensors to harvest more solar energy, and thus the sensors can afford to use larger radii to enhance the QoC. It is interesting to note the quadratic increase in the QoC with linear increase in the sampling time. The multiple jumps in the plot are due to the discrete nature of the cover (no. of targets) and the network topology.

## 6.9 Effect of Beam Width on QoC

The final design exploration experiment performed observes the effect of varying beam widths on QoC. In this experiment, the beam width was varied between $0^\circ$ and $360^\circ$. The power density was kept constant and maximum power output was capped at 100

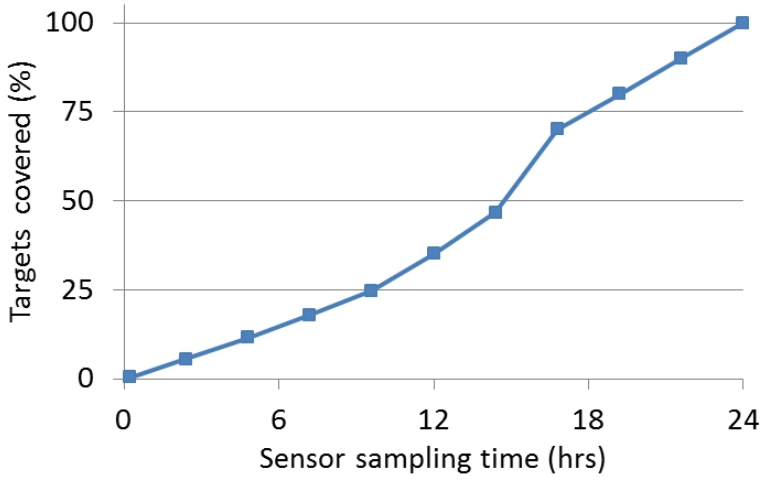



Figure 6.8: Sampling time vs. the number of sensors.

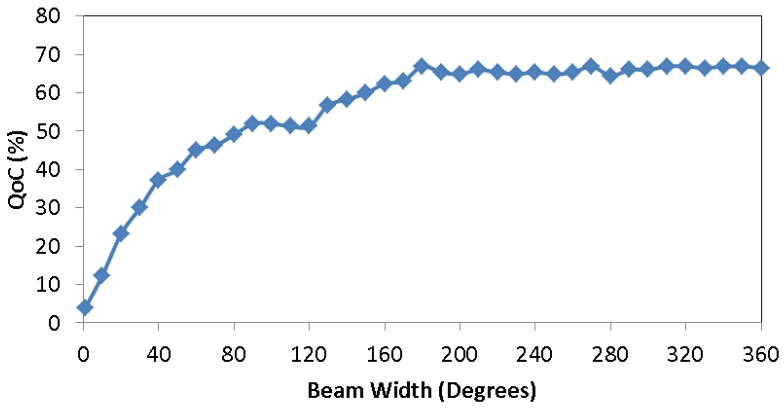



Figure 6.9: Beam Width vs. QoC.

W. Targets were dispersed in a $21x21$ grid across a 200 $m^2$ area. Likewise 25 sensors were equally spaced across this region. All sensor beams faced the same direction. The results are illustrated in Figure 6.9.

Despite the increase to power costs, increasing beam width almost always tends to increase QoC (with few exceptions which may be attributed to the discrete nature of the cover model). That said, the rate at which QoC increases diminishes nonlinearly with beam width until a threshold is reached. In the above case this threshold

is approximately 67% QoC first achieved at 180°. It should be noted that often times, omni-directional sensors are constructed from a ring of narrow-beam sensors. For this reason, design costs could be decreased by creating only semi-directional sensors with no loss to QoC.

# Chapter 7

# CONCLUSION AND OPEN PROBLEMS

## 7.1 Conclusions

There has been a proliferation of energy harvesting sensors in WSN's. With this surge of "green" technology come many additional challenges on optimal scheduling of sensor nodes to maximize the QoC. In this work, a novel, near optimal solution was presented to the scheduling problem of active sensor nodes in the context of the target cover problem which maximizes the minimum attainable QoC. The proposed quasiconvex solution not only considers the cover requirement, but also determines a proficient routing scheme to deliver all sensed data to a centralized base station – something many designers overlook. Our solution is demonstrated to have large speedup compared with the naive DP solution with minimal error in accuracy.

We broadened the usefulness of our solution by demonstrating how it may be applied to various design space explorations of energy harvesting sensor networks. The insights provided by these experiments show that sensor networks may be optimally sized to minimized startup cost. Furthermore, the arbitrary increasing of beam width provides decreasing returns on QoC improvement. The benefits of sampling rates was also explored, revealing a positive trend between average QoC and sampling time but at sporadic rates due to the discrete nature of the cover models.

## 7.2 Open Problems

This work presented several interesting observations into energy harvesting networks; however, there are many possible extensions which may be explored and integrated into the presented framework. Three possible extensions are briefly introduced below.

### *Real Time Distributed Networks*

All presented solutions have been solely centralized in nature. Furthermore, the validity of the solution for a real time implementation is somewhat questionable when considering immense networks (more than several thousand). For this reason, a distributed version of our algorithm would be extremely useful although very challenging to implement. This would allow for more independence in the network and tolerances to outside errors such as environmental damage. Possible first steps to solve such a problem may be from the artificial intelligence domain such as game theory.

### *Unknown Solar Profiles*

Continuing the real time theme introduced in the prior section, considering the randomness of solar profiles could provide interesting additions to the problem. Currently the solution assumes that the solar profile is known or inferred from a large pool of empirical data; however, in reality, solar profiles are highly random. For this reason, accurately predicting solar conditions would be very useful. Furthermore, determining how the network should respond in a timely manner to faulty predictions would be a very interesting investigation. Kalman filters offer possible solutions to such a problem.

### *Mobile Sensors and Targets*

Recently, much interest has been put into mobile networks due to cellular phones. In mobile networks either, the sensor nodes, targets, or both may be capable of movement. In some situations, this movement may be controllable by the network, other times it is not. Controllable movement would allow the network to maintain connectivity and cover even when a sensor node fails by moving excess sensor nodes to the failure point.

Circuit board layout techniques such as force directed placement algorithms may offer insight into these types of problems.

REFERENCES


[1] "Solar Resource & Meteorological Assessment Project Southwest Solar Solar CAT, January 12, 2011 Raw Data." [Online]. Available: http://www.nrel.gov/midc/swst_solarcat/

[2] D. Kirk, *Optimal Control Theory: An Introduction*. Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1970.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, no. 4, pp. 393 – 422, 2002.

[5] P. Torcellini, S. Pless, M. Deru, and D. Crawley, "Zero Energy Buildings: A Critical Look at the Definition," in *ACEEE Summer Study*. National Renewable Energy Laboratory, 2006.

[6] B. Gaudette, V. Hanumaiah, S. Vrudhula, and M. Krunz, "Optimal Range Assignment in Solar Powered Active Wireless Sensor Networks," in *Proc. of Infocom*, March 2012.

[7] G. Giorgetti, S. K. Gupta, and G. Manes, "Optimal RSS threshold selection in connectivity-based localization schemes," in *MSWiM '08: Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2008, pp. 220–228.

[8] M. Cardei, J. Wu, M. Lu, and M. Pervaiz, "Maximum Network Lifetime in Wireless Sensor Networks with Adjustable Sensing Ranges," in *Proc. of WiMob*, vol. 3, 2005, pp. 438 – 445.

[9] J. O'Rourke, *Computational Geometry in C*. Cambridge University Press, 1998.

[10] S. Dasika, S. Vrudhula, and K. Chopra, "Algorithms for optimizing lifetime of battery powered wireless sensor networks," *Sensor Network Operations, IEEE Press*, May 2006.

[11] C.-F. Huang and Y.-C. Tseng, "The Coverage Problem in a Wireless Sensor Network," in *Proc. of WSNA*. ACM, 2003, pp. 115–121.

[12] D. Diamond, S. Coyle, S. Scarmagnani, and J. Hayes, "Wireless Sensor Networks and Chemo-/Biosensing," *Chemical Reviews*, vol. 108, no. 2, pp. 652–679, 2008.

[13] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless Sensor Network Survey," *Computer Networks*, vol. 52, pp. 2292 – 2330, 2008.

[14] C. Liu, K. Wu, Y. Xiao, and B. Sun, "Random Coverage with Guaranteed Connectivity: Joint Scheduling for Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, pp. 562–575, 2006.

[15] H. Zhang and J. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," *NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, 2004.

[16] O. Younis, S. Ramasubramanian, and M. Krunz, "Operational Range Assignemtn in Sensor and Actor Networks," *Ad Hoc and Wireless Networks*, vol. 00, pp. 1–37, 2007.

[17] D. Niyato, E. Hossain, and A. Fallahi, "Sleep and Wakeup Strategies in Solar-Powered Wireless Sensor/Mesh Networks: Performance Analysis and Optimization," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 221–236, 2007.

[18] T. Voigt, A. Dunkels, J. Alonso, H. Ritter, and J. Schiller, "Solar-aware clustering in wireless sensor networks," *Computers and Communications*, vol. 1, pp. 238–243, 2004.

[19] J. J. Michalsky, "The Astronomical Almanac's Algorithm for Approximate Solar Position (1950-2050)," *Solar Energy*, vol. 40, pp. 227 – 235, 1988.

[20] "Solar Position Caculator," "National Renewable Energy Laboratory: Center for Renewable Energy Resources, Renewable Resource Data Center, Feb. 2000. [Online]. Available: http://www.nrel.gov/midc/solpos/

[21] "Most Efficient Solar Panels." [Online]. Available: http://sroeco.com/solar/most-efficient-solar-panels

[22] M. Skolnik, *Introduction to radar systems*. McGraw-Hill, 1962.

[23] H. Friis, "A Note on a Simple Transmission Formula," *Proceedings of the IRE*, vol. 34, no. 5, pp. 254 – 256, 1946.

[24] A. Wang, S. Cho, C. Sodini, and A. Chandrakasan, "Energy efficient Modulation and MAC for Asymmetric RF Microsensor Systems," in *Proceedings of the 2001 international symposium on Low power electronics and design*, ser. ISLPED '01. New York, NY, USA: ACM, 2001, pp. 106–111.

[25] T.-I. Kvaksrud, "Design Note DN018 Range Measurements in an Open Field Environment," Texas Instruments, 2008.

[26] "CC2500 Low-Cost Low-Power 2.4 GHz RF Transceiver (Rev. C)," Texas Instruments, 2011.

[27] D. Rakhmatov, S. Vrudhula, and D. Wallach, "A Model for Battery Lifetime Analysis for Organizing Applications on a Pocket Computer," *IEEE Transactions on VLSI Systems*, pp. 1019–1030, December 2003.

[28] R. Rao, S. Vrudhula, and D. Rakhmatov, "Battery Modeling for Energy-Aware System Design," *IEEE Computer, Special issue on Power-Aware & Temperature-Aware Computing*, vol. 36, no. 12, pp. 77–87, December 2003.

[29] D. Rakhmatov and S. Vrudhula, "Energy Management for Battery-Powered Embedded Systems," *ACM Transactions on Embedded Computing Systems*, vol. 2, no. 3, pp. 277–324, August 2003.

[30] S. Mehrotra, "On the Implementation of a Primal-Dual Interior Point Method," *SIAM Journal on Optimization*, vol. 2, no. 4, pp. 575–601, 1992.

[31] J. J. E. Kelley, "The Cutting-Plane Method for Solving Convex Programs," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 4, pp. 703–712, 1960.

[32] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, pp. 1–33, 1983.

[33] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1982.

[34] Y. Ye, "An O($n^3l$) potential reduction algorithm for linear programming," *Mathematical Programming*, vol. 50, pp. 239–258, 1991.

[35] "eZ430-RF2500 Development Tool User's Guide (Rev. E)," Texas Instruments, 2009.

[36] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power Management in Energy Harvesting Sensor Networks," *ACM Trans. Embed. Comput. Syst.*, vol. 6, 2007.

[37] "EZ430-RF2500T – MSP430 2.4-GHz Wireless Target Board." [Online]. Available: https://estore.ti.com/EZ430-RF2500T-MSP430-24-GHz-Wireless-Target-Board-P1295.aspx

[38] "Sun electronics." [Online]. Available: http://www.sunelec.com/

[39] Battery energy: What battery provides more? [Online]. Available: http://www.allaboutbatteries.com/Battery-Energy.html