Adaptive Selection Problems in Networked Systems

THÈSE N^{O} 6762 (2015)

PRÉSENTÉE LE 23 OCTOBRE 2015
À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE COMMUNICATIONS AUDIOVISUELLES
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Runwei ZHANG

acceptée sur proposition du jury:

Prof. E. Telatar, président du jury
Prof. M. Vetterli, Prof. P. Thiran, directeurs de thèse
Prof. M. Johansson, rapporteur
Prof. G. Zussman, rapporteur
Dr L. Olivier, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2015

Abstract

Networked systems are composed of interconnected nodes that work collaboratively to maximize a given overall utility function. Typical examples of such systems are wireless sensor networks (WSNs) and participatory sensing systems: sensor nodes, either static or mobile, are deployed for monitoring a certain physical field. In these systems, there are a set of problems where we need to adaptively select a strategy to run the system, in order to enhance the efficiency of utilizing the resources available to the system. In particular, we study four adaptive selection problems as follows.

We start by studying the problem of base-station (BS) selection in WSNs. Base stations are critical sensor nodes whose failures cause severe data losses. Deploying multiple fixed BSs improves the robustness, yet this scheme is not energy efficient because BSs have high energy consumptions. We propose a scheme that selects only one BS to be active at a time; other BSs are kept passive and act as regular sensor nodes. This scheme substantially reduces the energy supplies required by individual BSs. Then, we propose an algorithm for adaptively selecting the active BS so that the spatially and temporally varying energy resources are efficiently utilized. We also address implementation issues and apply the proposed algorithm on a real WSN. Field experiments have shown the effectiveness of the proposed algorithm.

We generalize the BS selection problem by considering both the energy efficiency of regular sensor nodes and that of BSs. In this scheme, a subset of active BSs (instead of only one) is adaptively selected and the routing of regular sensor nodes is adjusted accordingly. Because BSs have high fixed-energy consumptions and because the number of candidate subsets of active BSs is exponential with the number of BSs, this general BS selection problem is NP-hard. We propose a polynomial-time algorithm that is guaranteed, under mild conditions, to achieve a network lifetime at least 62% of the optimal one. Through extensive numerical simulations, we verify that the lifetime achieved by the proposed algorithm is always very close to the optimum.

We then study the problem of scheduling the sparse-sensing patterns in WSNs. We observe that the traditional scheme of periodically taking sensing samples is not energy efficient. Instead, we propose to adaptively schedule when and where to activate sensors for sampling a physical field, such that the energy efficiency is enhanced and the sensing precision is maintained. The schedules are learnt from the temporal signal models derived from the collected measurements. Then, using the obtained signal models and the sparse sensing-measurements, the original signal can be effectively recovered. This proposed method requires minimal on-board computation, no inter-node communications and achieves an appealing reconstruction performance. With experiments on real-world datasets, we demonstrate significant improvements over both traditional sensing schemes and the state-of-the-art sparse-sensing schemes, particularly when the measured data is characterized by a strong temporal correlation.

In the last part of the thesis, we discuss the sparse-sensing framework by exploiting the spatial correlations rather than the temporal correlations among the captured measurements. In this framework, application-specific utility functions can be employed. By adaptively selecting a small subset of active sensors for sensing, a certain utility is guaranteed and the efficiency of the sensing system is enhanced. We apply this framework both in static WSNs and participatory sensing systems where sensors move in an uncoordinated manner. Through extensive simulations, we show that our proposed algorithm enhances the resource efficiency.

Keywords: networked system, wireless sensor networks, participatory sensing, resource efficiency, adaptive selection problem, network optimization, sparse sampling.

Résumé

Les réseaux de systèmes sont composés de nœuds interconnectés qui travaillent en collaboration pour la maximisation d'une fonction objectif globale donnée. Deux exemples typiques sont les réseaux de capteurs sans fil (WSN) et les systèmes de collecte de données participatifs: les capteurs, soit statiques, soit mobiles, sont déployés pour superviser un certain champ de captage. Dans ces systèmes, il y a un certain nombre de situations où il est nécessaire de sélectionner de façon adaptative une stratégie de fonctionnement pour le système qui améliore l'efficacité de l'utilisation des ressources à disposition du système. Nous étudions en particulier les quatre problèmes de sélection adaptative suivant.

Nous commençons par étudier le problème de la re-sélection de la station de base (BS) dans les WSNs. Les stations de base sont des capteurs critiques dont la défaillance peut causer de graves pertes de données. L'utilisation de plusieurs BSs au sein du même réseau améliore sa robustesse, néanmoins cette stratégie n'offre pas un rendement optimal car les BSs ont une consommations en énergie élevée. Nous proposons une stratégie qui sélectionne uniquement une BS à la fois pour être active; les autres BSs sont maintenues passives et agissent en tant que capteurs réguliers. Cette stratégie réduit sensiblement les réserves d'énergie nécessaires aux BSs. Puis, nous proposons un algorithme pour re-sélectionner de manière adaptative les BS actives de façon à ce que les ressources énergétiques, variables dans l'espace et le temps, soient utilisées efficacement. Nous abordons également les questions de mise en oeuvre et appliquons l'algorithme proposé sur un WSN réel. Les expériences sur le terrain ont démontré l'efficacité de l'algorithme proposé.

Nous généralisons le problème de re-sélection des BSs en considérant le rendement énergétique des capteurs réguliers et des BSs. Dans cette stratégie, un sous-ensemble de BSs actif (au lieu d'une seule) est re-sélectionné de manière adaptative et le routage des capteurs réguliers est ajusté en conséquence. Puisque les BSs ont une consommation énergétique avec un coût fixe élevé et parce que le nombre de sous-ensembles de BSs actives candidates est exponentiel dans le nombre de BSs, ce problème général de re-sélection des BSs est NP-difficile. Nous proposons un algorithme à temps polynomial qui est garanti, sous des conditions modérées, d'atteindre une durée de vie du réseau d'au moins 62% de l'optimal. Par des simulations numériques poussées, nous vérifions que la durée de vie atteinte par l'algorithme proposé est toujours très proche de l'optimum.

Nous continuons notre étude du problème de la planification des motifs de captages parcimonieux dans les WSNs. Nous observons que la stratégie classique qui consiste à acquérir périodiquement des échantillons n'est pas efficace énergétiquement parlant. Nous proposons donc à la place de planifier adaptativement quand et où les capteurs sont activés pour échantillonner le champs physique de façon à améliorer l'efficacité énergétique tout en maintenant la précision des mesures. Les planning sont appris sur la base des des modèles de signaux temporelles dérivées des mesures collectées. Puis, à l'aide des modèles de signaux obtenus et des échantillons parcimonieux capturés, les signaux originaux peuvent être efficacement récupérés. La méthode proposée nécessite peu de calculs embarqués, aucune communications entre les capteurs et atteint une performance de reconstruction attrayante. En outre, grâce à des expériences sur des jeux de données provenant de systèmes utilisés en pratiques, nous démontrons une amélioration significative en comparaison l'état de l'art en matière de stratégies de captage traditionnelles et parcimonieuse, tout particulièrement quand les données mesurées sont caractérisée par une corrélation temporelle forte.

Dans la dernière partie de la thèse, nous discutons le cadre de captage parcimonieux qui exploite la corrélation spatiale plutôt que temporelle dans les mesures capturées. Dans ce cadre, des fonctions utilité spécifiées par l'utilisateur peuvent être utilisée. En re-sélectionnant adaptativement un petit sous-ensemble de capteurs actifs pour le captage, une utilité donnée est garantie et l'efficacité du captage du système est améliorée. Nous appliquons ce cadre à la fois dans les WSNs statiques et dans le captage mobile, où les capteurs se meuvent de façon non-coordinée. Par des simulations numériques poussées, nous démontrons que l'algorithme que nous proposons améliore l'efficacité de l'utilisation des ressources et maintient l'utilité du captage.

Mot clés: Les réseaux de systèmes, les réseaux de capteurs sans fil, les systèmes de collecte de données participatifs, l'efficacité de l'utilisation des ressources, problème de sélection adaptative, l'optimisation du réseau, échantillonnage clairsemé

Acknowledgments

First, I would like to express my sincere gratitude to my two supervisors: Martin Vetterli and Patrick Thiran. For a Ph.D student, the best thing that could happen is to have a good supervisor; I had such privileges in double. Both Martin and Patrick are enthusiastic researchers and they keep the pure scientific spirit. I am so lucky to have been supervised by these two world-renowned researchers. I still remember the days when I first met them; I knew so little about my thesis topic and research. They taught me the basic skills from scratch and continued to encourage me to achieve my full potential. Without them, I would never have had a chance to reach my current status. And, I really owe them a lot for correcting my English writing hundreds of times. I have to thank again for all their patience, for their trust, and for their continued encouragement. It is really amazing to work with both of them.

Then, I would like to take this opportunity to thank my committee members, Prof. Mikael Johansson, Prof. Gil Zussman, Prof. Ruediger Urbanke and Dr. Olivier Lévêque. Thanks go to them for accepting to read my thesis and to assess my work.

I really appreciate the help from my collaborators, Francois Ingelrest, Guillermo Barrenetxea, Zichong Chen, Juri Ranieri and Bejar Haro Benjamin. The discussions with them were very effective and inspiring; they always have the right tools to tackle my problems. Without their help, I would have a much harder time finishing my thesis.

Next, I thank the Swiss National Science Foundation (SNSF) and EPFL for their financial supports. Thanks to Jacqueline, the “actual manager” of the lab, whenever I needed anything or had any problems apart from research, she was always there to provide the solution. And thanks to Holly, my English tutor. She went through my drafts many, many times and made great efforts to teach me how to write as a native speaker. Thanks a lot for the lab mates in both LCAV and LCA, Ivan, Robin, Mihailo, Marta, Juri, Feng, Zhou, Yun, Hanjie, Farid, Vincent, Julien and Mohamed. I will always cherish the memories with them of the hiking days, the ski days, the Christmas dinners and the Chinese-fondue nights. Special thanks go to my office mate, Zichong. We shared many things together: coffee breaks, dinners, excursion trips and some random ideas on smart homes. He made my stay in the office a very pleasant experience!

I would also like to thank all the friends I met in Lausanne, Hu Xu, Li Pu, Feng Yang, Wenqi You, Charles Dubois, Xiaolu Sun, Weijia Gan, Jiaqing Du, Yuxuan Ji, Yun Bai, Na Li, Florent Garcin, Min Ye, Qingxia Zhong and many others I cannot list. Thanks them for making my life in Lausanne always fun.

Last but not least, I would like to thank my father Liangyu, my mother Aihong and my sister Kejia for their consistent support and unconditional love. They gave me the courage and almost everything else I needed to reach my current stage of life. My life has always been splendid with them.

Notations

Active Base-Station Selection in Wireless Sensor Networks

τ	Length of a time slot $\in \mathbb{R}^+$
V_b ($\lvert V_b \rvert$)	The set of BSs (number of BSs)
e	Initially available energy of every BS $\in \mathbb{R}^+$
$\boldsymbol{e}^{(n)}$	Available energy of all BSs at the n-th time slot $\in \mathbb{R}^{\lvert V_b \rvert}$
N	Number of time slots before the first BS drains out of energy$\in \mathbb{N}^+$
$\boldsymbol{u}$	The uniform vector $[1, 1, \cdots 1]^\top \in \mathbb{R}^{\lvert V_b \rvert}$
$\boldsymbol{v}^{(n)}$	Decision vector at the n-th time slot $\in \{0, 1\}^{\lvert V_b \rvert}$
$\boldsymbol{C}$	Cost matrix $\in \mathbb{R}^{\lvert V_b \rvert \times \lvert V_b \rvert}$
$\boldsymbol{s}^{(n)}$	Energy-recharge rates of all BSs at the n-th time slot $\in \mathbb{R}^{\lvert V_b \rvert}$
$\bar{\boldsymbol{s}}$	Average energy-recharge rates of all BSs $\in \mathbb{R}^{\lvert V_b \rvert}$

Joint Selection of Base Stations and Routing

τ	Length of a time slot $\in \mathbb{R}^+$
V_b ($\lvert V_b \rvert$)	Set of BSs (number of BSs)
V_r ($\lvert V_r \rvert$)	Set of RSNs (number of RSNs)
e_i	Initially availaible energy of BS or RSN i, $e_i \in \mathbb{R}^+$
$\boldsymbol{e}$	Initially availaible energy of all RSNs and BSs $\in \mathbb{R}^{\lvert V_b \cup V_r \rvert}$
$\boldsymbol{e}^{(n)}$	Availaible energy of all RSNs and BSs at the n-th time slot $\in \mathbb{R}^{\lvert V_b \cup V_r \rvert}$
N	Number of time slots before the first BS or RSN drains out of energy $\in \mathbb{N}^+$
r	Data-generating rate of each RSN $\in \mathbb{R}^+$
x_{ij}	Data rate on short-range communication link from node i to node j, $x_{ij} \in \mathbb{R}^+$
$\boldsymbol{x}$ ($\boldsymbol{x}^{(n)}$)	Data rates on all short-range communication links (at the n-th time slot)
y_i	Data rate on the long-range communication link of BS j, $y_i \in \mathbb{R}^+$
$\boldsymbol{y}$ ($\boldsymbol{y}^{(n)}$)	Data rates on long-range communication link of BSs (at the n-th time slot) $\in \mathbb{R}^{\lvert V_b \rvert}$
z_i	Indicator of activity state of BS i, $z_i \in \{0, 1\}$
$\boldsymbol{z}$ ($\boldsymbol{z}^{(n)}$)	Indicator vector of activity states of all BSs (at the n-th time slot) $\in \{0, 1\}^{\lvert V_b \rvert}$
c_i	Energy-consumption rate of node i, $c_i \in \mathbb{R}^+$
$\boldsymbol{c}$ ($\boldsymbol{c}^{(n)}$)	Energy-consumption rates of all RSNs and BSs (at the n-th time slot) $\in \mathbb{R}^{\lvert V_b \cup V_r \rvert}$
c_{c}, c_{st}, c_{sr}, c_{lc}, c_{lt}	Constant parameters for modelling the energy consumption rates $\in \mathbb{R}^+$
$\boldsymbol{\mathcal{L}}$	Set of all candidate configurations

$\mathcal{L}^{[k]}$	Set of candidate configurations given that $\boldsymbol{z} = \boldsymbol{z}^{[k]}$, $1 \leq k \leq 2^{\lvert V_b \rvert} - 1$
θ_i	Average energy-decrease rate of node i, $\theta_i \in \mathbb{R}^+$
$\boldsymbol{\theta}$ ($\boldsymbol{\theta}^{(n)}$)	Average energy-decrease rate of all RSNs and BSs (at the n-th time slot) $\in \mathbb{R}^{\lvert V_b \cup V_r \rvert}$
T^*	Optimal lifetime of the virtually-moving BSs problem

Sparse Sensor-Selection by Exploiting Temporal Correlations

N	Desired number of samples in a block $\in \mathbb{N}^+$
M	Number of measurements in a block, equals $\lfloor N\gamma \rfloor$, $M \in \mathbb{N}^+$
τ	Temporal resolution of original signal $\in \mathbb{R}^+$
f	Sampling frequency of original signal, equals $1/\tau$, $f \in \mathbb{R}^+$
f_s	Average sampling frequency of the sensor $\in \mathbb{R}^+$
γ	Subsampling rate f_s/f, $\gamma \in \mathbb{R}^+$
$\widetilde{\boldsymbol{x}}$	Reconstructed signal $\in \mathbb{R}^N$
$\boldsymbol{x}$	Original signal $\in \mathbb{R}^N$
$\boldsymbol{y}$	Measured signal $\in \mathbb{R}^M$
$\boldsymbol{\omega}$	Measurement noise $\in \mathbb{R}^M$
$\boldsymbol{\chi}^t$	Sampling pattern of the t-th block
$\boldsymbol{\Phi}^t$	Sampling matrix of the t-th block $\in \mathbb{R}^{M \times N}$
$\overline{\boldsymbol{x}}$	Mean of the signal $\in \mathbb{R}^N$
$\boldsymbol{\Psi}^t$	Signal model of the t-th block $\in \mathbb{R}^{N \times K}$
$\boldsymbol{\alpha}$	Low-dimensional representation of $\boldsymbol{x} \in \mathbb{R}^K$
$\widetilde{\boldsymbol{\Psi}}^t$	Rows of $\boldsymbol{\Psi}^t$ selected by $\boldsymbol{\tau}^t \in \mathbb{R}^{M \times K}$

Sparse Sensor-Selection by Exploiting Spatial Correlations

M	Number of sensors $\in \mathbb{N}^+$
$\mathcal{M}$	Set of all sensors $\{1, 2, \cdots, M\}$
τ	Length of a time slot $\in \mathbb{R}^+$
$g_n(\cdot)$	Utility function defined at the n-th time slot
$u_n(\cdot)$	The minimum of the utility function $g_n()$ and the desired utility q
$\mathcal{A}_n$	The subet of selected sensors $\in 2^{\mathcal{M}}$
e_m	Initially available resource of sensor m, $e_m \in \mathbb{R}^+$
$e_m^{(n)}$	Available resource of sensor m at the n-th time slot, $e_m^{(n)} \in \mathbb{R}^+$
N^*	Maximum number of time slots running our proposed algorithm $\in \mathbb{N}^+$
N_{opt}	Optimal number of time slots of problem 5.2, $\in \mathbb{N}^+$
$\lambda_m^{(n)}$	Penalty we assigned on sensor m at the n-th time slot, $\in \mathbb{R}^+$
θ_m ($\theta_m^{(n)}$)	Average resource-consumption rate of sensor m (in the first n time slots), $\in \mathbb{R}^+$
$\boldsymbol{\theta}$ ($\boldsymbol{\theta}^{(n)}$)	Average resouce-consumption rate of all sensors (at time slot n), $\in \mathbb{R}^M$
$\mathcal{M}(q)$	Set of all subsets of $\mathcal{M}$ that has a utility at least q
$p_{\mathcal{A}}$	Fraction of time that the subset of sensors $\mathcal{A}$ is activated $\in \mathbb{R}^+$

Contents

Abstract iii

Résumé v

Acknolwedgement vii

Notations viii

1 Introduction 1
- 1.1 Wireless Sensor Networks . . . 2
- 1.2 Participatory Sensing Systems . . . 4
- 1.3 Adaptive Selection Problems . . . 5
  - 1.3.1 Active Base-Station Selection . . . 5
  - 1.3.2 Joint Selection of Active Base-Stations and Routing . . . 6
  - 1.3.3 Sparse Sensor-Selection by Exploiting Temporal Correlations . . . 7
  - 1.3.4 Sparse Sensor-Selection by Exploiting Spatial Correlations . . . 8
- 1.4 Contributions . . . 8

2 Active Base-Station Selection in Wireless Sensor Networks 11
- 2.1 Introduction . . . 11
- 2.2 System Architecture . . . 12
  - 2.2.1 Network Details . . . 13
  - 2.2.2 Starting the Network . . . 13
  - 2.2.3 Gathering the Information for Adaptive Selections . . . 15
  - 2.2.4 Handing Over the Active BS . . . 15
  - 2.2.5 Recovering from Failures . . . 16
- 2.3 Adaptive BS-Selection Problem Formulation . . . 16
- 2.4 The “Highest Energy First” (HEF) Algorithm . . . 19
- 2.5 Simulations . . . 21
  - 2.5.1 General Settings . . . 21
  - 2.5.2 Performance of Different Algorithms . . . 22
  - 2.5.3 Validations of Optimality Conditions . . . 25
- 2.6 Real Experiments . . . 27
- 2.7 Conclusion . . . 29
- 2.A Appendix . . . 30

2.A.1 Azuma-Hoeffding inequality 30
2.A.2 Proof of Theorem 2.1 31
2.A.3 Proof of Theorem 2.2 34

3 Joint Selection of Base Stations and Routing 43
3.1 Introduction 43
3.2 System Model 45
3.2.1 Data Communication 45
3.2.2 Energy Consumption 45
3.3 Problem Formulations 46
3.3.1 The Optimization Space 46
3.3.2 The Virtually-Moving BSs Problem 46
3.3.3 Comparisons to Other Schemes 47
3.4 Complexity Analysis 48
3.5 Scheduling Algorithm 50
3.5.1 The Constrained Gradient Method 51
3.5.2 The Min-Weight Configuration Problem 53
3.6 Simulations 56
3.6.1 Parameter Selection 57
3.6.2 Performance Comparison 58
3.7 Conclusion 60
3.A Appendix 60
3.A.1 Proof of Theorem 3.2 60
3.A.2 Proof of Theorem 3.3 63

4 Sparse Sensor-Selection by Exploiting Temporal Correlations 67
4.1 Introduction 67
4.2 Related Works 69
4.3 Problem Formulation 71
4.4 Building Blocks 73
4.4.1 Signal Approximation and Reconstruction 74
4.4.2 Learning from Incomplete Data over Time 77
4.4.3 Sampling-Schedule Algorithm 78
4.5 Comparisons with Baseline Methods 79
4.6 Evaluations of DASS and Sparse-Sensing Methods 81
4.6.1 Components of DASS 81
4.6.2 DASS versus Baseline Methods 84
4.6.3 DASS on Multiple Sensor Nodes 85
4.7 Energy Saving over Traditional Data Collection Schemes 86
4.8 Conclusions 89
4.A Appendix 89

5 Sparse Sensor-Selection by Exploiting Spatial Correlations 91
5.1 Introduction 91
5.2 Related Works 92
5.3 Problem Formulations 93

5.4 The Proposed Framework 94
5.5 Analyses in the Static WSN Scenarios 97
5.6 Performance Evaluation 98
5.6.1 Ideal Scenarios 99
5.6.2 Practical Scenarios 100
5.7 Conclusions 101
5.A Appendix 102
5.A.1 Proof of Lemma 5.1 102
5.A.2 Proof of Theorem 5.1 102
5.A.3 Proof of Theorem 5.2 104

6 Conclusion and Future Work 107

Bibliography 111

Curiculum Vitæ 117

Chapter 1

Introduction

> It is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail.
>
> *Abraham Harold Maslow*

We live in networks. Everyday on this planet, communication networks, transportation networks, social networks, and the internet of things are connecting data, people, money and goods to improve the well-being of society. As these systems are of central importance to the human race, they serve as the basis of our modern society.

Communication networks include cellular networks, Wi-Fi local networks and wireless mesh networks. These networks facilitate the exchanges of information through texts, emails, voices, and videos. They enable people to collaborate on projects from wide distances and to execute transactions across the globe.

Social networks, as a new type of information networks, have exploded in popularity in the last decade. Popular sites such as Facebook, Twitter, and LinkedIn have reported to have hundreds of millions of users. As these sites facilitate information propagation, they are becoming increasingly important for online networking among people.

The internet of things designate the networks consisting of physical objects embedded with sensors and communication electronics. The components in these networks autonomously collect useful data from the environment and then execute certain pre-defined operations. Typical applications of the internet of things include wireless sensor networks (WSNs), smart grids, home automation, and smart transportation systems.

Although all the aforementioned systems play different roles in our lives, they share one essential characteristic in common: the components of these systems work cooperatively for achieving a certain overall utility. We call such systems **networked systems**. To fully understand these

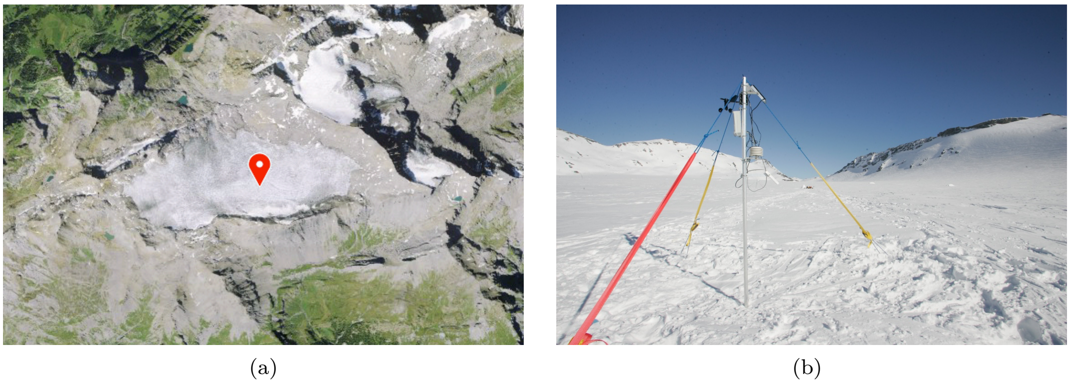

(a) (b)

Figure 1.1: The WSN deployment on a swiss glacier site Plaine Morte. As shown in Figure 1.1a, the Plaine Morte glacier is at 2750m above sea level in the canton of Berne in Switzerland. It is one of the largest flat-glaciers in the Alps. It extends for about 10km^2 and is located in the Wildstrubel range. To measure the environmental data on this site, dozens of sensor nodes were deployed as shown in Figure 1.1b.

systems, analyze them and optimize them, substantial research efforts have been made. This thesis is also a part of those efforts.

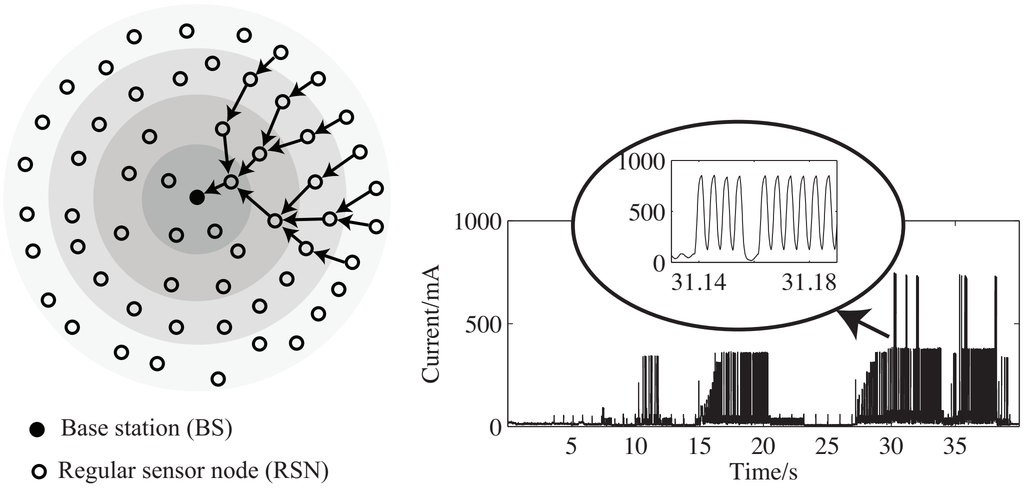
In particular, this thesis focuses on two networked systems: wireless sensor networks (WSNs) and participatory sensing systems.

1.1 Wireless Sensor Networks

Wireless sensor networks (WSNs) are used for monitoring physical conditions in many applications, *e.g.*, environmental sensing, building monitoring, surveillance, and precision agriculture. For example, the Sensorscope [34] project in EPFL, which we worked on for the last several years, provides the public and environmental scientists with real-time, continuous and fine-grained data. The deployments were in several alpine mountains where there are concerns about the melting of the permafrost and mud streams, as shown in Figure 1.1.

In WSNs, there are two types of sensor nodes, **regular sensor nodes (RSNs)** and **base stations (BSs)**. Regular sensor nodes are tiny devices that perform sensing and transmit the captured data using short-range communication, *e.g.*, 802.15.4/Zigbee. The low-cost RSNs are envisioned to be massively produced for ubiquitous sensing. They are usually equipped with very little resources for computation and communication, and they have to work cooperatively to achieve certain objectives, *e.g.*, relay data packets in a multi-hop manner to a BS. The BSs are the key sensor nodes that collect data across the whole network and then forward it to a remote server by using long-range communication, *e.g.*, GSM/GPRS, 3GPP or LTE. They serve as communication bridges between the sensing field and the remote server.

Energy efficiency is always a major concern in WSNs. This is because WSNs are designed to be used in scenarios where wireline sensor networks are not applicable, and in these scenarios, the energy supplies of WSNs are usually from batteries of limited capacity and from energy scavenging devices of limited capability. Therefore, given the limited available energy resources,



(a) (b)

Figure 1.2: The bottleneck-effects of the BS. In Figure 1.2a, we show a WSN with a single BS and some RSNs in rings around that BS. Regular sensor nodes in the rings near the BS have high communication loads and therefore have high energy consumptions. We use darker color for the rings with higher energy consumptions. In Figure 1.2b, we show the power consumption of a Telit GM-862 GSM/GPRS module when actively connecting to a remote server. This module has to be activated for around 40s with an average power of 296mW just to start and stop each connection. In the meantime, a typical power of a RSN is just a few milliwatts if duty cycling is used. Therefore, the energy consumption of a BS is much higher than that of RSNs.

it is imperative to improve the energy efficiency, or in other words, to extend the operating time of the WSNs. We call the operating time of WSNs the **lifetime**. There are many different ways to define lifetime, such as the time that the first sensor node depletes its energy or the time that a certain fraction of sensor nodes deplete their energy. Throughout this thesis, we define the lifetime as the time that the first sensor node depletes its energy.

In a traditional WSN, there is only one BS and a number of RSNs deployed around the BS. Researchers focus on designing energy-efficient MAC [22, 68] protocols and routing protocols [3, 16, 72] in order to optimize the multi-hop and short-range communication of RSNs. These techniques, to some extent, improve the energy efficiency of a WSN. However, researchers soon discover that the single BS is the bottleneck of the whole network due to the following three reasons:

(i) Using only one BS creates a single point of failure. In many applications, WSNs are deployed in adversarial and harsh environments (*e.g.*, on the glacier Plaine Morte as shown in Figure 1.1). Therefore, the BS is exposed to either human-induced destructions or natural hazards. Once the single BS experiences a failure, the whole WSN gets stuck and starts losing precious data.

(ii) Regular sensor nodes near the single BS deplete their batteries quickly. As shown in Figure 1.2a, these RSNs have to forward the data from the whole network, and therefore they have high energy consumptions. This problem is known as the “energy-hole” problem.

(iii) The single BS itself consumes a large and fixed amount of energy to forward the collected data by using long-range communication. As shown in Figure 1.2b, the BS has to activate its GSM/GPRS module for 40 seconds just to start and stop each connection with the remote server. The average power is 296mW during that time. In the meantime, the typical power of a RSN is just a few milliwatts if duty cycling is used. Depending on the sampling rate of RSNs, the energy consumption of the BS can be one or two orders of magnitude more than that of RSNs.

There are some existing solutions for partially solving the aforementioned problems: deploy multiple BSs or physically move BSs. (i) Deploying multiple BSs [7, 29, 59] solves the issue of single point of failure and alleviates the energy-hole problem. Using this scheme, the communication load will not be concentrated around a single BS, instead, it will be distributed around the multiple deployed BSs. This results in a balanced energy-consumption load among the whole network. However, the long-range communication on all BSs causes high energy consumptions of the whole network. Therefore, this scheme is not energy-efficient if we consider both the energy efficiency of RSNs and BSs. (ii) Physically moving BSs [4, 6, 25, 44, 45, 55, 56] can further enhance the energy efficiency of the WSN because the communication loads in the networks will be more balanced. However, physically moving BSs require additional implements like vehicles, and the movement itself might be quite energy-consuming.

1.2 Participatory Sensing Systems

Static WSNs are capable of providing accurate and stable sensing measurements. These WSNs are usually deployed by either authorities or specialists. Due to the high maintenance-costs, the number of installations are too limited to provide appealing sensing results on a large scale.

Participatory sensing systems provide an alternative means for people to share sensory information. We envision that low-cost sensors are massively produced and are embedded on agents like public vehicles, bicycles or mobile phones. These sensors travel with the agents and take sensing measurements along the moving trajectories without intervening in the agents' activities. The captured sensing samples collectively form a body of knowledge and achieve application-specific utilities. Due to the increasing number of smart phones and the versatile sensors installed on them, participatory sensing has already enabled many interesting applications. For example, in the project **CarTel**[33], a number of mobile phones are installed on private vehicles to monitor traffic information. In the project **Earphone** [50], researchers use microphones on mobile devices to collect the environmental noise level in urban areas. In the project **OpenSense** [2] which we worked on at EPFL, sensor boxes are installed on public vehicles, such as buses and metros, for monitoring air pollution levels in the city (as shown in Figure 1.3).

Although these applications are promising, they face several challenges when used on a large scale. One major challenge is due to the limited resources of these mobile devices, that is, battery energy, communication bandwidth and sensitive location information. If a mobile user keeps the sensing application on, the energy-hungry sensors (e.g., GPS, microphones) will drain the battery quickly, their communication bandwidth will be occupied and their location privacy will be easily compromised. Therefore, a major design concern in participatory sensing systems is to extend the lifetime of these systems given a limited amount of resources. To address this concern, researchers consider two approaches:

(i) The first approach is to exploit free-lunch opportunities [42, 61], that is, those times when mobile users place phone calls or use applications. Using this approach, mobile users do not have

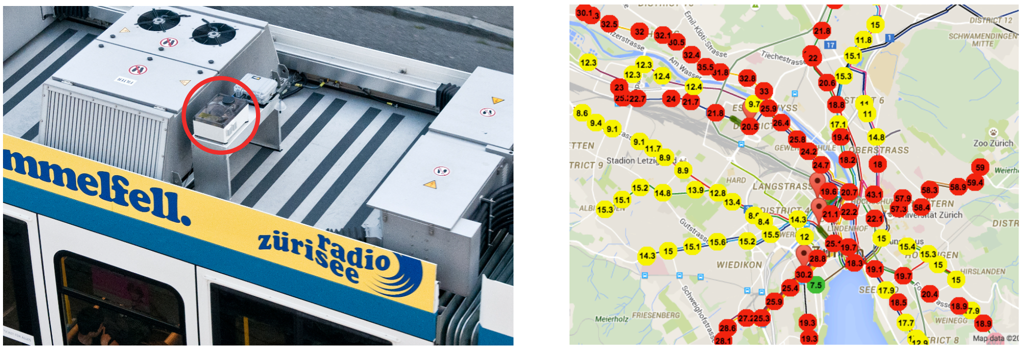

(a) (b)

Figure 1.3: The Opensense [2] project using community-based sensing techniques to monitor air pollutions in the city of Zurich [1]. The sensor boxes (in the red circle in Figure 1.3a) are placed on public transportation vehicles for real-time monitoring. The captured Ozone levels are shown in Figure 1.3b.

to pay extra resources or leak extra privacy information in the sensor data.

(ii) The second approach is to schedule the sensing activities of mobile sensors and to let them cooperate [67]. Using this approach, mobile sensors take turns to actively sense and rest. Through careful scheduling, the number of sensing samples to be taken will be reduced, the high loads of resource consumptions will be evenly distributed onto all mobile sensors, and in the meantime, certain sensing utility will be guaranteed.

1.3 Adaptive Selection Problems

In the aforementioned networked systems, there are a set of problems we call **adaptive selection problems**. In these problems, we need to adaptively select a strategy to run the system, in order to perform load balancing or to enhance the efficiency of utilizing the resources available to the system. We study in particular four adaptive selection problems as follows.

1.3.1 Active Base-Station Selection

As shown in Figure 1.2b, the energy consumption for using long-range communication is much higher compared to that for using short-range communication. Therefore, the BS usually causes the bottleneck of the lifetime in a medium-sized WSN. To alleviate the high energy consumption on the single BS, we propose a novel scheme for organzing the WSN, as shown in Figure 1.4. In this scheme, we deploy multiple BSs and adaptively re-select one active BS to use the long-range communication, so as to emulate the physical movement of a single BS. The active BS collects data from all RSNs and maintains long-range communication with the remote server. Meanwhile, passive BSs behave as RSNs. They turn off their long-range communication devices, only sampling and forwarding data by using short-range communications. Using this scheme, the high energy consumption of the active BS for using long-range communication is shared

---

[1] The figures we use here are from the Opensense website http://www.opensense.ethz.ch.

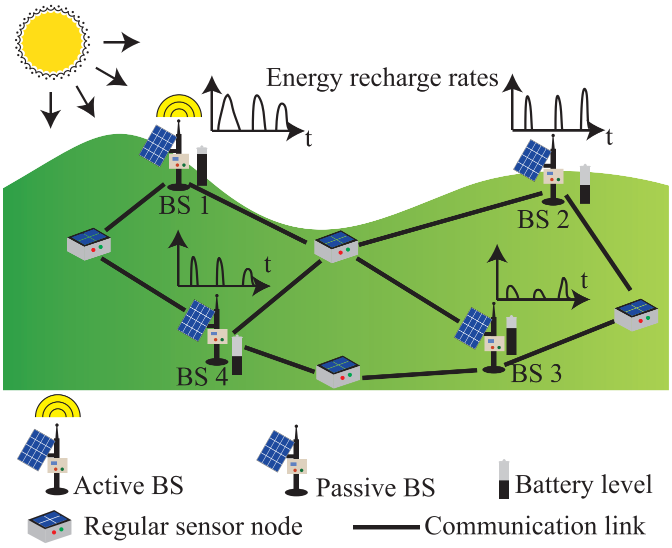


Figure 1.4: A WSN with the proposed scheme that deploys multiple BSs, keeps only one of them active and adaptively re-selects this active BS. At the current time, BS 1 is active. Some time later, the active BS will be re-selected based on the state of the network, *e.g.*, battery levels. By using this scheme, the temporally and spatially varying energy resources of all BSs are fully utilized.

among all candidate BSs. The batteries of all candidate BSs form a pool, thus emulating a larger global power source. To build a sustainable WSN, the requirement is that the total energy harvested by all candidate BSs sustains the consumption of the active BS. Consequently, the size of the individual power sources can be substantially reduced. In the meantime, the energy-hole problem is alleviated because the BS is virtually moving. Also, the hassles for physically moving the BSs is avoided. This scheme also avoids the single-point of failure due to the existence of multiple BSs. When the network has connectivity problems and splits into several connected components, the aforementioned active-BS selection process automatically takes place in all these small components.

Because we adaptively re-select one active BS, this scheme is also called *"virtually moving one BS"*. When designing the algorithm for selecting the active BS, we should only use easily-available information as the input due to the limited capability of WSN hardwares. Also, the proposed algorithm should maximally use the temporally-varying energy available to all BSs when sensor nodes are equipped with energy scavenging devices like solar panels.

1.3.2 Joint Selection of Active Base-Stations and Routing

In Section 1.3.1, we select the active BS for evenly distributing the high energy consumption of long-range communication. However, this does not necessarily optimize the short-range communication among RSNs. In a large-scale WSN, the energy consumption of the short-range communication among RSNs accumulates quickly along the routing paths. In this scenario, the bottleneck of the network lifetime might be caused by RSNs instead of BSs. Therefore, a more general problem is to optimize the energy efficiency of BSs and RSNs together. In order to solve

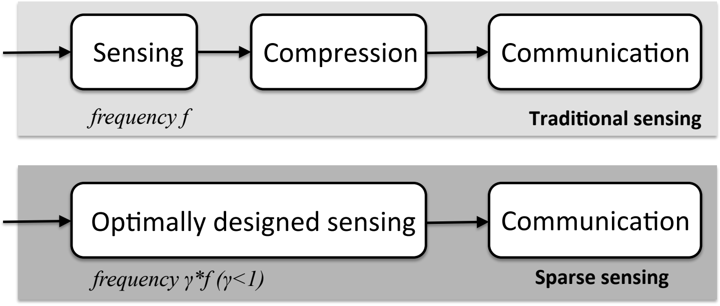


Figure 1.5: Two approaches to sensing in a WSN node. (a) Traditional scheme: collect periodical samples at a frequency f, compress and transmit the compressed data. (b) The sparse sensing scheme: collect samples with an optimized temporal pattern at an average frequency $\gamma \cdot f$ and transmit the raw data.

this problem, we generalize the scheme *"virtually moving one BS"* into *"virtually moving multiple BSs"*. In this scheme, a subset of active BSs, instead of one active BS, need to be adaptively re-selected and the routing need to be adjusted accordingly. Using this scheme, the available energy of all BSs and RSNs across the WSN will be optimally utilized.

The general scheme of virtually moving multiple BSs turns out to be complex, due to the increased number of degrees of freedom. Because active BSs consume a large and fixed amount of energy for using long-range communications, scheduling the virtual movement of multiple BSs is a mixed integer programming problem. Therefore, it is much more complex compared to the traditional maximum lifetime routing problem (e.g., [16]). Indeed, we will show later that the fixed energy-consumption of long-range communication makes the problem NP-hard using a reduction from the 3-SAT [9].

1.3.3 Sparse Sensor-Selection by Exploiting Temporal Correlations

Another adaptive selection problem that we consider is the sparse-sampling problem, both in static WSNs and in participatory sensing systems.

The energy consumption of a sensor node comes mainly from three activities: sensing, data-processing and communication. Traditionally, the costs for processing and communication are assumed to dominate the overall energy consumption, whereas the cost for sensing is considered negligible. Therefore, in a traditional sensing system, as shown in the upper part of Figure 1.5, sensor nodes take samples as often as possible, they are subsequently compressed and transmitted with the lowest possible rate.

However, sensors also consume a large amount of energy for frequent sampling. The ratio between the energy consumption for taking one sample and that for transmitting one sample is not as small as we might think. Therefore, ignoring the energy cost for sensing is sub-optimal. As shown in the lower part of Figure 1.5, a novel sampling scheme is to take sparse sensing-samples and reconstruct the missing data using algorithms that exploit the structure available in the measured samples.

When designing the sparse-sampling scheme, we exploit the intra-node correlation, that is, the temporal correlation in the signal to be sensed. In particular, we need to resolve the following three problems: i) learning the temporally-varying signal model by using the measured samples, ii) selecting the sensors for sampling by using the signal model, and iii) reconstructing the original sensing field by using the signal model and the measured samples.

1.3.4 Sparse Sensor-Selection by Exploiting Spatial Correlations

In Section 1.3.3, we raise the problem of sparse-sensor selection by exploiting temporal correlations. We observe that in many sensing fields, the signal to be sensed has also strong inter-node correlations (spatial correlations), especially in participatory sensing systems where the sensor nodes are dense. Therefore, we consider the problem of exploiting this spatial correlation in order to design sparse-sensing schemes.

Most existing works that exploit spatial correlations aim at minimizing the number of sensing samples in a single time instant. In these works, a utility function is defined on the subset of the selected sensors in that time instant. The utility function might be defined as the sensing coverage [41], the mutual information [31], the frame potential [51], or the log determinant of a confidence ellipsoid [38]. Leveraging on the submodular property of these utility functions, greedy algorithms are proposed to select the sparse sensors in a single time instant.

However, in many practical WSNs or participatory sensing systems, we need to select sparse sensors in a continuous sensing period instead of a single time instant. Therefore, the objective is to extend the lifetime of the sensing system that guarantees a certain sensing utility given limited amounts of resources for all sensors. In this problem, the load of sensing has to be fairly distributed onto all sensor nodes, and therefore, the selected subset of sensors has to be adaptively changed. Compared to the sparse-sensing problem in a single time instant, this problem is more complicated due to the increased size of the optimization space. We have to adaptively decide when, how many, and which sensors need to be activated for taking sensing samples.

1.4 Contributions

We now describe the contributions of the thesis. The details of the algorithms and implementations will be discussed in separate chapters.

In **Chapter** 2, we address the problem of active-BS selection in WSNs. The contributions we make in this chapter are as follows:

1. We propose a novel scheme that deploys multiple BSs, keeps only one BS active at a time and adaptively re-selects the active BS. By using this scheme, the temporally and spatially varying energy resources available to all BSs are efficiently utilized. Consequently, the energy supplies of individual BSs can be reduced substantially.
2. We propose an adaptive algorithm HEF for re-selecting the active BS. This algorithm requires little exchange of information in the WSN and it is easy to implement. We show that under certain mild conditions, this algorithm is asymptotically optimal.
3. We discuss the implementation issues of HEF on real WSNs. In particular, we discuss how to start the network, how to gather the needed information and how to hand over the active BS. The solutions we provide are distributed and robust.
4. To evaluate the proposed scheme, we run simulations on the simulator Omnet++/Castalia [47] and real experiments on an outdoor testbed. To the best of our knowledge, ours is the first

installation of a real testbed with multiple cooperative BSs. The obtained results show that our proposed scheme is energy-efficient, has low communication-overhead and reacts rapidly to network changes.

This chapter is based on the paper [69].

In **Chapter** 3, we generalize the active-BS selection problem to the problem of jointly selecting active BSs and routing in WSNs. In this problem, multiple BSs might be active simultaneously and the energy consumption of RSNs are considered, as well as that of BSs. The contributions we make in this chapter are as follows:

1. We analyze the general problem of jointly selecting BSs and routing in WSNs, and we show that this problem is NP-hard. Because the formulation of this problem is general, the analyses we provide can be used in many other networked systems, including super-node selection in peer-to-peer networks and cooperative beamforming in cellular networks.
2. We propose an adaptive algorithm to solve the joint-selection problem. The proposed algorithm is computationally light, it only requires easily available information as input and guarantees, under mild conditions, to yield a lifetime that is at least 62% of the optimal one. We verify the effectiveness of this algorithm through extensive numerical simulations.

This chapter is based on the paper [70].

In **Chapter** 4, we address the sparse-sampling problem by exploiting the temporal correlation among the signal to be sensed. We propose a scheme to adaptively learn temporal model from the measurements and use the model to schedule when to take sensing samples. The main contributions of this chapter are as follows:

1. The proposed method does not impose on-sensor computation nor inter-node communication. Each sensor node simply collects measurements according to a designated sampling pattern and transmits the data to a common server. The server receives all the data from one or multiple sensor nodes and performs signal reconstruction. This is actually in accordance with the setup of distributed source coding [60], where no inter-node communication is used. Hence, the proposed algorithm provides an alternative solution to the distributed coding problem: the communication rate is reduced and the reconstruction error is bounded without using any inter-node communication.
2. The proposed algorithm is tested on different sets of real-word data, outperforming both the traditional sensing schemes and the state-of-the-art sparse-sensing schemes, in terms of reconstruction quality given a fixed amount of measurements.

This chapter is based on the paper [17].

In **Chapter** 5, we address the sparse-sampling problem by exploiting spatial correlations in both static WSNs and participatory sensing systems. The main contributions of this chapter are as follows:

1. We propose a general framework for adaptively selecting sparse sensors, so that the sensing lifetime is extended and in the meantime, a certain level of application-specific utility is always guaranteed.
2. In static WSN scenarios, the proposed algorithm for selecting the sparse sensors guarantees to achieve a lifetime at least $1/\log M$ the optimal one (M denotes the number of all sensors); in participatory sensing systems where sensors are moving, this algorithm is also resource efficient, as we validate that through simulations.

Chapter 2

Active Base-Station Selection in Wireless Sensor Networks

> If you cannot explain it simply, you don't know it well enough.
>
> *Albert Einstein*

In this chapter, we will discuss the first adaptive-selection problem that is our motivation for this thesis: the base-station (BS) selection problem. In the following chapters, we will discuss the generalizations of this problem.

2.1 Introduction

In the past few years, we worked on the Sensorscope Project [34]; the objective was to deploy a WSN on glaciers in the mountains to monitor climate changes. Due to the harsh environment, a BS might fail and the network might split into smaller networks. For the network to be *"robust"*, or in other words, to be able to recover from the aforementioned incidents, we had to install multiple BSs in the sensing field, as many others do [7, 59]. Because of the high energy-consumption of long-range communications, all BSs were required to be equipped with large batteries and large solar panels. This is definitely undesirable because of the increased difficulty of both deployment and maintenance. It is imperative to have a new scheme for coordinating the energy resources available to all the deployed BSs such that the sizes of energy sources for individual BSs can be substantially reduced.

We propose a novel scheme to do just that. The idea is to deploy multiple BSs, to shut down unnecessary BSs and to keep only one BS active, as shown in Figure 1.4. To share the high load for using the long-range communication, we adaptively and iteratively select the BS that is activated. Meanwhile, other BSs remain passive and behave as RSNs: They turn off their long-range communication devices, only sampling and forwarding data by using short-range communication. When the network has connectivity problems and splits into several connected components, the aforementioned active-BS selection process automatically takes place in all these small components. In each connected component, the high energy-consumption of using long-range communication for the active BS is shared among all BSs. The batteries of all BSs form a pool, thus resulting in a larger global power source. To build a sustainable WSN, the requirement is that the total energy harvested by all BSs sustains the consumption of the active BS. Consequently, the size of the individual power sources can be substantially reduced.

Because in this scheme we adaptively select only one active BS, we call it *"virtually moving one BS"*. In this scheme, we have to solve the following practical issues: (i) when the network is connected, how to start the WSN into a state with only one active BS (the bootstrap problem), (ii) how to adaptively gather the information and decide upon the next active BS, (iii) how to manage the handover of the active BS and (iv) how to detect and recover from a network split or from a failure of the active BS. The solutions we provide to these issues are distributed and robust.

In each connected component of the network, we have to adaptively select one active BS. The first idea was to use the Round-Robin (RR) protocol: we let all BSs be sequentially active during an equal time. However, RR is not necessarily optimal because of the the heterogeneity of BSs: (i) The energy recharged from solar panels of different BSs might be different because the solar panels might have different positions, different angles to the sun and different energy conversion efficiency. (ii) The circuit power of different BSs might be different, both when being active or when being passive. To achieve the optimal lifetime, different BSs should be active for different fractions of time, and these fractions cannot be computed beforehand due to the unknown profile of the energy recharging process. We propose an adaptive algorithm that enables all BSs to gradually achieve the optimal fractions of active time, *i.e.*, *"highest energy first"* (HEF). This algorithm adaptively selects the BS with the highest available energy to be active. The appealing feature of HEF is that it requires little information as input and yet fits perfectly for a solution of the WSN optimization problem. The active BS needs to gather only the battery levels of passive BSs. We prove that this algorithm is asymptotically optimal under mild conditions.

To evaluate our proposed scheme, we first run several simulations on the simulator Omnet++/Castalia [47] and next we run real experiments on an outdoor testbed. Simulation results show that HEF is energy efficient, has low communication-overhead and reacts rapidly to network changes. The real experiments lasted for 15 days, and they show that by using HEF to coordinate 3 BSs, the lifetime of the WSN can be prolonged by a factor of 3 to 4. The enhancement will be more pronounced if HEF is used on a larger number of cooperative BSs.

2.2 System Architecture

Our system architecture for coordinating multiple BSs is as follows. In our architecture, time is partitioned into slots whose lengths are two hours each. At the beginning of each time slot, one active BS is selected. This active BS begins broadcasting beacons and notifying the whole network. Upon receiving these beacons, passive BSs and RSNs update their routing tables and forward these beacons. Every sensor node takes sensing samples at a constant rate. The sensed data are then forwarded to the active BS by using short-range communication in a multi-hop fashion. The active BS collects all the data packets and forwards them to the remote server. In the next time slot, the active BS remains the same or it hands over the role to its successor, depending on the output result of HEF. Then, the new active BS starts broadcasting the beacons and the whole process is repeated.

In this architecture, we have to address the following problems: (i) how the network starts into the state with only one active BS, (ii) how the active BS gathers the information needed for the selections, (iii) how the active BS hands over the role to the selected successor, and (iv) how the network recovers from unexpected failures. Before discussing these issues, we first briefly review some details of our system.

2.2.1 Network Details

We will show how the network manages synchronization, MAC protocols, routing protocols and the usage of GSM/GPRS. The interested reader can refer to the previously published work for more details [34].

Synchronization

All sensor nodes are synchronized on *universal coordinated time* (UTC), retrieved by the active BSs when they connect to our server. The current time T_c is inserted into beacons through MAC time-stamping [26]. To estimate T_c, we use the crystal of sensor nodes to compute the elapsed time since the last update of UTC. This mechanism, although simplistic, enables a synchronization in the order of one millisecond, which is sufficient in our application.

MAC protocols

In the MAC layer, we adopt the commonly used T-MAC [58]. With T-MAC, sensor nodes dynamically adjust their duty cycles, based on the communication loads.

Routing protocols

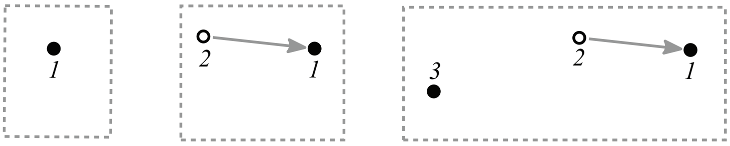
We use gradient routing where sensor nodes send the data packets to their neighbors who have the shortest hop-distances to the active BS. We also make a few modifications on the classic gradient-routing protocol, so that control messages for updating the active BS are handled by a specific procedure, as will be discussed later.

GSM/GPRS usage

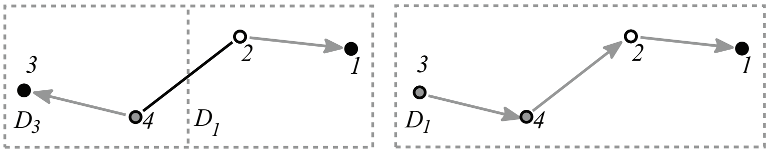
As the GSM/GPRS chip is an energy-hungry device (two orders of magnitude more than the short-range radio transceiver), its connection to the server is duty-cycled. There is an obvious trade-off between real-time information and energy savings. The typical connection interval that we use is 5 minutes.

2.2.2 Starting the Network

In our architecture, starting the network is a bit more complex than in a traditional WSN. Multiple BSs have to make a consensus on the one that should be the only active BS. We give a



(a) Step 1 (b) Step 2 (c) Step 3



(d) Step 4 (e) Step 5

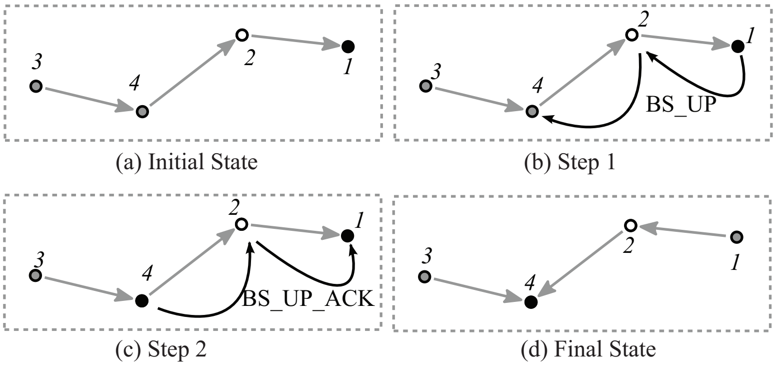
Figure 2.1: Starting the network.

de-centralized solution to this problem.

Once a BS is booted, it is passive and listens for beacons from other sensor nodes. If, after some timeout, it still has not heard any beacon, it becomes active and begins broadcasting beacons. Other sensor nodes receive the beacons and know their hop distances to the active BSs. Because sensor nodes use the gradient routing protocol, they join the nearest active BS. Notice that there might be several active BSs co-existing at this stage. The whole network is virtually split into several clusters, where each cluster has one active BS.

Then, the network automatically merges these clusters in a de-centralized way. For simplicity of discussion, we assume that the network has only two clusters B_i and B_j with the active BSs b_i and b_j, respectively. There are obviously some nodes on the boundaries, belonging to one cluster and having neighbors belonging to the other one. These nodes can detect the presence of the two active BSs due to the beacon messages, as those belonging to B_i will eventually hear about b_j from their neighbors belonging to B_j. When these nodes detect the presence of the two active BSs, it is their duty to fix the problem. For the sake of simplicity, we decide that the active BS with the smaller identifier should be kept active. Assuming $i < j$, the boundary nodes belonging to B_j would thus send a BS_DOWN message to b_j, asking it to become passive in favor of b_i. Upon reception of this request, the BS b_j stops sending beacons and becomes passive. As a result, routes to b_j in the cluster B_j gradually disappear while routes to b_i propagate from B_i to B_j. When the process is over, the cluster B_j has been merged with B_i, resulting in only one cluster. This merging process is also applicable when multiple clusters are present.

Figure 2.1 provides an example showing the whole starting process. At the first step, BS 1 is started. As it cannot hear from any other sensor node, it becomes active, gathering its own data and sending them to the server. Then, a RSN 2 is started. It detects the active BS 1 and joins it to form a two-node network. At step three, BS 3 is started. It is too far away to hear from BS 1 and RSN 2, so it becomes active. At step four, another BS 4 is added. It hears from both active BS 3 and RSN 2, and it decides to join BS 3 rather than the small network $\{1, 2\}$ due to the shorter routing paths. Hence, there are two clusters: $B_1 = \{1, 2\}$ and $B_3 = \{3, 4\}$. The boundary nodes are RSN 2 and BS 4, and they advertise about active BSs 1 and 3. When RSN 2 hears about BS 3, it does nothing as RSN 1, its active BS, has a lower identifier than BS 3. BS 4, however, sends a BS_DOWN message to BS 3. Once BS 3 becomes a passive BS and stops sending beacons, the route from BS 4 to BS 3 breaks, so that at some point, BS 4 joins



(a) Initial State (b) Step 1

(c) Step 2 (d) Final State

Figure 2.2: The handover process.

B_1, as well as BS 3 later on, resulting in the final state of Figure 2.1.

2.2.3 Gathering the Information for Adaptive Selections

Before adaptively selecting the active BSs, the network needs to learn about the existence of other passive BSs and their battery levels. For this purpose, we use a specific type of message, called BS_ADVERT. The BS_ADVERT messages are periodically generated by passive BSs, and then routed to the active BS, like any other data message using gradient routing. The BS_ADVERT messages are specifically handled. All sensor nodes include their own IDs in the packet, when forwarding the BS_ADVERT messages. When the active BS receives the BS_ADVERT messages, it knows exactly the paths that these messages have traveled through. By reverting these paths contained in the BS_ADVERT messages, the active BS stores a *handover table* that is used when sending notifications to the next active BSs. This mechanism is well-known in ad hoc networks (*e.g.*, dynamic source routing [36]) and is sometimes called *piggybacking*. The active BS also maintains a list of battery levels for all BSs. When the active BS receives a BS_ADVERT message, it updates the corresponding elements in the list or table if this message contains newer timestamps.

2.2.4 Handing Over the Active BS

Knowing the locations of all passive BSs and their battery levels, the active BS will decide the next active BS based on the algorithm that will be described in Section 2.4. If the active BS decides to hand over its role to another BS, it will send out a BS_UP message to notify its successor. This BS_UP message contains the routing information from the handover table. Once a RSN receives a BS_UP message, it forwards the message if it is on the route, and otherwise it drops the message. When a BS receives the BS_UP message, it checks whether it is the destination of the BS_UP message. If it is, this BS sends back a BS_UP_ACK message to the currently active BS and becomes active by advertising its status through the beacon messages. The previously active BS, upon reception of a BS_UP_ACK, becomes passive and stops its beacons. In the case where no BS_UP_ACK is received (*e.g.*, node unreachable), the active BS tries again with the next best candidate. This process continues until a suitable candidate takes over the active role.

The whole process of executing the handover decision is illustrated in Figure 2.2. Initially, BS 1 is active. It selects BS 4 as its successor. During step 1, a BS_UP message is routed from BS 1 to BS 4 to inform BS 4 the decision made by BS 1. At step 2, BS 4 receives the BS_UP message and becomes active. At the same time, it sends back a BS_UP_ACK message to BS 1. Finally, BS 1 becomes passive and BS 4 is the only active BS.

2.2.5 Recovering from Failures

In a sensor network, the active BS might fail and the network might split into small connected components. With our architecture, the network automatically recovers from these incidents. When the active BS fails, it either reboots or stops working; both cases lead to the disappearance of active BS beacons. Should this happen, all routes in the network would disappear, and one or multiple BSs would eventually decide to become active, just like during the starting process. If multiples of them become active, the merging process would apply, eventually leading to only one active BS. When the network splits into small components, the passive BSs within each component are able to detect the disappearance of beacons from the active BS in this component. Then, the bootstrapping process mentioned in Section 2.2.2 will ensure that there will eventually be one active BS in each small component.

2.3 Adaptive BS-Selection Problem Formulation

On top of the system architecture we discussed in Section 2.2, we now consider the problem of optimally selecting the active BS, so that the energy resources on all BSs are efficiently used. We only consider the scenario where the network is fully connected. If the network splits into small components, as we have seen in Section 2.2.5, the problem is the same within each small component.

Consider a set of BSs V_b deployed in the sensing field. Time is discretized into slots $n \in \mathbb{N}^+$, and we denote the length of a time slot by τ.

Decision vector: As the active BS is adaptively selected in different time slots, we let $v_m^{(n)}$ indicate whether BS $m \in V_b$ is active at a given time slot $n \in \mathbb{N}^+$, *i.e.*,

$$v_m^{(n)} = \mathbb{I}\left(\text{BS } m \text{ is active at the } n\text{-th time slot}\right),$$

where $\mathbb{I}(A)$ denotes the indicator function: $\mathbb{I}(A) = 1$ if argument A is true and $\mathbb{I}(A) = 0$ otherwise. Collect all $v_m^{(n)}$, $m \in V_b$, in a $|V_b| \times 1$ column vector $\boldsymbol{v}^{(n)} = \left[v_1^{(n)}\ v_2^{(n)}\ \cdots\ v_{|V_b|}^{(n)}\right]^\top$ with $\top$ denoting transposition. Call $\boldsymbol{v}^{(n)}$ the *decision vector* at the n-th time slot. Because only one BS is active at one time slot, $\boldsymbol{v}^{(n)}$ has $|V_b| - 1$ zero entries and one entry equal to 1. We denote the sequence of decision vectors up to time slot n by $\mathcal{V}^{(n)} = \left\{\boldsymbol{v}^{(t)}\right\}_{t=1}^{n}$.

Cost matrix: The energy consumption of BSs is due to three sources: sensing, short-range communication, and long-range communication. We assume that the sensing costs are negligible. Let the MAC protocol and routing protocol of the WSN be predefined. Therefore, when a specific BS is selected to be active, both the energy consumption from short-range communication and from long-range communication of each BS is deterministic. Denote by C_{ml} the energy-consumption rate of BS m ($m \in V_b$) when BS l ($l \in V_b$) is active. We group all energy-consumption rates in a $|V_b| \times |V_b|$ matrix $\boldsymbol{C}$, which we call the *cost matrix*. If we neglect the energy consumption from short-range communication, the passive BSs do not consume any energy, and therefore the cost matrix becomes diagonal. In practice, the ratio between the energy consumption from long-range communication and that from short-range communication might be a factor of 5 ∼ 20, based on different settings of the network.

Available energy: We denote the remaining amount of energy of BS $m \in V_b$ at the end of time slot n by $e_m^{(n)}$ and we call it *available energy*. We gather the available energy of all BSs in a vector $\boldsymbol{e}^{(n)} = \left[e_1^{(n)}\ e_2^{(n)}\ \cdots\ e_{|V_b|}^{(n)}\right]^\top$. In practice, the available energy is lower-bounded by zero and upper-bounded by the storage capacity. For simplicity, however, we assume that it is not upper-bounded. Without loss of generality, we assume that all BSs have the same available energy $e \in \mathbb{R}^+$ initially, with $\boldsymbol{e}^{(0)} = e\boldsymbol{u}$, where $\boldsymbol{u} = [1\,1\,\cdots\,1]^\top$ is a $|V_b| \times 1$ all-ones vector.

Energy-recharge rates: At each time slot $n \in \mathbb{N}^+$, each BS $m \in V_b$ receives a certain amount of incoming energy. Denote the average rate of incoming energy at this time slot by $s_m^{(n)}$ and call it the *energy-recharge rate*. We group all the energy-recharge rates at the n-th time slot into a vector $\boldsymbol{s}^{(n)} = \left[s_1^{(n)}\ s_2^{(n)}\ \cdots\ s_{|V_b|}^{(n)}\right]^\top$. We denote the sequence of energy-recharge rates up to time slot n by $\mathcal{S}^{(n)} = \left\{\boldsymbol{s}^{(t)}\right\}_{t=1}^{n}$. In particular, $\mathcal{S}^{(\infty)}$ denotes the sequence of energy-recharge rates over an infinite time horizon. We make the following realistic assumptions on $\mathcal{S}^{(\infty)}$:

— D1:

$$\mathbb{E}\left(\boldsymbol{s}^{(n)} \mid \mathcal{S}^{(n-1)}\right) = \bar{\boldsymbol{s}}, \forall n \in \mathbb{N}^+, \tag{2.1}$$

where $\bar{\boldsymbol{s}}$ is a constant vector and $\mathbb{E}\left(\cdot \mid \mathcal{S}^{(n-1)}\right)$ denotes the expectation conditioned on the sequence $\mathcal{S}^{(n-1)}$. Let $\bar{s}_m$ be the m-th element of $\bar{\boldsymbol{s}}$. This assumption is weaker than assuming $\mathcal{S}^{(\infty)}$ is an i.i.d process.

— D2:

$$\left\|\boldsymbol{s}^{(n)}\right\|_\infty \leq S, \forall n \in \mathbb{N}^+. \tag{2.2}$$

where S is a constant with $0 \leq S < +\infty$.

Condition D1 amounts to saying that the energy-recharge rates are always constant conditioned on the previous information. Condition D2 states that the energy-recharge rates are bounded by a constant S.

Relations among the aforementioned parameters: At the n-th time slot, the amounts of energy recharged for all BSs are given by $\tau\boldsymbol{s}^{(n)}$ and the amounts of energy consumed are given by $\tau\boldsymbol{C}\boldsymbol{v}^{(n)}$. Therefore, the available energy evolves according to

$$\boldsymbol{e}^{(n)} = \boldsymbol{e}^{(n-1)} + \tau\boldsymbol{s}^{(n)} - \tau\boldsymbol{C}\boldsymbol{v}^{(n)}. \tag{2.3}$$

If we sum up the iterative equation (2.3) from time 0 to time n and use $\boldsymbol{e}^{(0)} = e\boldsymbol{u}$, we have

$$\boldsymbol{e}^{(n)} = e\boldsymbol{u} + \tau\sum_{t=1}^{n}\boldsymbol{s}^{(t)} - \tau\boldsymbol{C}\sum_{t=1}^{n}\boldsymbol{v}^{(t)}. \tag{2.4}$$

Adaptive BS-selection problem: Denote by N the maximum number of time slots before the first BS drains out of energy. The lifetime of the network is therefore τN. If the realization of $\mathcal{S}^{(\infty)}$ is already known to us, the goal is to schedule the selections of active BSs, such that τN is maximized. In other words, we want to find the longest sequence of decision vectors $\mathcal{V}^{(N)}$ such that for any $1 \leq n \leq N$, the available energy $\boldsymbol{e}^{(n)} \geq \boldsymbol{0}$.[2] Therefore, we formulate the problem

---

[2] Without special mentioning, the inequalities between vectors are all component-wise.

as an optimization problem

$$\begin{aligned}
\max_{\mathcal{V}^{(N)}} \quad & N \\
\text{s.t.} \quad & \tau \boldsymbol{C} \sum_{t=1}^{n} \boldsymbol{v}^{(t)} \leq e\boldsymbol{u} + \tau \sum_{t=1}^{n} \boldsymbol{s}^{(t)}, \forall 1 \leq n \leq N, \\
& \boldsymbol{u}^{\top} \boldsymbol{v}^{(n)} = 1, \ \forall 1 \leq n \leq N, \\
& \boldsymbol{v}^{(n)} \in \{0,1\}^{|V_b|}, \ \forall 1 \leq n \leq N,
\end{aligned} \tag{2.5}$$

where the first constraint follows from (2.4) and the constraint that $\boldsymbol{e}^{(n)} \geq \boldsymbol{0}, \forall 1 \leq n \leq N$.

We denote the optimal objective value of problem (2.5) by N_{opt}. We denote the offline scheduling algorithm that optimizes (2.5) by OPT. We will use it for comparison in Section 2.5. We note that (i) problem (2.5) is not a standard optimization problem because the number of constraints depends on the objective value N, and that (ii) the optimal objective value N_{opt} depends on the realization of the stochastic process $\mathcal{S}^{(\infty)}$. In the following, we will analyze the performance of the optimal objective value N_{opt} via an auxiliary optimization problem.

Denote the fraction of active time of BS $m \in V_b$ by $\bar{v}_m$. Group these fractions into a vector $\bar{\boldsymbol{v}} = \begin{bmatrix} \bar{v}_1 \ \bar{v}_2 \ \cdots \ \bar{v}_{|V_b|} \end{bmatrix}^{\top}$. Notice that we have $\boldsymbol{u}^{\top}\bar{\boldsymbol{v}} = 1$. Let

$$\boldsymbol{R} = \boldsymbol{C} - \bar{\boldsymbol{s}}\boldsymbol{u}^{\top}. \tag{2.6}$$

where $\bar{\boldsymbol{s}}$ is defined by (2.1). If we select active BSs according to the fractions of active time $\bar{\boldsymbol{v}}$, the expected energy-decrease rates of all BSs are $\boldsymbol{C}\bar{\boldsymbol{v}} - \bar{\boldsymbol{s}}$, which are equivalent to $\boldsymbol{R}\bar{\boldsymbol{v}}$ because of (2.6) and $\boldsymbol{u}^{\top}\bar{\boldsymbol{v}} = 1$. Because the lifetime of the network is decided by the maximum energy-decrease rate among all BSs, maximizing the lifetime amounts to minimizing the maximum energy-decrease rate. Therefore, to analyze the asymptotic property of the optimal lifetime τN_{opt}, we define the auxiliary optimization problem

$$\begin{aligned}
\min_{\bar{\boldsymbol{v}}, f} \quad & f \\
\text{s.t.} \quad & \boldsymbol{R}\bar{\boldsymbol{v}} \leq f\boldsymbol{u}, \\
& \boldsymbol{u}^{\top}\bar{\boldsymbol{v}} = 1, \\
& \bar{\boldsymbol{v}} \geq \boldsymbol{0},
\end{aligned} \tag{2.7}$$

where f denotes the maximum energy-decrease rate among all sensors. The optimal solution of problem (2.7) is denoted by $(\bar{\boldsymbol{v}}^*, f^*)$.

In the following, we will show the relation between the optimal objective value N_{opt} of problem (2.5) and the optimal objective value f^* of problem (2.7) under assumptions D1 and D2: (i) If $f^* < 0$, by selecting the active BSs according to the optimal fractions $\bar{\boldsymbol{v}}^*$, the available energy of all BSs has a tendency to increase with time. For any given e, there is a probability that the optimal lifetime N_{hef} is infinite, and this probability becomes arbitrarily close to 1 when e becomes large; and (ii) if $f^* > 0$, any scheduling algorithm will almost surely result in a finite lifetime. By selecting the active BSs according to the optimal fractions $\bar{\boldsymbol{v}}^*$, there is a high probability that the optimal lifetime is within the range $[(1-\delta)e/\tau f^*, (1+\delta)e/\tau f^*]$, for any $\delta > 0$. This probability becomes arbitrarily close to 1 when e becomes large. The arguments above are summarized in Theorem 2.1.

Theorem 2.1

If assumptions D1 and D2 on the energy-recharge rates $\mathcal{S}^{(\infty)}$ hold, the optimal objective value N_{opt} of problem (2.5) has the following asymptotic performance:

— *when $f^* < 0$,*

$$\lim_{e\to\infty} \mathbb{P}\left(N_{\text{opt}} = \infty\right) = 1, \tag{2.8}$$

— *when $f^* > 0$,*

$$\lim_{e\to\infty} \mathbb{P}\left(\left|\frac{N_{\text{opt}}}{e/(\tau f^*)} - 1\right| < \delta\right) = 1, \forall\delta > 0. \tag{2.9}$$

The detailed proof can be found in Appendix 2.A.2, which we briefly sketch here. In the simple deterministic scenario where the energy-recharge rates $\boldsymbol{s}^{(n)} = \bar{\boldsymbol{s}}$ for any $n \in \mathbb{N}$, we can easily show that: given that $f^* < 0$, if e is sufficiently large, $N_{\text{opt}} = \infty$; and given that $f^* > 0$, if e is sufficiently large, N_{opt} is deterministically within the range $[(1-\delta)e/\tau f^*, (1+\delta)e/\tau f^*]$, for any $\delta > 0$. Then, in the stochastic scenario, we rely on the assumptions D1 and D2 to relate it to the deterministic scenario. Notice that the energy recharged in the first n time slots in the deterministic scenario is $n\bar{\boldsymbol{s}}$ and that in the stochastic scenario is $\sum_{t=1}^{n} \boldsymbol{s}^{(t)}$. We show that their difference $\sum_{t=1}^{n} \boldsymbol{s}^{(t)} - \bar{\boldsymbol{s}}$ is a martingale with bounded difference. We use the Azuma-Hoeffding inequality for martingales [30, p. 476] to show that the probability distribution of the distance from $\sum_{t=1}^{n} \boldsymbol{s}^{(t)} - \bar{\boldsymbol{s}}$ to the zero vector decays exponentially. Using this result, we will show that when $e \to \infty$, the optimal lifetime τN_{opt} in the stochastic scenario converges in probability to that in the simple deterministic scenario.

Solving (2.5) or (2.7) is however infeasible in practice for the following reasons: (i) Measuring the cost matrix $\boldsymbol{C}$ requires expensive equipments such as high-frequency data loggers, and (ii) estimating the energy-recharge rates $\mathcal{S}^{(n)}$ is hard, because they depend on too many factors. For example, the energy-recharge rate from a solar panel might depend on its location, the angle of its surface to the sunlight, its energy conversion efficiency, and the weather. In a real WSN, the only easy-to-capture information is the battery level, which can be used as an indicator of the available energy. In the following, we will discuss an algorithm for selecting the active BS; it only uses information on available energy as input.

2.4 The “Highest Energy First” (HEF) Algorithm

In this section, we propose the algorithm *“highest energy first”* (HEF) for solving the adaptive BS-selection problem. In practice, this algorithm is easy to implement because it only requires the battery levels of all BSs as the input.

The procedure of running HEF is summarized in Algorithm 2.1. At any time slot n, BS $m^* \in V_b$) is chosen to be active at the n-th time slot if and only if its available energy $e_{m^*}^{(n-1)}$ is the highest, *i.e.*,

$$v_{m^*}^{(n)} = \mathbb{I}\Big(e_{m^*}^{(n-1)} \geq e_{m}^{(n-1)},\ \forall 1 \leq m \neq m^* \leq |V_b|\Big), \tag{2.10}$$

with ties broken uniformly at random.

Let N_{hef} be the maximum number of time slots before the first BS drains out of energy using the HEF scheduling algorithm, that is,

$$N_{\text{hef}} = \inf\{\{\infty\} \cup \{n \mid \exists l^* \in V_b, e_{l^*}^{(n+1)} < 0\}\}.$$

Algorithm 2.1: The "Highest Energy First" Algorithm

```
Input: The initially available energy $\boldsymbol{e}^{(0)}$, the sequence of energy-recharge rates $\mathcal{S}^{(\infty)}$
Output: $N_{\text{hef}}$, $\mathcal{V}^{(N_{\text{hef}})}$
Set $n \longleftarrow 0$
while $\boldsymbol{e}^{(n)} \geq \boldsymbol{0}$ do
    Find $m^*$ such that $e_{m^*}^{(n-1)} \geq e_m^{(n-1)}, \forall 1 \leq m \neq m^* \leq |V_b|$
    Set $\boldsymbol{v}^{(n)}$ where $v_{m^*}^{(n)} \leftarrow 1$ and $v_m^{(n)} \leftarrow 0$, for any $m \neq m^*$
    Update $\boldsymbol{e}^{(n)} = \boldsymbol{e}^{(n-1)} - \tau \boldsymbol{C}\boldsymbol{v}^{(n)} + \tau \boldsymbol{s}^{(n)}$, and $n = n + 1$
Set $N_{\text{hef}} \longleftarrow n - 1$ and $\mathcal{V}^{(N_{\text{hef}})} = \{\boldsymbol{v}^{(t)}\}_{t=1}^{N_{\text{hef}}}$
```

The HEF algorithm is a heuristic algorithm, yet we will show that it is asymptotically optimal under mild conditions. We use the optimal objective value f^* of problem (2.7) as a link between N_{hef} and N_{opt}: (i) If $f^* < 0$, for any large constant K, there is a high probability that $N_{\text{hef}} > Ke$ when the initial available energy e is large. This probability converges to 1 when $e \to \infty$. This result is a bit weaker than that $\lim_{e \to \infty} \mathbb{P}(N_{\text{hef}} = \infty) = 1$ as in (2.8). (ii) If $f^* > 0$, when e is large, there is a high probability that N_{hef} is within the range $[(1-\delta)e/\tau f^*, (1+\delta)e/\tau f^*]$, for any $\delta > 0$. This probability converges to 1 when $e \to \infty$. We summarize the arguments above in Theorem 2.2.

Theorem 2.2

If assumptions D1 and D2 on the energy-recharge rates $\mathcal{S}^{(\infty)}$ hold, and if in addition

— *D3:* $R_{ij} = C_{ij} - \bar{s}_i < 0, \forall 1 \leq i \neq j \leq |V_b|$, *and*

— *D4:* $(\boldsymbol{C}^\top)^{-1}\boldsymbol{u} > \boldsymbol{0}$,

then

— *when* $f^* < 0$,

$$\forall K, \lim_{e \to \infty} \mathbb{P}\left(N_{\text{hef}} > Ke\right) = 1, \tag{2.11}$$

— *when* $f^* > 0$,

$$\lim_{e \to \infty} \mathbb{P}\left(\left|\frac{N_{\text{hef}}}{e/(\tau f^*)} - 1\right| < \delta\right) = 1, \forall \delta > 0. \tag{2.12}$$

We interpret conditions D3 and D4 in Theorem 2.2 as follows: (i) Condition D3 states that for any passive BS, the expected energy-recharge rate is larger than the energy-consumption rate, regardless of the selection of the active BS. (ii) Condition D4 is satisfied when energy-consumption rates of active BSs (diagonal elements of $\boldsymbol{C}$) are much larger than the differences among the energy-consumption rates of all passive BSs (differences among non-diagonal elements of $\boldsymbol{C}$). Indeed, we define $c_{\text{pb}} = \min_{1 \leq i \neq j \leq |V_b|} C_{ij}$ and decompose $\boldsymbol{C}$ as $\boldsymbol{C} = \boldsymbol{\Lambda} + c_{\text{pb}}\boldsymbol{u}\boldsymbol{u}^\top$. Then, the diagonal elements of $\boldsymbol{\Lambda}$ are much larger than the non-diagonal elements. It follows that $\boldsymbol{\Lambda}$ is near diagonal and therefore $(\boldsymbol{\Lambda}^\top)^{-1}\boldsymbol{u} > \boldsymbol{0}$. Using the Sherman-Woodbury-Morrison identity [3], we see that

$$(\boldsymbol{C}^\top)^{-1}\boldsymbol{u} = (\boldsymbol{\Lambda}^\top)^{-1}\boldsymbol{u}/(1 + c_{\text{pb}}\boldsymbol{u}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{u}) > \boldsymbol{0}.$$

---

[3] The Sherman-Woodbury-Morrison identity states that for any matrix $\boldsymbol{A}$ and for any two vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$, if $1 + \boldsymbol{w}_2^\top\boldsymbol{A}^{-1}\boldsymbol{w}_1 \neq 0$, we have $(\boldsymbol{A} + \boldsymbol{w}_1\boldsymbol{w}_2^\top)^{-1} = \boldsymbol{A}^{-1} - (\boldsymbol{A}^{-1}\boldsymbol{w}_1\boldsymbol{w}_2^\top\boldsymbol{A}^{-1})/(1 + \boldsymbol{w}_2^\top\boldsymbol{A}^{-1}\boldsymbol{w}_1)$.

Table 2.1: Simulation settings

Sensing field	200 m × 200 m
Sensor node positions	uniformly at random
Radio layer model	XE1205 chip, unit-disk model, the transmitting range is 40 m
Radio energy-consumptions in TX\RX\Sleep mode	79.45\46\1.4 mW
Data-generating rate	1 packet/sec
Control message rate	1 packet/5 min
GSM/GPRS connection rate	once/5 minutes
Average power consumption of GSM/GPRS per connection	296 mW × 40 sec
Active BS handover interval	every 2 hours
Initial available energy	14400 J
Solar panel	50 cm^2

More justifications of conditions D3 and D4 through simulations are shown in Section 2.5.3.

The detailed proof of theorem 2.2 can be found in Appendix 2.A.3. Here, we sketch the intuition for the proof: (i) First, we show that with condition D3, there is a high probability that all BSs use up their available energy at time $N_{\mathrm{hef}}+1$ when e is large. (ii) Secondly, we show that the event that all BSs use up the energy at time $N_{\mathrm{hef}}+1$ implies that the average decision vector $\sum_{n=1}^{N_{\mathrm{hef}}+1} \boldsymbol{v}^{(n)}/(N_{\mathrm{hef}}+1)$ converges to $\boldsymbol{R}^{-1}\boldsymbol{u}/\boldsymbol{u}^{\top}\boldsymbol{R}^{-1}\boldsymbol{u}$ in probability. Under condition D4, we show that the optimal solution of problem (2.7) is $\bar{\boldsymbol{v}}^{*}=\boldsymbol{R}^{-1}\boldsymbol{u}/\boldsymbol{u}^{\top}\boldsymbol{R}^{-1}\boldsymbol{u}$. (iii) Thirdly, given that the average decision vector converges in probability to the optimal solution $\bar{\boldsymbol{v}}^{*}$ of problem (2.7), we use the Azuma-Hoeffding inequality and deduce that: if $f^{*}<0$, there is a high probability that $N_{\mathrm{hef}}>Ke$; and if $f^{*}>0$, there is a high probability that $N_{\mathrm{hef}}>(1-\delta)e/(\tau f^{*})$. Noticing that $N_{\mathrm{hef}}\leq N_{\mathrm{opt}}$ and (2.8), we conclude the proof.

2.5 Simulations

In this section, we will evaluate the proposed scheme by running several simulations on the simulator Castalia/OMNeT++ [47].

2.5.1 General Settings

The general settings of the simulations are chosen to closely approximate our hardware specifications, as listed in Table 2.1. We simulate a sensor network composed of 5 BSs ($|V_b|=5$) and 35 RSNs; they are distributed uniformly at random in a 200m × 200m sensing field. In the physical layer of all sensor nodes, we simulate the XE1205 radio transceiver, with the transmitting power fixed to 0 dbm. We adopt the ideal unit-disk model for the wireless channel and choose the parameters so that the transmitting range is fixed to 40 m. In the MAC layer, the T-MAC protocol is used. All sensors generate data packets at a rate of 1 packet/sec. The BS_ADVERT message (Section 2.2) is transmitted at a rate of 1 packet/5 minutes. Then, the energy-consumption rates of sensor nodes for using the short-range communications are captured using the built-in modules of the simulator Castalia/OMNeT++. The active BS connects to the remote server with GSM/GPRS every 5 minutes. Because the transmitted data volume during each connection is small, the major part of the energy consumption comes from starting, maintaining and closing the communication. We assume that for each GSM/GPRS connection, the active time and the average power consumption is 40 seconds and 296 mW (we choose these values based on the measurements with a digital oscilloscope). The active BS decides whether to transfer its role every 2 hours, which amounts to $\tau = 2$ hours for each time slot. Each BS is assumed to have a set of AA NiMH rechargeable batteries with an initial energy of 800 mAh × 5 V = 14400 J and a solar panel. We assume that the energy-recharge rate for BS $m \in V_b$ at time slot $n \in \mathbb{N}$ is

$$s_m^{(n)} = \eta_m \gamma_m I_m^{(n)} \Gamma_{\text{default}},$$

where η_m denotes the energy conversion efficiency of the solar panel for BS m, γ_m denotes the coefficient for losses (inverter loss, temperature loss, energy transmission loss, energy conservation loss and low radiation loss), $I_m^{(n)}$ denotes the solar radiation on BS m at time n, and Γ_{default} is the default size of the solar panel. The solar radiations $\{I_m^{(n)}\}_n$ $(m \in V_b)$ we use are real data captured in a Swiss valley during the Sensorscope Project [34]. We set $\Gamma_{\text{default}} = 50\,\text{cm}^2$. For all $m \in V_b$, we let η_m be drawn from $[0.05, 0.15]$ uniformly at random, and we set $\gamma_m = 0.2$. The settings discussed above are default unless other settings are explicitly mentioned.

2.5.2 Performance of Different Algorithms

In the following, we show the performances of four different algorithms for organizing the WSN, *i.e.*, FIXED, *Round-Robin* (RR), OPT and HEF. FIXED denotes the scheme with the active BS fixed to be BS 1. RR denotes the algorithm where all BSs take turns being active and have perfectly identical active times. OPT is the offline optimal scheduling algorithm. It is not applicable in practice and is only used for comparison. From the simulator Castalia/Omnet++, we obtain the energy-consumption rates of all sensor nodes when different BSs are active. We list these energy-consumption rates into the cost matrix $\boldsymbol{C}$. Then, we solve the optimization problem (2.5) and have the optimal selections of active BSs. Finally, HEF is the "highest energy first" algorithm described in Section 2.4. In the following, we will compare the performances of the aforementioned four schemes. To avoid the simulation running an infinitely long time, we restrict the maximum running time to be 2400 time slots (200 days): if a network can sustain 2400 time slots, we consider its lifetime as infinite.

Available energy versus time: First, we show the available energy $\boldsymbol{e}^{(n)}$ during 20 days $(1 \leq n \leq 240)$, when the network runs different algorithms in Figure 2.3. We see that HEF leads $\boldsymbol{e}^{(n)}$ to be a uniform vector despite different energy harvested for different BSs. RR cannot fully utilize all the energy because different BSs can harvest different amount of energy from solar panels. FIXED leads to a fast energy-decrease rate of the only active BS, resulting in an early death of the WSN.

Lifetime versus size of solar panels: In Figure 2.4, we show that the lifetime of the network increases with the size of the solar panel equipped on each sensor node. When the size of the solar panel is large enough, the lifetime becomes infinite. To achieve an infinite lifetime,

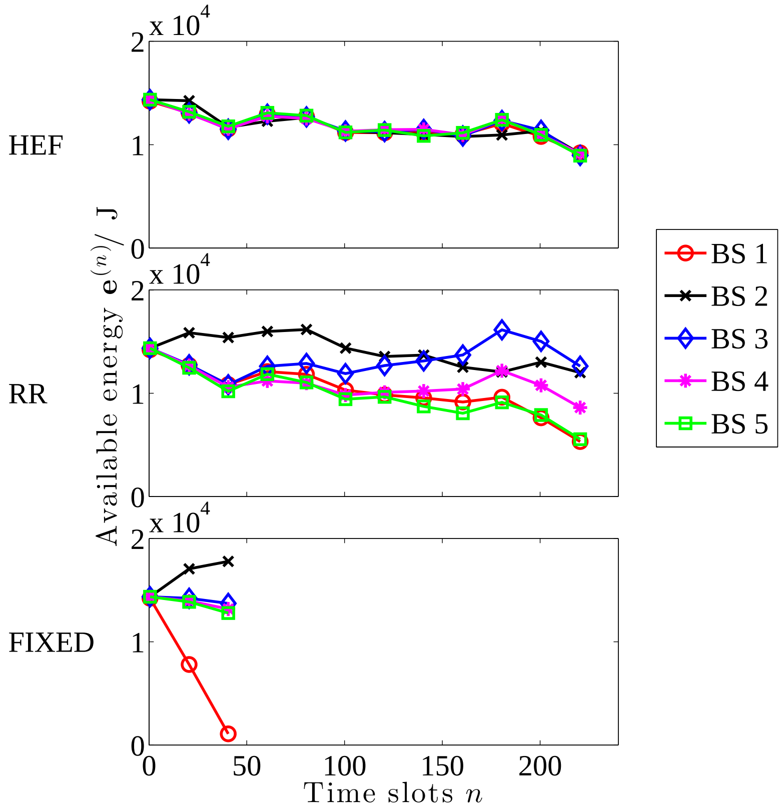


Figure 2.3: The available energy $\boldsymbol{e}^{(n)}$ $(1 \leq n \leq 240)$ when running different algorithms for selecting active BSs. FIXED depletes the battery of the only active BS quickly, thus leading to an early death of the WSN. RR cannot fully utilize all the energy because different BSs can have very different energy recharged from solar panels. HEF equalizes the available energy of all BSs despite different energy harvested on different BSs.

the minimum sizes of solar panels in a network running HEF, RR and FIXED are 62.5cm^2, 112.5cm^2 and 187.5cm^2, respectively. The lifetime of HEF is always better than that of RR and FIXED, and is close to that of OPT.

Lifetime versus initial available energy: In Figure 2.5, we show how the lifetime changes when the sensor network is given different amounts of initial available energy e. Here all solar panels have the default size 50cm^2, which is not sufficient for the network to have an infinite lifetime when running any algorithm. We see that in this scenario, the lifetime of both OPT and HEF increases linearly with the initial available energy, as indicated by the arguments used to prove Theorem 2.1 and 2.2 when $f^* > 0$. HEF is close to OPT and is better than RR and FIXED.

Lifetime versus number of BSs: In Figure 2.6, we show how the lifetime changes when the sensor network has a different number of BSs $|V_b|$. We see that when running HEF or RR, the lifetime increases with the number of BSs. This is because a large number of installed BSs will average out the high cost of being the active BS. On the contrary, the lifetime of FIXED

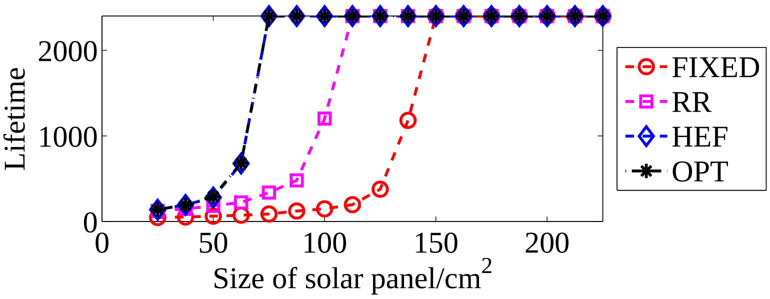


Figure 2.4: Lifetime versus size of the solar panels. The minimum sizes of solar panels to achieve an infinite lifetime in a network running HEF, RR and FIXED are 62.5cm^2, 112.5cm^2 and 187.5cm^2, respectively.

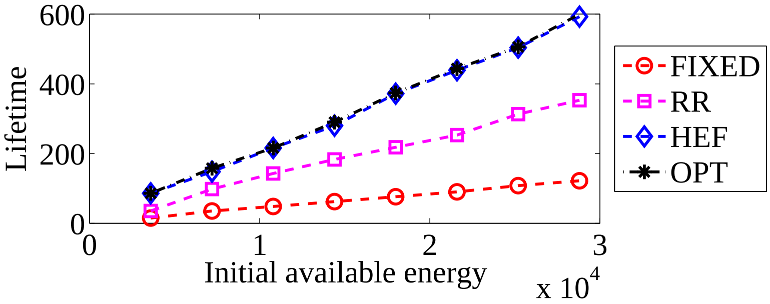


Figure 2.5: Lifetime versus initial available energy. We see that the lifetime of HEF and OPT increases linearly with the amount of the initial available energy. HEF is always close to OPT and is better than RR and FIXED.

remains constant when the number of BSs increases because the burden of using long-range communication is not shared among all BSs. From Figure 2.6, we see that the number of BSs needed to sustain an infinite lifetime required by HEF and RR are 9 and 18, respectively.

To sum up, HEF is more energy efficient than RR and FIXED, and it is very close to OPT in all simulated scenarios. We list the results in the second column of Table 2.2.

Communication overhead: Figure 2.7 shows the overall number of packets transmitted per hour through short-range communication when the network runs different algorithms. FIXED only transmits data packets and does not need to exchange any other control messages. It serves as a baseline in the comparisons. HEF has additional packet exchanges of BS_ADVERT, BS_UP and BS_UP_ACK messages. Because these messages are sent at low rates, *e.g.*, 1 packet per 5 minutes for BS_ADVERT and 1 packet every 2 hours for BS_UP and BS_UP_ACK, the communication overhead of HEF is almost negligible. The communication overhead of RR is the same as HEF because they have the same amount of control messages. We summarize the result in the fourth column of Table 2.2.

Reactions to network changes: We consider the following two incidents: (i) The active BS fails at time slot $n = 120$, and (ii) the network suddenly experiences a connectivity problem and splits into two components (one component contains BS 1 and BS 2 and the other component

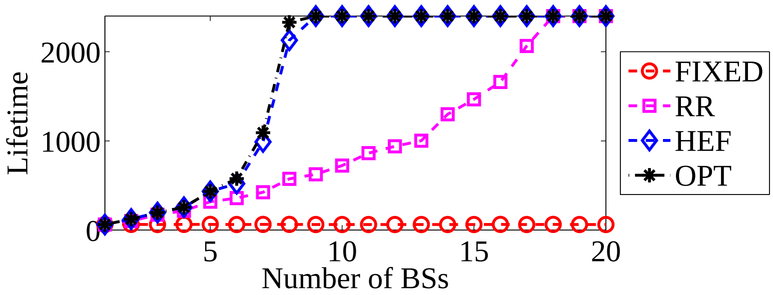


Figure 2.6: Lifetime versus number of BSs. We see that the lifetime when running HEF or RR increases with the number of BSs. The number of BSs to sustain an infinite lifetime required by HEF and RR are 9 and 18, respectively. When running FIXED, larger number of BSs does not result in longer lifetime because the burden is not shared among all BSs.

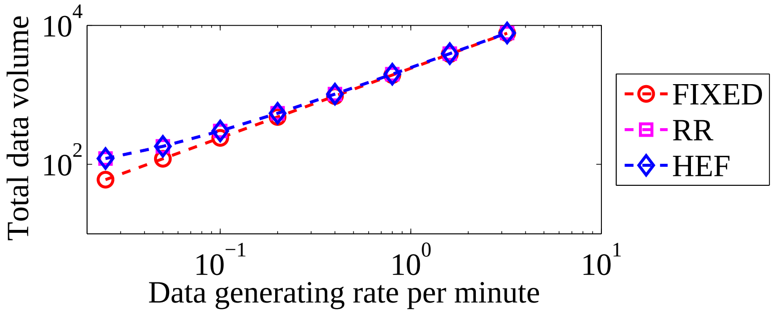


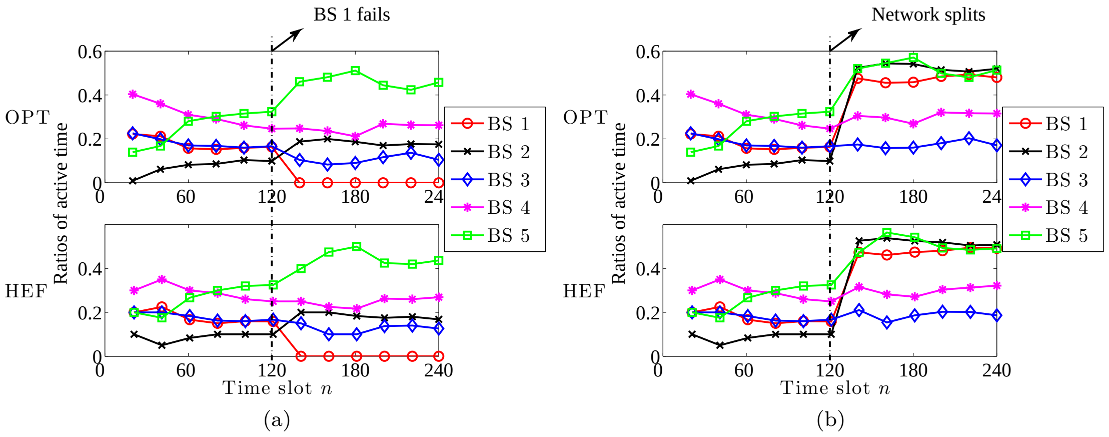
Figure 2.7: Overall number of packets transmitted per hour versus the sensing rate of each sensor node. We see that the communication overheads of both RR and HEF are very small.

contains BS 3, BS 4 and BS 5) at time $n = 120$. Due to the proposed architecture in Section 2.2.5, RR and HEF are robust to the aforementioned incidents, and FIXED is not. We record the "robustness" of all these three schemes in Table 2.2. If we run the RR algorithm, the remaining BSs will have the same active time, which is not necessarily optimal. In Figure 2.8, we show the ratios of active time for all BSs in both scenarios. We see that the performance of HEF is always close to that of OPT, before and after the network changes. Consequenlty, this shows that HEF reacts rapidly to network changes.

2.5.3 Validations of Optimality Conditions

In Theorem 2.2, we need conditions D3 and D4 to ensure the asymptotic optimality of HEF. In the following, we test the validity of these conditions.

Condition D3 requires that for any passive BS, the expected energy gain from the solar panel be larger than the energy consumption regardless of the selection of the active BS. It equals that the sizes of solar panels are large enough to support the operations for any passive BSs. To validate condition D3, we generate 50 sensor networks with the sensor nodes distributed in the sensing field uniformly at random. In Figure 2.9, we show the average of the required sizes of solar panels in all these generated random networks under different data-generating rates. Confidence



(a) (b)

Figure 2.8: The reactions to network changes when running HEF. We use the ratios of active time for all BSs as a metric. Figure 2.8a shows the scenario where BS 1 fails at time slot $n = 120$. Figure 2.8b shows the scenario where the network splits into two small components (one component has BS 1 and BS 2 and the other component has BS 3, BS 4 and BS 5) at time slot $n = 120$. We see that in both scenarios, HEF reacts rapidly to network changes and always closely follows OPT.

Table 2.2: Comparisons of Different Algorithms

Algorithms	Energy efficiency	Robustness	Overhead
FIXED	low	no	none
RR	medium	yes	low
HEF	high	yes	low
OPT	high	-	-

intervals of 95% are used. We see that condition D3 is easily satisfied: equipping all BSs with a 50 cm^2 solar panel is enough, when the data-generating rates are less than 60 packets/min.

Condition D4 requires that the energy-consumption rates of active BSs be much larger than the differences of energy-consumption rates among all passive BSs. The energy-consumption rates of active BSs depend mainly on the time interval between every two GPRS connections. The larger the GPRS connection interval is , the smaller the energy-consumption rates of active BSs are. In Figure 2.10, we randomly generate 50 sensor networks and test the validity of condition D4 under different GPRS connection intervals. We define the *condition fulfilled ratio* (CFR) as the fractions of instances that the generated sensor network fulfils condition D4. We see that condition D4 is always satisfied with a GPRS connection interval less than 20 minutes.

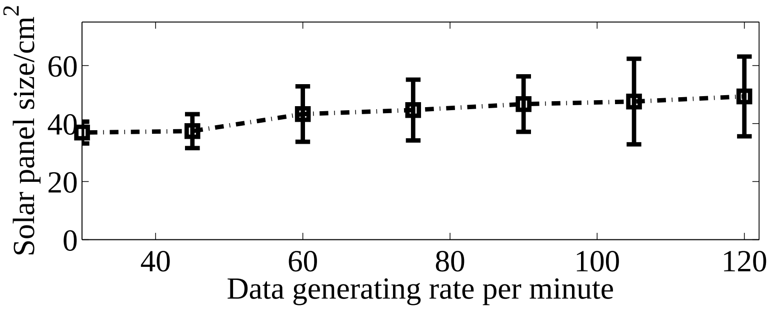


Figure 2.9: The minimum size of solar panels required by condition D3 in Theorem 2.2 under different data-generating rates. Confidence intervals of 95% are used. We see that the required size of solar panels slightly increases with the data-generating rate. Equipping all BSs with a 50 cm^2 solar panel is sufficient to satisfy condition D3 with a data-generating rate at 60 packets/min.

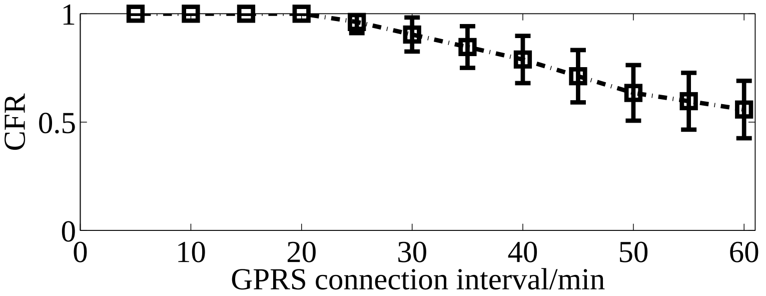


Figure 2.10: The condition fulfilled ratio (CFR) versus the GPRS connecting interval. Confidence intervals of 95% are used. We see that condition D4 is always satisfied with a GPRS interval less than 20 minutes.

2.6 Real Experiments

We run a 15-day experiment on an outdoor testbed on the EPFL campus. As shown in Figure 2.11, we deploy 2 different networks at the same 9 locations, resulting in a total number of 18 sensor nodes. These two networks use 868 MHz and 870 MHz frequency bands separately and thus do not interfere with each other. The general architecture of these two networks is the same as discussed in Section 2.2. The first network N_1 is composed of 3 BSs ($A1$, $A2$ and $A3$) and 6 RSNs ($A4$, $A5$, $A6$, $A7$, $A8$ and $A9$). This network runs HEF to adaptively choose one active BS. The second network N_2 is also composed of 3 BSs ($B1$, $B2$ and $B3$) and 6 RSNs ($B4$, $B5$, $B6$, $B7$, $B8$ and $B9$). It runs FIXED, which keeps BS $B2$ active and BSs $B1$, $B3$ passive. The data packet is generated as follows. Each sensor node generates a 2-byte counter every 30 seconds. The value of the counter changes according to a triangular waveform. Then, each counter is attached with a 4-byte timestamp and a 2-byte indicator for indicating message types. We duplicate them into four copies and then encapsulate them into data packets. Each data packet has a 3-byte header containing the node IDs and the hop distances to the active BS. The average data-generating rate of each sensor node is 35 byte/30 sec. All these data packets are routed to the active BS that connects to the remote server by using GSM/GPRS every 5

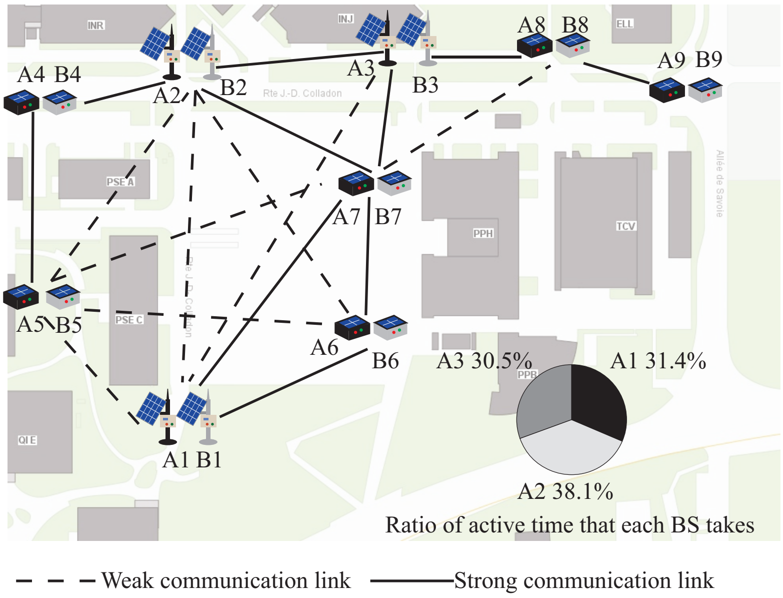


Figure 2.11: Experiment testbed on the EPFL campus. Two groups of 9 sensor nodes are installed at the same locations. The two groups use different communication radio frequencies and thus form two separate networks. In the network with the black nodes, we use HEF to coordinate 3 BSs, $A1$, $A2$ and $A3$. They are active for 31.4%, 38.1% and 30.5% of the total time respectively. The network with the white nodes has a fixed active BS $B2$. The solid lines represent communication links of sustained good quality. The dotted lines represent temporary communication links.

minutes. On average, the active BS transmits $9 \times 35\,\text{byte} \times 5\,\text{min}/30\,\text{sec} = 3150\,\text{byte}$ data every 5 minutes.

In the experiment, we use the battery level as the indicator for the available energy. Every 5 minutes, each BS sends its battery level to the active BS in a BS_ADVERT message (Section 2.2.3). The active BS then forwards this message to the remote server, hence we are able to observe the variations of the available energy in the WSN. Notice that this message is transmitted with a low rate and it will not add much communication burden to the network.

BSs and RSNs are equipped with solar panels with areas of $100\,\text{cm}^2$ and $50\,\text{cm}^2$, respectively. They are all equipped with 4 AA NiMH rechargeable batteries (each battery has a capacity of 800 mAh). In Figure 2.12, we show the battery levels of the six BSs. We see that in network N_1, the 3 BSs with ID $A1$, $A2$ and $A3$ almost always keep the same battery levels, although their solar panels harvest different amounts of energy. During this period of 15 days, their batteries do not deplete. Meanwhile, in network N_2, the passive BSs $B1$ and $B3$ always have high battery levels because of their low energy-consumption rates. The always-active BS $B2$ consumes its battery quickly and on the 4th, 8th and 12th days, the batteries of $B2$ drain out and we have to change them. From the experiment, we conclude that by deploying multiple BSs and adaptively choosing the active BS, the harvested energy is fully used and the network lifetime is prolonged.

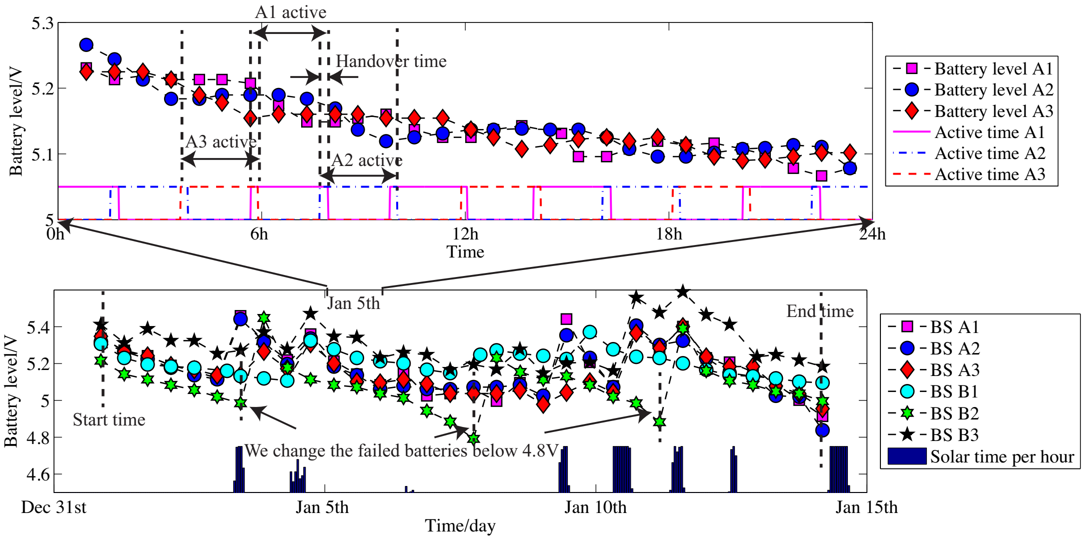


Figure 2.12: Battery levels versus time in the real experiment with six BSs. BSs with ID $A1$,$A2$ and $A3$ share the burden of being active and run the HEF algorithm. As a comparison, $B2$ is an always-active BS while BSs $B1$ and $B3$ are always passive. We observe two facts as follows. First, the amounts of available energy of the BSs $A1$, $A2$ and $A3$ are almost all the same during this 15 days. To clarify this point, we especially investigate the data on Jan 5th. We see that the active BS consumes energy quickly in each time interval of two hours. However, BSs running HEF take turns to share this high cost and averages out the temporal and spatial variations of the energy captured from solar panels. Second, by running the proposed scheme, the lifetime of the WSN is prolonged. We have to change 3 times the batteries of BS $B2$ on Jan 3th, Jan 7th and Jan 11th. Meanwhile, we do not need to change the batteries for the network running HEF during the entire 15 days.

2.7 Conclusion

In this chapter, we presented and evaluated a novel scheme for organizing WSNs, in which multiple BSs are deployed, but only one BS is adaptively selected to be active. By using the proposed scheme, we efficiently utilize the temporally and spatially varying energy resources available to all BSs. Therefore, the large batteries and energy harvesting devices of individual BSs can be substantially reduced.

To adaptively choose the active BS, we proposed a simple yet powerful algorithm HEF. We proved its asymptotic optimality under mild conditions.

Through simulations on the simulator Omnet++/Castalia and real experiments on an outdoor testbed, we showed that the proposed scheme is energy-efficient, is adaptable to network changes and is low in communication overhead.

2.A Appendix

2.A.1 Azuma-Hoeffding inequality

In both the proofs of Theorem 2.1 and Theorem 2.2, we will use the Azuma-Hoeffding inequality [30, p. 476] for martingales with bounded differences. We repeat it here for convenience.

Lemma 2.1

Suppose $\{H^{(n)}\}_{n\in\mathbb{N}}$ *is a martingale and* $|H^{(n)} - H^{(n-1)}| < c_n$ *almost surely, where* c_n *is positive real for any* $n \in \mathbb{N}$*. Then for any positive integer* N *and any positive* γ*,*

$$P(H^{(N)} - H^{(0)} \geq \gamma) \leq \exp\left(\frac{-\gamma^2}{2\sum_{n=1}^{N} c_n^2}\right),$$

and conversely

$$P(H^{(N)} - H^{(0)} \leq -\gamma) \leq \exp\left(\frac{-\gamma^2}{2\sum_{n=1}^{N} c_n^2}\right).$$

In our sensor network scenario, for any BS $m \in V_b$, we construct a martingale $\{h_m^{(n)}\}_{n\in\mathbb{N}}$ with

$$h_m^{(n)} = e_m^{(0)} + \tau \sum_{t=1}^{n} \left(s_m^{(t)} - \bar{s}\right), \tag{2.13}$$

where

$$\mathbb{E}_{n-1} h_m^{(n)} = h_m^{(n-1)}$$

because of (2.1). Here $\mathbb{E}_{n-1}$ is shorthand for denoting the conditional expectation given $\mathcal{S}^{(n-1)}$. Using (2.4) and (2.6), we have

$$e_m^{(n)} = e_m^{(0)} + \tau \sum_{t=1}^{n} \left(s_m^{(t)} - \bar{s}\right) - \tau \sum_{t=1}^{n} (\boldsymbol{R}\boldsymbol{v}^{(t)})_m.$$

Therefore, (2.13) is equivalent to

$$h_m^{(n)} = e_m^{(n)} + \tau \sum_{t=1}^{n} (\boldsymbol{R}\boldsymbol{v}^{(t)})_m. \tag{2.14}$$

Using Lemma 2.1, we have the following result.

Corollary 2.1

When the energy-recharge rates $\{\boldsymbol{s}^{(n)}\}_{n\in\mathbb{N}}$ *satisfy conditions (2.1) and (2.2), for any* $n_1 < n_2 \in \mathbb{N}$ *and for any* $\Delta_1, \Delta_2 > 0$,

$$\begin{aligned}\mathbb{P}\left(h_m^{(n_2)} - h_m^{(n_1)} \le -\Delta_1 - (n_2-n_1)\Delta_2\right) \\ \le \exp\left(\frac{-(\Delta_1 + (n_2-n_1)\Delta_2)^2}{2(n_2-n_1)\tau^2 S^2}\right) \\ \le \exp\left(\frac{-(n_2-n_1)\Delta_2^2}{2\tau^2 S^2}\right)\cdot \exp\left(-\frac{\Delta_1\Delta_2}{\tau^2 S^2}\right),\end{aligned}$$

because of (2.13) and because (2.2) yields that $s_m^{(n)} \le S$. *Similarly*

$$\begin{aligned}\mathbb{P}\left(h_m^{(n_2)} - h_m^{(n_1)} \ge \Delta_1 + (n_2-n_1)\Delta_2\right) \\ \le \exp\left(\frac{-(n_2-n_1)\Delta_2^2}{2\tau^2 S^2}\right)\cdot \exp\left(-\frac{\Delta_1\Delta_2}{\tau^2 S^2}\right).\end{aligned}$$

2.A.2 Proof of Theorem 2.1

(i) We first show that (2.8) holds when $f^* < 0$. We consider a scheduling algorithm that lets the cumulative active time for BS m at time slot n be

$$\sum_{t=1}^{n} v_m^{(t)} = \begin{cases} \lfloor n\bar{v}_m^* \rfloor, & 1 \le m \le |V_b|-1, \\ n - \sum_{k=1}^{|V_b|-1} \lfloor n\bar{v}_k^* \rfloor, & m = |V_b|, \end{cases}$$

where $\bar{v}_m^*$ is the m-th element of the optimal solution $\bar{\boldsymbol{v}}^*$ of problem (2.7). We see that: for any $1 \le m \le |V_b|-1$, $n\bar{v}_m^* - 1 \le \sum_{t=1}^{n} v_m^{(t)} \le n\bar{v}_m^*$; for $m = |V_b|$, $n\bar{v}_m^* \le \sum_{t=1}^{n} v_m^{(t)} \le n\bar{v}_m^* + |V_b| - 1$ due to $\sum_{m=1}^{|V_b|} n\bar{v}_m^* = n$ and $n\bar{v}_k^* - 1 \le \lfloor n\bar{v}_k^* \rfloor \le n\bar{v}_k^*$ for $1 \le k \le |V_b|-1$. Consequently,

$$\left\| n\bar{\boldsymbol{v}}^* - \sum_{t=1}^{n} \boldsymbol{v}^{(t)} \right\|_\infty < |V_b|. \tag{2.15}$$

We define the lifetime of each BS m ($1 \le m \le |V_b|$) as

$$N_m = \inf\{\{\infty\} \cup \{n | e_m^{(n+1)} < 0, n \in \mathbb{N}\}\}.$$

The lifetime of the whole network using the aforementioned scheduling algorithm is

$$N^* = \min_{1 \le m \le |V_b|} N_m.$$

Because N_{opt} is the optimal lifetime, we have $N^* \le N_{\text{opt}}$. To show (2.8), it suffices to show that

$$\lim_{e\to\infty} \mathbb{P}(N^* = \infty) = 1. \tag{2.16}$$

Because of (2.14), $e_m^{(n)} < 0$ implies that

$$\begin{aligned} h_m^{(n)} &< \tau \sum_{t=1}^{n} (\boldsymbol{R}\boldsymbol{v}^{(t)})_m \\ &= n\tau(\boldsymbol{R}\bar{\boldsymbol{v}}^*)_m + \tau \boldsymbol{R}(\sum_{t=1}^{n} \boldsymbol{v}^{(t)} - n\bar{\boldsymbol{v}}^*)_m \\ &< n\tau(\boldsymbol{R}\bar{\boldsymbol{v}}^*)_m + |V_b|\tau \left\|\boldsymbol{R}\right\|_\infty \cdot \left\| \sum_{t=1}^{n} \boldsymbol{v}^{(t)} - n\bar{\boldsymbol{v}}^* \right\|_\infty \\ &< n\tau a_m + |V_b|^2 \tau \|\boldsymbol{R}\|_\infty, \end{aligned} \tag{2.17}$$

where in the second inequality we use (2.15) and in the third inequality we define $a_m = (\boldsymbol{R}\bar{\boldsymbol{v}}^*)_m$. Due to $\boldsymbol{R}\bar{\boldsymbol{v}}^* \leq f^* \boldsymbol{u}$ in problem (2.7), we have $a_m \leq f^* < 0$.

Using the union bound, we have

$$\begin{aligned} &\mathbb{P}\left(N_m < \infty\right) \\ &\quad = \mathbb{P}(\exists n \geq 1 : e_m^{(n+1)} < 0) \\ &\quad = \mathbb{P}(\exists n \geq 2 : e_m^{(n)} < 0) \\ &\quad \leq \mathbb{P}(\exists n \geq 2 : h_m^{(n)} < n\tau a_m + |V_b|^2 \tau \|\boldsymbol{R}\|_\infty) \\ &\quad \leq \sum_{n=2}^{\infty} \mathbb{P}(h_m^{(n)} < n\tau a_m + |V_b|^2 \tau \|\boldsymbol{R}\|_\infty) \end{aligned} \tag{2.18}$$

Using Corollary 2.1, we have

$$\begin{aligned} &\mathbb{P}(h_m^{(n)} < n\tau a_m + |V_b|^2 \tau \|\boldsymbol{R}\|_\infty) \\ &\quad = \mathbb{P}\left(h_m^{(n)} - h_m^{(0)} < -e + n\tau a_m + |V_b|^2 \tau \|\boldsymbol{R}\|_\infty\right) \\ &\quad \leq \exp\left(\frac{-n a_m^2}{2S^2}\right) \cdot \exp\left(\frac{(e - |V_b|^2 \tau \|\boldsymbol{R}\|_\infty) a_m}{\tau S^2}\right) \end{aligned} \tag{2.19}$$

By using (2.18) and (2.19) together and by noticing that $\sum_{n=2}^{\infty} \exp\left(-n a_m^2 / 2S^2\right) = \beta < \infty$, we have that

$$\mathbb{P}\left(N_m < \infty\right) < \beta \exp\left(\frac{(e - |V_b|^2 \tau \|\boldsymbol{R}\|_\infty) a_m}{\tau S^2}\right).$$

Using the union bound and taking the limit $e \to \infty$, we have

$$\lim_{e\to\infty} \mathbb{P}(N^* < \infty) \leq \lim_{e\to\infty} \sum_{m=1}^{|V_b|} \mathbb{P}(N_m < \infty) = 0,$$

which implies (2.16). Therefore, we have (2.8).

(ii) Then, we will show that (2.9) holds when $f^* > 0$ in the following two separate parts:

$$\begin{cases} \lim_{e\to\infty} \mathbb{P}\left(N_{\text{opt}} \leq (1-\delta)e/(\tau f^*)\right) = 0, \forall \delta > 0, & (2.20) \\ \lim_{e\to\infty} \mathbb{P}\left(N_{\text{opt}} \geq (1+\delta)e/(\tau f^*)\right) = 0, \forall \delta > 0. & (2.21) \end{cases}$$

1) In the first part, to show (2.20), it suffices to show that

$$\lim_{e\to\infty} \mathbb{P}(N^* \le (1-\delta)e/(\tau f^*)) = 0, \forall \delta > 0, \tag{2.22}$$

because $N^* \le N_{\text{opt}}$.

Because of (2.17), $e_m^{(n+1)} < 0$ implies that $h_m^{(n+1)} < (n+1)\tau a_m + |V_b|^2 \tau \|\boldsymbol{R}\|_\infty$. Using the union bound,

$$\begin{aligned}
&\mathbb{P}\left(N_m \le \frac{(1-\delta)e}{\tau f^*}\right) \\
&\quad = \mathbb{P}\left(\exists n \le \frac{(1-\delta)e}{\tau f^*} : e_m^{(n+1)} < 0\right) \\
&\quad = \mathbb{P}\left(\exists n \le \frac{(1-\delta)e}{\tau f^*} + 1 : h_m^{(n)} < n\tau a_m + |V_b|^2 \tau \|\boldsymbol{R}\|_\infty\right) \\
&\quad \le \sum_{n=1}^{(1-\delta)e/\tau f^* + 1} \mathbb{P}\left(h_m^{(n)} < n\tau a_m + |V_b|^2 \tau \|\boldsymbol{R}\|_\infty\right).
\end{aligned} \tag{2.23}$$

Now, using Lemma 2.1, we have

$$\begin{aligned}
&\mathbb{P}(h_m^{(n)} < n\tau a_m + |V_b|^2 \tau \|\boldsymbol{R}\|_\infty) \\
&\quad = \mathbb{P}\left(h_m^{(n)} - h_m^{(0)} < -e + |V_b|^2 \tau \|\boldsymbol{R}\|_\infty + n\tau a_m\right) \\
&\quad \le \exp\left(\frac{-(-e + |V_b|^2 \tau \|\boldsymbol{R}\|_\infty + n\tau a_m)^2}{2n\tau^2 S^2}\right) \\
&\quad \le \exp\left(\frac{-(\delta e - |V_b|^2 \tau \|\boldsymbol{R}\|_\infty)^2}{2(1-\delta)e\tau S^2/f^*}\right),
\end{aligned} \tag{2.24}$$

where the inequality on the fourth line is due to $n \le (1-\delta)e/(\tau f^*)$, and $a_m \le f^*$ which follows from $a_m = (\boldsymbol{R}\bar{\boldsymbol{v}}^*)_m$ and $\boldsymbol{R}\bar{\boldsymbol{v}}^* \le f^* \boldsymbol{u}$.

Combining (2.23) and (2.24) yields that

$$\begin{aligned}
&\mathbb{P}\left(N_m \le \frac{(1-\delta)e}{\tau f^*}\right) \\
&\quad \le \left(\frac{(1-\delta)e}{\tau f^*} + 1\right) \exp\left(\frac{-(\delta e - |V_b|^2 \tau \|\boldsymbol{R}\|_\infty)^2 f^*}{2(1-\delta)e\tau S^2}\right).
\end{aligned} \tag{2.25}$$

Using the union bound, we see that

$$\mathbb{P}\left(N^* \le \frac{(1-\delta)e}{\tau f^*}\right) \le \sum_{m=1}^{|V_b|} \mathbb{P}\left(N_m \le \frac{(1-\delta)e}{\tau f^*}\right).$$

Plugging in (2.25) and taking $e \to \infty$, we have (2.22).

2) In the second part, we will show (2.21). Consider the event that $N_{\text{opt}} \ge (1+\delta)e/(\tau f^*)$. It implies that there exists a sequence of decision vectors $\{\boldsymbol{v}^{(t)}\}_{t\in\mathbb{N}}$ such that the available energy at time $(1+\delta)e/(\tau f^*)$ is non-negative, *i.e.*,

$$\boldsymbol{e}^{(\frac{(1+\delta)e}{\tau f^*})} \ge \boldsymbol{0}. \tag{2.26}$$

Let $\bar{\boldsymbol{v}}_{\text{opt}}$ be the average decision vector from time 1 to $(1+\delta)e/(\tau f^*)$, that is,

$$\bar{\boldsymbol{v}}_{\text{opt}} = \frac{1}{(1+\delta)e/(\tau f^*)} \sum_{n=1}^{(1+\delta)e/(\tau f^*)} \boldsymbol{v}^{(n)}. \tag{2.27}$$

Using (2.4), (2.26) and that $\boldsymbol{e}^{(0)} = e\boldsymbol{u}$, we have that

$$e\boldsymbol{u} + \sum_{n=1}^{(1+\delta)e/(\tau f^*)} \tau \boldsymbol{s}^{(n)} \geq \tau \boldsymbol{C} \sum_{n=1}^{(1+\delta)e/(\tau f^*)} \boldsymbol{v}^{(n)}.$$

Substracting $(1+\delta)e/(\tau f^*)\bar{\boldsymbol{s}}$ on both sides of this inequality and using (2.6) and (2.27), we have that

$$e\boldsymbol{u} + \tau \sum_{n=1}^{(1+\delta)e/(\tau f^*)} (\boldsymbol{s}^{(n)} - \bar{\boldsymbol{s}}) \geq \frac{(1+\delta)e}{f^*} \boldsymbol{R}\bar{\boldsymbol{v}}_{\text{opt}}. \tag{2.28}$$

Because f^* is the optimal objective value of (2.7), there exists $1 \leq m^* \leq |V_b|$, such that $(\boldsymbol{R}\bar{\boldsymbol{v}}_{\text{opt}})_{m^*} \geq f^*$. Plugging it into (2.28) and dividing both sides of the equation by $(1+\delta)e/f^*$, we have

$$\frac{1}{(1+\delta)e/(\tau f^*)} \sum_{t=1}^{(1+\delta)e/(\tau f^*)} (s_{m^*}^{(t)} - \bar{s}_{m^*}) \geq \frac{f^*\delta}{(1+\delta)}. \tag{2.29}$$

Because $s_{m^*}^{(t)} - \bar{s}_{m^*}$ is a martingale difference term, it is uncorrelated at different time. Using the weak law of large numbers, $\sum_{n=1}^{n_0}(s_{m^*}^{(n)} - \bar{s}_{m^*})/n_0$ weakly converges to 0 when $n_0 \to \infty$. Therefore, the probability of (2.29) converges to zero when $e \to \infty$. Remember that event (2.29) is implied by $N_{\text{opt}} \geq (1+\delta)e/(\tau f^*)$, which concludes the proof.

2.A.3 Proof of Theorem 2.2

Notations and a short summary

For the ease of discussion, we define four events:

$$\begin{aligned}
A_1 &= \{\exists N_{\text{hef}} \leq \frac{(1-\delta)e}{\tau f^*}, l^* \in [1, |V_b|], s.t., e_{l^*}^{(N_{\text{hef}}+1)} < 0\}, \\
A_1' &= \{\exists N_{\text{hef}} \leq Ke, l^* \in [1, |V_b|], s.t., e_{l^*}^{(N_{\text{hef}}+1)} < 0\}, \\
A_2 &= \{e_m^{(N_{\text{hef}}+1)} < \epsilon_1 e, \forall 1 \leq m \leq |V_b|\}, \\
A_3 &= \left\{ \left\| \frac{1}{N_{\text{hef}}+1} \sum_{n=1}^{N_{\text{hef}}+1} \boldsymbol{v}^{(n)} - \frac{\boldsymbol{R}^{-1}\boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{R}^{-1}\boldsymbol{u}} \right\|_\infty < \epsilon_2 \right\},
\end{aligned} \tag{2.30}$$

where K is the constant defined in (2.11) and ϵ_1, ϵ_2 are two positive numbers which can be set arbitrarily small.

Events A_1 and A_1' mean that the lifetime incurred by HEF N_{hef} is not larger than $(1-\delta)e/\tau f^*$ and Ke, respectively. Event A_2 occurs when the available energy of any BS is smaller than $\epsilon_1 e$ at time $N_{\text{hef}}+1$. Event A_3 means that the average of decision vectors up to time $N_{\text{hef}}+1$ is arbitrarily close to $\boldsymbol{R}^{-1}\boldsymbol{u}/\boldsymbol{u}^\top\boldsymbol{R}^{-1}\boldsymbol{u}$.

Because $N_{\mathrm{hef}} \leq N_{\mathrm{opt}}$ and because of (2.9), when $f^* > 0$,

$$\lim_{e \to \infty} \mathbb{P}\left(\frac{N_{\mathrm{hef}}}{e/(\tau f^*)} - 1 < \delta\right) = 1, \forall \delta > 0. \tag{2.31}$$

Therefore, Theorem 2.2 can be recast as:

$$\lim_{e \to \infty} \mathbb{P}(A_1) = 0, \text{ if } f^* > 0, \tag{2.32}$$

$$\lim_{e \to \infty} \mathbb{P}(A_1') = 0, \text{ if } f^* < 0. \tag{2.33}$$

First, using Azuma-Hoeffding inequality [30, p. 476] and condition D3, we will show that the probability that HEF uses up all the available energy of all BSs converges to 1 when $e \to \infty$. More precisely, we will show that $\lim_{e\to\infty} \mathbb{P}(A_1' \cap \bar{A}_2) = 0$ and $\lim_{e\to\infty} \mathbb{P}(A_1 \cap \bar{A}_2) = 0$.

Secondly, using the duality of linear programming and condition D4, we will show that $\boldsymbol{R}^{-1}\boldsymbol{u}/\boldsymbol{u}^\top \boldsymbol{R}^{-1}\boldsymbol{u}$ is the optimal fractions of active time for all BSs. Then, we will show that if event A_3 is true, HEF performs optimally. We will deduce that: if $f^* < 0$, $\lim_{e\to\infty} \mathbb{P}(A_1' \cap A_3) = 0$; and if $f^* > 0$, $\lim_{e\to\infty} \mathbb{P}(A_1 \cap A_3) = 0$.

Thirdly, we will show that if HEF uses up the available energy of all BSs (event A_2 is true), the average of decision vector $\sum_{n=1}^{N_{\mathrm{hef}}+1} \boldsymbol{v}^{(n)}/(N_{\mathrm{hef}}+1)$ is close to $\boldsymbol{R}^{-1}\boldsymbol{u}/\boldsymbol{u}^\top \boldsymbol{R}^{-1}\boldsymbol{u}$ (event A_3 is true). Then, we will deduce that: if $\lim_{e\to\infty} \mathbb{P}(A_1' \cap A_3) = 0$, we have $\lim_{e\to\infty} \mathbb{P}(A_1' \cap A_2) = 0$; and if $\lim_{e\to\infty} \mathbb{P}(A_1 \cap A_3) = 0$, we have $\lim_{e\to\infty} \mathbb{P}(A_1 \cap A_2) = 0$.

To sum things up, to prove (2.32) and (2.33), we will show the following six separate points:

- Point $1A$: $\lim_{e\to\infty} \mathbb{P}(A_1' \cap \bar{A}_2) = 0$.
- Point $1B$: $\lim_{e\to\infty} \mathbb{P}(A_1 \cap \bar{A}_2) = 0$.
- Point $2A$: If $f^* < 0$, we have $\lim_{e\to\infty} \mathbb{P}(A_1' \cap A_3) = 0$.
- Point $2B$: If $f^* > 0$, we have $\lim_{e\to\infty} \mathbb{P}(A_1 \cap A_3) = 0$.
- Point $3A$: Given that $\lim_{e\to\infty} \mathbb{P}(A_1' \cap A_3) = 0$, we have $\lim_{e\to\infty} \mathbb{P}(A_1' \cap A_2) = 0$.
- Point $3B$: Given that $\lim_{e\to\infty} \mathbb{P}(A_1 \cap A_3) = 0$, we have $\lim_{e\to\infty} \mathbb{P}(A_1 \cap A_2) = 0$.

Then, combining Points $1A$, $2A$ and $3A$, we have

$$\lim_{e \to \infty} \mathbb{P}(A_1') = \lim_{e \to \infty} (\mathbb{P}(A_1' \cap A_2) + \mathbb{P}(A_1' \cap \bar{A}_2)) = 0,$$

which proves (2.33), and likewise, Points $1B$, $2B$ and $3B$ yield (2.32).

For the ease of discussion, we define two constants

$$\begin{cases} d_1 = \max\limits_{m \neq j} R_{mj} < 0, & (2.34) \\ d_2 = \max\limits_{m,j} C_{mj} > 0. & (2.35) \end{cases}$$

Proof for point $1A$ and point $1B$

We only show the proof for point $1A$ here. The proof for point $1B$ is identical if we replace K with $(1-\delta)/(\tau f^*)$. Let l^* be a BS that drains out of energy at time $N_{\mathrm{hef}}+1$, *i.e.*, $e_{l^*}^{(N_{\mathrm{hef}}+1)} < 0$. Let N' be the last time that BS l^* is selected as the active BS (we set $N' = 0$ if BS l^* is never selected to be active), that is,

$$N' = \sup\{\{0\} \cup \{n \mid \exists l^*, v_{l^*}^{(n)} = 1, n \leq N_{\mathrm{hef}}\}\}.$$

We define the event A_4 as

$$A_4 = \{N_{\text{hef}} - N' + 1 \geq \epsilon_3 e\}.$$

Using (2.14), we see that

$$\begin{aligned} h_{l^*}^{(N_{\text{hef}}+1)} - h_{l^*}^{(N')} &= e_{l^*}^{(N_{\text{hef}}+1)} - e_{l^*}^{(N')} + \tau \sum_{n=N'+1}^{N_{\text{hef}}+1} (\boldsymbol{Rv}^{(n)})_{l^*} \\ &< d_1 \tau (N_{\text{hef}} - N' + 1), \end{aligned} \tag{2.36}$$

where the inequality holds because $e_{l^*}^{(N_{\text{hef}}+1)} < 0 \leq e_{l^*}^{(N')}$, because BS l^* is not selected as the active BS from time $N'+1$ to $N_{\text{hef}}+1$ and because of (2.34).

Using Corollary 2.1, we see that

$$\begin{aligned} &\mathbb{P}\left(h_{l^*}^{(N_{\text{hef}}+1)} - h_{l^*}^{(N')} < d_1 \tau (N_{\text{hef}} - N' + 1)\right) \\ &\qquad \leq \exp\left(\frac{-(N_{\text{hef}} - N' + 1) d_1^2}{2S^2}\right). \end{aligned} \tag{2.37}$$

Using the union bound, we have

$$\begin{aligned} &\mathbb{P}(A_1' \cap A_4) \\ &\quad \leq \sum_{N'=1}^{Ke} \sum_{N_{\text{hef}}=1}^{Ke} \mathbb{P}\left(h_{l^*}^{(N_{\text{hef}}+1)} - h_{l^*}^{(N')} < d_1 \tau (N_{\text{hef}} - N' + 1)\right) \\ &\quad \leq (Ke)^2 \exp\left(-\frac{(N_{\text{hef}} - N' + 1) d_1^2}{2S^2}\right) \\ &\quad \leq (Ke)^2 \exp\left(-\frac{\epsilon_3 e d_1^2}{2S^2}\right), \end{aligned} \tag{2.38}$$

where the second inequality follows from $N', N_{\text{hef}} \leq Ke$ and (2.37), and the third inequality follows from $N_{\text{hef}} - N' + 1 \geq \epsilon_3 e$ in the definition of A_4.

Taking $e \to \infty$ on both sides of (2.38), we see that $\lim_{e\to\infty} \mathbb{P}(A_1' \cap A_4) = 0$.

In the following, we will show that $\bar{A}_4 \subseteq A_2$. If $\bar{A}_4$ is true, $N_{\text{hef}} - N' + 1 < \epsilon_3 e$. Then, by summing up (2.3) from time N' to $N_{\text{hef}}+1$, we derive the following upper-bound for the available energy $e_{l^*}^{(N')}$

$$\begin{aligned} e_{l^*}^{(N')} &= e_{l^*}^{(N_{\text{hef}}+1)} - \sum_{n=N'+1}^{N_{\text{hef}}+1} s_{l^*}^{(n)} + \sum_{n=N'+1}^{N_{\text{hef}}+1} (\boldsymbol{Cv}^{(n)})_{l^*}, \\ &< \sum_{n=N'+1}^{N_{\text{hef}}+1} (\boldsymbol{Cv}^{(n)})_{l^*} \\ &< d_2 \epsilon_3 e, \end{aligned} \tag{2.39}$$

where the first inequality holds because $e_{l^*}^{(N_{\text{hef}}+1)} < 0$ and because $s_{l^*}^{(n)} > 0$ for all n, and the second inequality holds because $N_{\text{hef}} - N' + 1 < \epsilon_3 e$, because l^* is not selected as the active BS between time $N'+1$ to time $N_{\text{hef}}+1$ and because of (2.35).

Because BS l^* is selected by HEF at time N', it has the highest available energy among all BSs. Therefore, we have $e_m^{(N')} < d_2\epsilon_3 e$ for all $m \in V_b$. This leads to the following upper-bound for the available energy of all BSs at time $N_{\text{hef}}+1$,

$$\begin{aligned} e_m^{(N_{\text{hef}}+1)} &= e_m^{(N')} + \sum_{n=N'+1}^{N_{\text{hef}}+1} s_m^{(n)} - \sum_{n=N'+1}^{N_{\text{hef}}+1} (\boldsymbol{C}\boldsymbol{v}^{(n)})_m \\ &< d_2\epsilon_3 e + (N_{\text{hef}} - N' + 1)S \\ &< d_2\epsilon_3 e + \epsilon_3 e S = \epsilon_1 e. \end{aligned} \tag{2.40}$$

where in the first inequality we use $0 \leq s_m^{(n)} \leq S$ and in the second inequality we use $N_{\text{hef}} - N' + 1 < \epsilon_3 e$ and set $\epsilon_1 = (d_2 + S)\epsilon_3$.

Therefore, if $\bar{A}_4$ holds, so does A_2. Hence, $\bar{A}_4 \subseteq A_2$ which follows that $\bar{A}_2 \subseteq A_4$. Consequently, $A_1' \cap \bar{A}_2 \subseteq A_1' \cap A_4$. It follows that $\lim_{e\to\infty} \mathbb{P}(A_1' \cap \bar{A}_2) \leq \lim_{e\to\infty} \mathbb{P}(A_1' \cap A_4) = 0$.

Proof for point $2A$ and point $2B$

We consider the following two scenarios when showing both points $2A$ and $2B$: (i) neither $\boldsymbol{R}^{-1}\boldsymbol{u} \geq \boldsymbol{0}$ nor $\boldsymbol{R}^{-1}\boldsymbol{u} \leq \boldsymbol{0}$ is satisfied; (ii) either $\boldsymbol{R}^{-1}\boldsymbol{u} \geq \boldsymbol{0}$ or $\boldsymbol{R}^{-1}\boldsymbol{u} \leq \boldsymbol{0}$ is satisfied.

(i) In the first scenario, the vector $\boldsymbol{R}^{-1}\boldsymbol{u}/\boldsymbol{u}^\top\boldsymbol{R}^{-1}\boldsymbol{u}$ contains at least one negative element. Let d_3 be the maximum value among all the negative elements of $\boldsymbol{R}^{-1}\boldsymbol{u}/\boldsymbol{u}^\top\boldsymbol{R}^{-1}\boldsymbol{u}$. Noticing that $\sum_{n=1}^{N_{\text{hef}}+1} \boldsymbol{v}^{(n)}/(N_{\text{hef}}+1) \geq \boldsymbol{0}$, we see that if we select $\epsilon_2 < -d_3$, event A_3 is always false. Therefore, by selecting a small enough ϵ_2, we have both

$$\lim_{e\to\infty} \mathbb{P}(A_1 \cap A_3) \leq \lim_{e\to\infty} \mathbb{P}(A_3) = 0, \tag{2.41}$$

$$\lim_{e\to\infty} \mathbb{P}(A_1' \cap A_3) \leq \lim_{e\to\infty} \mathbb{P}(A_3) = 0. \tag{2.42}$$

(ii) We are left with the second scenario where either $\boldsymbol{R}^{-1}\boldsymbol{u} \geq \boldsymbol{0}$ or $\boldsymbol{R}^{-1}\boldsymbol{u} \leq \boldsymbol{0}$. In this scenario, we compute in Lemma 2.2 the optimal solution of problem (2.7) analytically.

Lemma 2.2

Under the conditions that: (i) either $\boldsymbol{R}^{-1}\boldsymbol{u} \geq \boldsymbol{0}$ or $\boldsymbol{R}^{-1}\boldsymbol{u} \leq \boldsymbol{0}$, and (ii) $(\boldsymbol{C}^\top)^{-1}\boldsymbol{u} \geq \boldsymbol{0}$, the optimal solution of problem (2.7) is

$$\begin{cases} \bar{\boldsymbol{v}}^* = \boldsymbol{R}^{-1}\boldsymbol{u}/\boldsymbol{u}^\top\boldsymbol{R}^{-1}\boldsymbol{u}, & (2.43) \\ f^* = 1/\boldsymbol{u}^\top\boldsymbol{R}^{-1}\boldsymbol{u}. & (2.44) \end{cases}$$

Proof.

The general idea of the proof is to use the duality properties of linear programmings. When either $\boldsymbol{R}^{-1}\boldsymbol{u} \geq \boldsymbol{0}$ or $\boldsymbol{R}^{-1}\boldsymbol{u} \leq \boldsymbol{0}$, $\bar{\boldsymbol{v}} = \boldsymbol{R}^{-1}\boldsymbol{u}/\boldsymbol{u}^\top\boldsymbol{R}^{-1}\boldsymbol{u}$ is a feasible solution of problem (2.7), whose corresponding objective value is $1/\boldsymbol{u}^\top\boldsymbol{R}^{-1}\boldsymbol{u}$. Because f^* is the optimal objective value of (2.7), we have $1/\boldsymbol{u}^\top\boldsymbol{R}^{-1}\boldsymbol{u} \geq f^*$. In the following, we will show $f^* \geq 1/\boldsymbol{u}^\top\boldsymbol{R}^{-1}\boldsymbol{u}$.

The dual problem of (2.7) is written as:

$$\begin{aligned}
\max_{w,\boldsymbol{\lambda}} \quad & w \\
\text{s.t.} \quad & \boldsymbol{u}^\top \boldsymbol{\lambda} = 1, \\
& \boldsymbol{R}^\top \boldsymbol{\lambda} \geq w \boldsymbol{u}, \\
& \boldsymbol{\lambda} \geq \mathbf{0}.
\end{aligned} \tag{2.45}$$

Because of the Sherman-Woodbury-Morrison formula and because $\boldsymbol{R} = \boldsymbol{C} - \bar{\boldsymbol{s}} \cdot \boldsymbol{u}^\top$, we have that

$$(\boldsymbol{R}^\top)^{-1}\boldsymbol{u} = (\boldsymbol{C}^\top)^{-1}\boldsymbol{u}/(1 - \boldsymbol{u}^\top \boldsymbol{C}^{-1}\bar{\boldsymbol{s}}).$$

Because $(\boldsymbol{C}^\top)^{-1}\boldsymbol{u} \geq \mathbf{0}$, we either have $(\boldsymbol{R}^\top)^{-1}\boldsymbol{u} \geq \mathbf{0}$ or $(\boldsymbol{R}^\top)^{-1}\boldsymbol{u} \leq \mathbf{0}$, depending on the difference between $\boldsymbol{u}^\top \boldsymbol{C}^{-1}\bar{\boldsymbol{s}}$ and 1. In both cases, we have a feasible solution of the dual problem

$$\begin{cases} \boldsymbol{\lambda} = \dfrac{(\boldsymbol{R}^\top)^{-1}\boldsymbol{u}}{\boldsymbol{u}^\top (\boldsymbol{R}^\top)^{-1}\boldsymbol{u}}, \\ w = \dfrac{1}{\boldsymbol{u}^\top (\boldsymbol{R}^\top)^{-1}\boldsymbol{u}}. \end{cases}$$

Consequently, the objective value $1/\left(\boldsymbol{u}^\top (\boldsymbol{R}^\top)^{-1}\boldsymbol{u}\right)$ reached by this feasible solution of the dual problem provides a lower bound of the objective value for the original problem (2.7), that is, $f^* \geq 1/\boldsymbol{u}^\top (\boldsymbol{R}^\top)^{-1}\boldsymbol{u}$. By noticing that

$$1/\boldsymbol{u}^\top \cdot (\boldsymbol{R}^\top)^{-1} \cdot \boldsymbol{u} = 1/\boldsymbol{u}^\top \cdot (\boldsymbol{R})^{-1} \cdot \boldsymbol{u},$$

we see that $f^* = 1/\boldsymbol{u}^\top \boldsymbol{R}^{-1}\boldsymbol{u}$. The solution that attains this optimal objective value is $\bar{\boldsymbol{v}}^* = \boldsymbol{R}^{-1}\boldsymbol{u}/\boldsymbol{u}^\top \boldsymbol{R}^{-1}\boldsymbol{u}$. ☐

When event A_3 is true, Using (2.14), we see that

$$\begin{aligned}
& h_{l^*}^{(N_{\text{hef}}+1)} - h_{l^*}^{(0)} \\
& \quad = e_{l^*}^{(N_{\text{hef}}+1)} - e + \tau \sum_{n=1}^{N_{\text{hef}}+1} (\boldsymbol{R}\boldsymbol{v}^{(n)})_{l^*} \\
& \quad < -e + \tau \sum_{n=1}^{N_{\text{hef}}+1} (\boldsymbol{R}(\boldsymbol{v}^{(n)} - \bar{\boldsymbol{v}}^*))_{l^*} + \tau \sum_{n=1}^{N_{\text{hef}}+1} (\boldsymbol{R}\bar{\boldsymbol{v}}^*)_{l^*} \\
& \quad < -e + \tau(N_{\text{hef}} + 1) \cdot |V_b|\epsilon_2 \|\boldsymbol{R}\|_\infty + \tau(N_{\text{hef}} + 1) \cdot f^*,
\end{aligned} \tag{2.46}$$

where the inequality on the third line is because $e_{l^*}^{(N_{\text{hef}}+1)} < 0$, and the inequality on the fourth line follows from (2.30) and $\boldsymbol{R}\bar{\boldsymbol{v}}^* \leq f^*\boldsymbol{u}$. We will use (2.46) in both the proof for points $2A$ and $2B$.

1) We first look at point $2A$ and consider the event $A_1' \cap A_3$.

We denote by

$$d_4 = -f^* - |V_b|\epsilon_2 \|\boldsymbol{R}\|_\infty,$$

which is positive when we select $\epsilon_2 < -f^*/|V_b|\|\boldsymbol{R}\|_\infty$.

Plugging the definition of d_4 into (2.46) and using Corollary 2.1, we have

$$\begin{aligned}&\mathbb{P}(h_{l^*}^{(N_{\text{hef}}+1)} - h_{l^*}^{(0)} < -e - \tau d_4(N_{\text{hef}}+1))\\&\quad\leq \exp\left(\frac{-(N_{\text{hef}}+1)d_4^2}{2S^2}\right)\exp\left(-\frac{ed_4}{\tau S^2}\right)\\&\quad\leq \exp\left(-\frac{ed_4}{\tau S^2}\right).\end{aligned}$$

Using the union bound and using $N_{\text{hef}} \leq Ke$, we have

$$\begin{aligned}&\mathbb{P}(A_1' \cap A_3)\\&\quad\leq \sum_{N_{\text{hef}}=1}^{Ke} \mathbb{P}\left(h_{l^*}^{(N_{\text{hef}}+1)} - h_{l^*}^{(0)} < -e - \tau d_4(N_{\text{hef}}+1)\right)\\&\quad\leq Ke\exp\left(-\frac{ed_4}{\tau S^2}\right).\end{aligned}$$

Taking $e \to \infty$, we see that $\lim_{e\to\infty} \mathbb{P}(A_1' \cap A_3) = 0$.

2) Then, we show point $2B$ by considering the event $A_1 \cap A_3$. Through (2.46), we have

$$\begin{aligned}&h_{l^*}^{(N_{\text{hef}}+1)} - h_{l^*}^{(0)}\\&\quad< -e + (|V_b|\epsilon_2\|\boldsymbol{R}\|_\infty + f^*)\tau(N_{\text{hef}}+1)\\&\quad\leq -e + (|V_b|\epsilon_2\|\boldsymbol{R}\|_\infty + f^*)((1-\delta)e/f^* + \tau)\\&\quad= -\left(\delta - \frac{|V_b|\epsilon_2(1-\delta)\|\boldsymbol{R}\|_\infty}{f^*}\right)e + \tau(|V_b|\epsilon_2\|\boldsymbol{R}\|_\infty + f^*),\end{aligned} \tag{2.47}$$

where the inequality on the third line is because $N_{\text{hef}} \leq (1-\delta)e/(\tau f^*)$. We define a small constant $\zeta > 0$ and see that $\tau(|V_b|\epsilon_2\|\boldsymbol{R}\|_\infty + f^*) \leq \zeta e$ when e is large. Define

$$d_5 = \delta - |V_b|\epsilon_2(1-\delta)\|\boldsymbol{R}\|_\infty/f^* - \zeta,$$

which is positive when we set $\epsilon_2 < (\delta-\zeta)f^*/|V_b|(1-\delta)\|\boldsymbol{R}\|_\infty$.

Plugging d_5 into (2.47) and using Corollary 2.1,

$$\mathbb{P}(h_{l^*}^{(N_{\text{hef}}+1)} - h_{l^*}^{(0)} < -d_5 e) \leq \exp\left(-\frac{d_5^2 e^2}{2\tau^2 S^2}\right).$$

Using the union bound and using $N_{\text{hef}} \leq (1-\delta)e/(\tau f^*)$, we have

$$\begin{aligned}&\mathbb{P}(A_1 \cap A_3)\\&\quad\leq \sum_{N_{\text{hef}}=1}^{(1-\delta)e/(\tau f^*)} \mathbb{P}\left(h_{l^*}^{(N_{\text{hef}}+1)} - h_{l^*}^{(0)} < -d_5 e\right)\\&\quad\leq \frac{(1-\delta)e}{\tau f^*}\exp\left(-\frac{d_5^2 e^2}{2\tau^2 S^2}\right).\end{aligned}$$

Taking $e \to \infty$, we see that $\lim_{e\to\infty} \mathbb{P}(A_1 \cap A_3) = 0$.

Proof for point $3A$ and point $3B$

The proofs for $3A$ and point $3B$ are identical. Therefore, we only show the proof for $3A$ here. We define the event A_5 as

$$A_5 = \left\{ \frac{1}{N_{\text{hef}}+1} \left\| \sum_{n=1}^{N_{\text{hef}}+1} \boldsymbol{s}^{(n)} - \bar{\boldsymbol{s}} \right\|_\infty < \epsilon_4 \right\}, \tag{2.48}$$

where ϵ_4 is a small constant.

We first show that event $A_1' \cap A_2 \cap A_5 \subseteq A_1' \cap A_3$. Using (2.4) and (2.6), we calculate the difference between the available energy of all BSs at time 0 and that at time $N_{\text{hef}}+1$:

$$\boldsymbol{e}^{(N_{\text{hef}}+1)} = \boldsymbol{e}^{(0)} - \sum_{n=1}^{N_{\text{hef}}+1} \boldsymbol{R}\boldsymbol{v}^{(n)} + \sum_{n=1}^{N_{\text{hef}}+1} \left(\boldsymbol{s}^{(n)} - \bar{\boldsymbol{s}}\right),$$

which is equivalent to

$$\begin{aligned} &\frac{1}{N_{\text{hef}}+1} \sum_{n=1}^{N_{\text{hef}}+1} \boldsymbol{v}^{(n)} - \frac{1}{N_{\text{hef}}+1} \boldsymbol{R}^{-1}(\boldsymbol{e}^{(0)} - \boldsymbol{e}^{(N_{\text{hef}}+1)}) \\ &\quad = (\boldsymbol{R})^{-1} \frac{\sum_{n=1}^{N_{\text{hef}}+1} \left(\boldsymbol{s}^{(n)} - \bar{\boldsymbol{s}}\right)}{N_{\text{hef}}+1}. \end{aligned} \tag{2.49}$$

Multiplying $\boldsymbol{u}^\top$ on both sides and using $\boldsymbol{u}^\top \cdot \sum_{n=1}^{N_{\text{hef}}+1} \boldsymbol{v}^{(n)} = N_{\text{hef}+1}+1$, we see that

$$1 - \boldsymbol{u}^\top \boldsymbol{R}^{-1} \frac{\boldsymbol{e}^{(0)} - \boldsymbol{e}^{(N_{\text{hef}}+1)}}{N_{\text{hef}}+1} = \boldsymbol{u}^\top \boldsymbol{R}^{-1} \frac{\sum_{n=1}^{N_{\text{hef}}+1} \left(\boldsymbol{s}^{(n)} - \bar{\boldsymbol{s}}\right)}{N_{\text{hef}}+1}. \tag{2.50}$$

Taking the infinite norm on both sides and knowing that event A_5 (2.48) holds true, we transform (2.50) into

$$\left| 1 - \frac{\boldsymbol{u}^\top \boldsymbol{R}^{-1}(\boldsymbol{e}^{(0)} - \boldsymbol{e}^{(N_{\text{hef}}+1)})}{N_{\text{hef}}+1} \right| < |V_b| \|\boldsymbol{u}^\top \boldsymbol{R}^{-1}\|_\infty \epsilon_4. \tag{2.51}$$

Then, since $\boldsymbol{e}^{(0)} = e\boldsymbol{u}$, (2.51) becomes

$$\begin{aligned} \left| 1 - \frac{e\boldsymbol{u}^\top \boldsymbol{R}^{-1}\boldsymbol{u}}{N_{\text{hef}}+1} \right| &< |V_b| \|\boldsymbol{u}^\top \boldsymbol{R}^{-1}\|_\infty \epsilon_4 + \left| \frac{\boldsymbol{u}^\top \boldsymbol{R}^{-1} \boldsymbol{e}^{(N_{\text{hef}}+1)}}{N_{\text{hef}}+1} \right| \\ &< |V_b| \|\boldsymbol{u}^\top \boldsymbol{R}^{-1}\|_\infty \left(\epsilon_4 + \frac{\|\boldsymbol{e}^{(N_{\text{hef}}+1)}\|_\infty}{N_{\text{hef}}+1} \right) \\ &< |V_b| \|\boldsymbol{u}^\top \boldsymbol{R}^{-1}\|_\infty \left(\epsilon_4 + \epsilon_1 d_2\right) \\ &= \epsilon_5, \end{aligned} \tag{2.52}$$

where the third inequality holds because $N_{\text{hef}}+1 \geq e/d_2$ and event A_2 holding true gives $\|\boldsymbol{e}^{(N_{\text{hef}}+1)}\|_\infty \leq \epsilon_1 e$, and the fourth inequality comes from the definition

$$\epsilon_5 = |V_b| \|\boldsymbol{u}^\top \boldsymbol{R}^{-1}\|_\infty \cdot \left(\epsilon_4 + \epsilon_1 d_2\right).$$

We recast (2.52) as

$$\frac{1-\epsilon_5}{\boldsymbol{u}^\top \boldsymbol{R}^{-1} \boldsymbol{u}} < \frac{e}{N_{\text{hef}}+1} < \frac{1+\epsilon_5}{\boldsymbol{u}^\top \boldsymbol{R}^{-1} \boldsymbol{u}}. \tag{2.53}$$

We transform (2.49) into

$$\begin{aligned}
&\left\| \frac{1}{N_{\mathrm{hef}}+1} \sum_{n=1}^{N_{\mathrm{hef}}+1} \boldsymbol{v}^{(n)} - \frac{\boldsymbol{R}^{-1}\boldsymbol{e}^{(0)}}{N_{\mathrm{hef}}+1} \right\|_\infty \\
&\quad\le \|\boldsymbol{R}^{-1}\|_\infty \cdot \left\| \frac{-\boldsymbol{e}^{(N_{\mathrm{hef}}+1)} + \sum_{n=1}^{N_{\mathrm{hef}}+1} (\boldsymbol{s}^{(n)} - \bar{\boldsymbol{s}})}{N_{\mathrm{hef}}+1} \right\|_\infty \\
&\quad\le \|\boldsymbol{R}^{-1}\|_\infty \left(\left\| \frac{\boldsymbol{e}^{(N_{\mathrm{hef}}+1)}}{N_{\mathrm{hef}}+1} \right\|_\infty + \sum_{n=1}^{N_{\mathrm{hef}}+1} \frac{\|\boldsymbol{s}^{(n)} - \bar{\boldsymbol{s}}\|_\infty}{N_{\mathrm{hef}}+1} \right) \\
&\quad\le \|\boldsymbol{R}^{-1}\|_\infty \left(\frac{\epsilon_1 e}{e/d_2} + \epsilon_4 \right) \\
&\quad\le \|\boldsymbol{R}^{-1}\|_\infty (\epsilon_4 + \epsilon_1 d_2), \qquad (2.54)
\end{aligned}$$

where the third inequality is because $\|\boldsymbol{e}^{(N_{\mathrm{hef}}+1)}\|_\infty < \epsilon_1 e$, because $N_{\mathrm{hef}} + 1 \ge e/d_2$ and because of (2.48). Plugging (2.53) into (2.54), we have

$$\left| \frac{1}{N_{\mathrm{hef}}+1} \sum_{n=1}^{N_{\mathrm{hef}}+1} \boldsymbol{v}^{(n)} - \frac{\boldsymbol{R}^{-1}\boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{R}^{-1}\boldsymbol{u}} \right| < \epsilon_2 \boldsymbol{u}, \qquad (2.55)$$

where we set

$$\epsilon_2 = \left\|\boldsymbol{R}^{-1}\right\|_\infty (\epsilon_4 + \epsilon_1 d_2) + \frac{\epsilon_5 \|\boldsymbol{R}^{-1}\boldsymbol{u}\|_\infty}{\boldsymbol{u}^\top \boldsymbol{R}^{-1}\boldsymbol{u}}.$$

Then, (2.55) shows that A_3 occurs, and therefore $A_1' \cap A_2 \cap A_5 \subseteq A_1' \cap A_3$ if we choose $\epsilon_1, \epsilon_2, \epsilon_4, \epsilon_5$ properly. Hence, if $\lim_{e\to\infty} \mathbb{P}(A_1' \cap A_3) = 0$,

$$\lim_{e\to\infty} \mathbb{P}(A_1' \cap A_2 \cap A_5) = 0.$$

Moreover, because of the rule of additions of probabilities,

$$\begin{aligned}
\mathbb{P}(A_1' \cap A_2) + \mathbb{P}(A_5) &= \mathbb{P}(A_1' \cap A_2 \cap A_5) + \mathbb{P}((A_1' \cap A_2) \cup A_5) \\
&\le \mathbb{P}(A_1' \cap A_2 \cap A_5) + 1. \qquad (2.56)
\end{aligned}$$

Because $\lim_{e\to\infty} \mathbb{P}(A_5) = 1$ holds true using the weak law of large numbers with (2.1) and (2.2), we have $\lim_{e\to\infty} \mathbb{P}(A_1' \cap A_2) = 0$.

Chapter 3

Joint Selection of Base Stations and Routing

> First, there is a mountain, then there is no mountain, then there is.
>
> *Traditional Buddhist saying*

In the previous chapter, we discussed the scheme of virtually moving one BS. In this chapter, we discuss the general problem of virtually moving multiple BSs, where we adaptively select both the active BSs and routing. Due to the general formulation of this problem, the proposed algorithms in this chapter have merits beyond the WSNs. We envision that they can be used in many other networked systems including super-node selection in peer-to-peer networks and cooperative beam-forming in cellular networks.

3.1 Introduction

By virtually moving one BS, the available energy of all BSs is efficiently utlitized. However, in a large-scale WSN, having only one active BS is not energy efficient because the average hop-distance from RSNs to the only active BS is large. The energy consumption of RSNs quickly accumulates along the communication paths from RSNs to that BS. Consequently, the overall energy consumption of the WSN is high.

Therefore, for a high energy-efficiency in large-scale WSNs, we have to enable multiple BSs to be simultaneously active, or in other words, we have to *"virtually move multiple BSs"*. In this scheme, we have to consider the overall energy efficiency of both BSs and RSNs. There is a tradeoff between the energy consumption of the long-range communication and that of short-

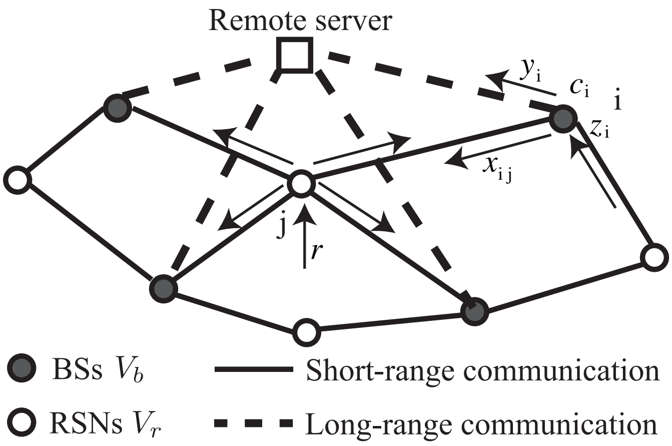


Figure 3.1: An example of the considered WSN where $|V_r| = 4$ and $|V_b| = 4$. The data rate on any short-range communication link (i, j) from i to j is denoted by x_{ij}, and the data rate on any long-range communication link from BS $i \in V_b$ to the remote server is denoted by y_i. The data generating rate of every RSN is r. The binary variable $z_i = 1$ if BS $i \in V_b$ is active and $z_i = 0$ otherwise. The energy-consumption rate of any RSN $i \in V_b \cup V_r$ is denoted by c_i.

range communication. The more the BSs are activated, the higher the energy consumption of long-range communication, and the lower the energy consumption of short-range communication.

To efficiently use the available energy of all BSs and RSNs, we have to address the following problems: i) When, how many and which active BSs should we use at each time? and ii) how should we adapt the routing among RSNs when different subsets of BSs are active? Therefore, we need to adaptively select both the active BSs and routing for RSNs.

This general problem of virtually moving BSs is much harder than the traditional maximum-lifetime routing problem (e.g., [16]), because active BSs consume a large and fixed amount of energy for using long-range communications as shown in Figure 1.2b. This fixed energy consumption makes the problem combinatorial. The traditional Garg-Konemann typed algorithms [27] cannot be directly applied here, as they only deal with linear-packing problems.

In this chapter, we propose a novel scheme that transforms the problem into a sequence of *uncapacitated facility-location* (UFL) problems [35] by using the constrained gradient method [24]. Under very mild conditions, the obtained UFL problems have approximate solutions with an approximation ratio of 1.61 [35]. By adaptively solving the sequence of UFL problems, our proposed algorithm yields a lifetime at least 62% of the optimum. The proposed algorithm is adaptive, has low computational complexity and uses only easily available information as input. It is therefore a perfect fit for the WSN paradigm.

Through extensive simulations, we show that virtually moving multiple BSs is more energy efficient compared to other existing schemes for organizing WSNs. We also show that the achieved lifetime of the proposed scheduling algorithm is always close to the optimum.

3.2 System Model

As shown in Figure 3.1, we model the WSN as a directed graph whose vertices are a set of BSs V_b, a set of RSNs V_r, and a remote server S. The edges in the directed graph consist of (i) short-range communication links among all BSs and RSNs $V_b \cup V_r$, and (ii) long-range communication links from BSs V_b to the remote server S. We now introduce the definitions for data communication and energy consumption in the considered WSN.

3.2.1 Data Communication

Each RSN $i \in V_r$ generates data with a constant rate $r \in \mathbb{R}^+$, whereas BSs do not generate data. All data have to be transmitted to active BSs by short-range and multi-hop communications. Active BSs forward the collected data to a remote server S via long-range communication. Although active BSs are adaptively changed, RSNs always actively sense and upload data to the remote server via active BSs. Let $x_{ij} \in \mathbb{R}_0^+$ be the data rate sent on the short-range communication link from i to j $(i, j \in V_b \cup V_r)$. Let $y_i \in \mathbb{R}_0^+$ be the data rate sent on the long-range communication link from BS $i \in V_b$ to the remote server S. Because the rate of data inflow and outflow should be balanced both on RSNs and on BSs, we have

$$\begin{cases} \sum_{j \in V_b \cup V_r} x_{ij} = \sum_{j \in V_b \cup V_r} x_{ji} + r, \ \forall i \in V_r, \\ \sum_{j \in V_b \cup V_r} x_{ij} + y_i = \sum_{j \in V_b \cup V_r} x_{ji}, \ \forall i \in V_b, \\ x_{ij} \geq 0, \ \ i, j \in V_b \cup V_r, \\ y_i \geq 0, \ \ i \in V_b. \end{cases} \tag{3.1}$$

Let z_i $(i \in V_b)$ be a binary variable indicating the state of BS i : if BS i is active, $z_i = 1$, and otherwise $z_i = 0$. When BS i is passive, its data rate of long-range communication y_i should be 0. Therefore,

$$y_i \leq |V_r| r z_i, \tag{3.2}$$

where $|V_r|r$ is the total data rate transmitted in the whole WSN. If BS i is active, (3.2) poses no constraint on y_i.

3.2.2 Energy Consumption

We denote the energy-consumption rate of either a RSN or a BS $i \in V_b \cup V_r$ by c_i $(c_i \in \mathbb{R}^+)$.

(i) On the one hand, for a RSN,

$$c_i = c_{\mathrm{c}} + c_{\mathrm{st}} \sum_{j \in V_b \cup V_r} x_{ij} + c_{\mathrm{sr}} \sum_{j \in V_b \cup V_r} x_{ji}, \forall i \in V_r, \tag{3.3}$$

where c_{c} denotes the constant energy-consumption rate for sensing and initiating the short-range communication, and c_{st}, c_{sr} denote the energy-consumption rates for transmitting and receiving unit data via short-range communication, respectively.

(ii) On the other hand, for a BS,

$$c_i = c_{\mathrm{c}} + c_{\mathrm{st}} \sum_{j \in V_b \cup V_r} x_{ij} + c_{\mathrm{sr}} \sum_{j \in V_b \cup V_r} x_{ji} + c_{\mathrm{lc}} z_i + c_{\mathrm{lt}} y_i, \forall i \in V_b, \tag{3.4}$$

where c_{lc} denotes the fixed energy-consumption rate for initiating and closing long-range communication and c_{lt} denotes the energy-consumption rate for actually transmitting unit data via long-range communication.

Let the initially available energy of each RSN or BS i ($i \in V_b \cup V_r$) be denoted by e_i.

For simplicity of discussion, we group $\{x_{ij}\}_{i,j \in V_b \cup V_r}$ into a vector $\boldsymbol{x}$, group $\{y_i\}_{i \in V_b}$ into a vector $\boldsymbol{y}$, group $\{z_i\}_{i \in V_b}$ into a vector $\boldsymbol{z}$, group $\{c_i\}_{i \in V_b \cup V_r}$ into a vector $\boldsymbol{c}$ and group $\{e_i\}_{i \in V_b \cup V_r}$ into a vector $\boldsymbol{e}$.

Notice that the state of a WSN is characterized by the data rates of short-range communication $\boldsymbol{x}$, the data rates of long-range communication $\boldsymbol{y}$, the selection of active BSs $\boldsymbol{z}$ and the energy-consumption rates $\boldsymbol{c}$. We call the four-tuple $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{c})$ a *configuration* of the network.

3.3 Problem Formulations

Scheduling the virtual movement of BSs is equivalent to finding a set of configurations and finding the time durations. In this section, we first describe the optimization space for selecting the configurations. Then, we formally formulate the scheduling problem, namely, the *virtually-moving BSs problem.* At the end of this section, we show why our proposed scheme is more energy-efficient than the previously proposed schemes [4, 7, 29, 59, 69].

3.3.1 The Optimization Space

We denote the set of all candidate configurations by

$$\mathcal{L} = \{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{c}) \mid (3.1), (3.2), (3.3) \text{ and } (3.4) \text{ hold}\}.$$

Because $\boldsymbol{z}$ is a binary vector taking $2^{|V_b|} - 1$ possible values (there should be at least one active BS, therefore we cannot have $\boldsymbol{z} = \boldsymbol{0}$), we can separate $\mathcal{L}$ into $2^{|V_b|} - 1$ subsets based on different values of $\boldsymbol{z}$. We denote the possible values of $\boldsymbol{z}$ by $\boldsymbol{z}^{[1]}$, $\boldsymbol{z}^{[2]}$, $\cdots$, $\boldsymbol{z}^{[2^{|V_b|}-1]}$ with a non-decreasing order of the number of active BSs. In particular, the first $|V_b|$ elements, $\boldsymbol{z}^{[1]}, \boldsymbol{z}^{[2]}, \cdots \boldsymbol{z}^{[|V_b|]}$ denote the BSs selections where only one BS is active. For any $1 \leq k \leq 2^{|V_b|} - 1$, we denote by $\mathcal{L}^{[k]}$ the subset of $\mathcal{L}$ satisfying $\boldsymbol{z} = \boldsymbol{z}^{[k]}$:

$$\mathcal{L}^{[k]} = \{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}^{[k]}, \boldsymbol{c}) \mid (3.1), (3.2), (3.3), (3.4) \text{ hold}\},$$

which is a simplex because constraints (3.1), (3.2), (3.3), (3.4) are linear after fixing $\boldsymbol{z} = \boldsymbol{z}^{[k]}$. Decomposing $\mathcal{L} = \bigcup_{k=1}^{2^{|V_b|}-1} \mathcal{L}^{[k]}$ enables us to simplify the optimization problem, as will be explained in the following section.

3.3.2 The Virtually-Moving BSs Problem

Because $\mathcal{L}^{[k]}$ is a simplex for all $1 \leq k \leq 2^{|V_b|} - 1$, the convex combination of any two configurations in $\mathcal{L}^{[k]}$ is still in $\mathcal{L}^{[k]}$. Let $(\boldsymbol{x}^{[1]}, \boldsymbol{y}^{[1]}, \boldsymbol{z}^{[k]}, \boldsymbol{c}^{[1]}) \in \mathcal{L}^{[k]}$ and $(\boldsymbol{x}^{[2]}, \boldsymbol{y}^{[2]}, \boldsymbol{z}^{[k]}, \boldsymbol{c}^{[2]}) \in \mathcal{L}^{[k]}$

be used for time durations $t^{[1]}$ and $t^{[2]}$. Using these two configurations is equivalent to using their convex combination

$$\left(\frac{t^{[1]}\boldsymbol{x}^{[1]}+t^{[2]}\boldsymbol{x}^{[2]}}{t^{[1]}+t^{[2]}}, \frac{t^{[1]}\boldsymbol{y}^{[1]}+t^{[2]}\boldsymbol{y}^{[2]}}{t^{[1]}+t^{[2]}}, \boldsymbol{z}^{[k]}, \frac{t^{[1]}\boldsymbol{c}^{[1]}+t^{[2]}\boldsymbol{c}^{[2]}}{t^{[1]}+t^{[2]}}\right),$$

which is still in $\mathcal{L}^{[k]}$, for a time duration $t^{[1]}+t^{[2]}$. This further implies that any number of configurations in the same set $\mathcal{L}^{[k]}$ can be substituted by only one configuration—their convex combination.

Therefore, in the virtually-moving BSs problem, we seek at most $2^{|V_b|}-1$ configurations: For each $1 \le k \le 2^{|V_b|}-1$, we select a configuration $(\boldsymbol{x}^{[k]}, \boldsymbol{y}^{[k]}, \boldsymbol{z}^{[k]}, \boldsymbol{c}^{[k]}) \in \mathcal{L}^{[k]}$ and a time duration $t^{[k]}$, such that the total lifetime $\sum_{k=1}^{2^{|V_b|}-1} t^{[k]}$ is maximized given the initially available energy $\boldsymbol{e}$:

$$\begin{aligned} \max_{\{\boldsymbol{x}^{[k]},\boldsymbol{y}^{[k]},\boldsymbol{z}^{[k]},\boldsymbol{c}^{[k]},t^{[k]}\}_{k=1}^{2^{|V_b|}-1}} \quad & \sum_{k=1}^{2^{|V_b|}-1} t^{[k]} \\ \text{s.t.} \quad & \sum_{k=1}^{2^{|V_b|}-1} t^{[k]}\boldsymbol{c}^{[k]} \le \boldsymbol{e}, \\ & \left(\boldsymbol{x}^{[k]}, \boldsymbol{y}^{[k]}, \boldsymbol{z}^{[k]}, \boldsymbol{c}^{[k]}\right) \in \mathcal{L}^{[k]}, \forall 1 \le k \le 2^{|V_b|}-1, \\ & t^{[k]} \ge 0, \forall 1 \le k \le 2^{|V_b|}-1. \end{aligned} \tag{3.5}$$

We denote the optimal lifetime of (3.5) by T^*. To differentiate with other schemes that will be mentioned below, we call this scheme *MultiMove*, which schedules the virtual movement of multiple BSs by solving problem (3.5).

3.3.3 Comparisons to Other Schemes

In the following, we compare MultiMove with some other existing schemes.

MultiFixed [7, 29, 59]: This scheme selects a subset of BSs to be always active and selects the routes of short-range communications. This boils down to selecting only one configuration $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{c}) \in \mathcal{L}$, so that the lifetime of the WSN is maximized:

$$\begin{aligned} \max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{c},t} \quad & t \\ \text{s.t.} \quad & (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{c}) \in \mathcal{L}, \\ & t\boldsymbol{c} \le \boldsymbol{e}, \end{aligned} \tag{3.6}$$

where t denotes the lifetime of the network, and $t\boldsymbol{c}$ denotes the total energy consumption that should be no larger than the initially available energy $\boldsymbol{e}$.

OneMove [4, 69]: In this scheme, we schedule the virtual movement of one BS on a finite set of candidate locations for maximizing lifetime. This is equivalent to selecting at most $|V_b|$ configurations: for each $1 \le k \le |V_b|$ (in contrast to $1 \le k \le 2^{|V_b|}-1$ as in the MultiMove scheme), we select one configuration $(\boldsymbol{x}^{[k]}, \boldsymbol{y}^{[k]}, \boldsymbol{z}^{[k]}, \boldsymbol{c}^{[k]}) \in \mathcal{L}^{[k]}$ in each set $\mathcal{L}^{[k]}$ with the time

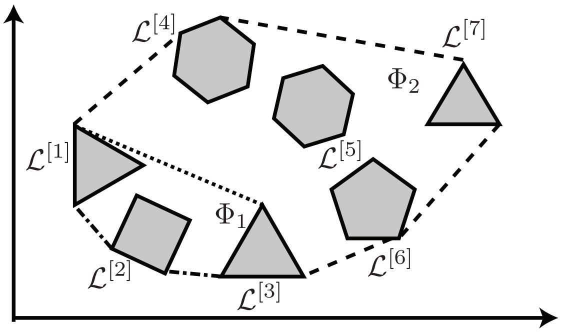


Figure 3.2: Optimization spaces of different schemes when the WSN has three BSs. The whole space for candidate configurations $\mathcal{L}$ is the union of simplexes $\mathcal{L}^{[1]}$, $\mathcal{L}^{[2]}$, $\cdots$, $\mathcal{L}^{[7]}$ which are denoted by the grey polygons. The optimization spaces of MultiFixed is $\mathcal{L}$. The optimization space of OneMove is illustrated by the space $\Phi_1 = \text{conv}(\bigcup_{\text{k}=1}^{3} \mathcal{L}^{[\text{k}]})$, whose edges are denoted by the dotted lines. The optimization space of MultiMove is illustrated by the space $\Phi_2 = \text{conv}(\mathcal{L})$, whose edges are denoted by the dashed lines. The common edges of Φ_1 and Φ_2 are denoted by dash-dot lines. We see that MultiMove has an optimization space larger than both MultiFixed and OneMove.

duration $t^{[k]}$, such that the total lifetime $\sum_{k=1}^{|V_b|} t^{[k]}$ is maximized:

$$\begin{aligned}
\max_{\{\boldsymbol{x}^{[k]},\boldsymbol{y}^{[k]},\boldsymbol{z}^{[k]},\boldsymbol{c}^{[k]},t^{[k]}\}_{k=1}^{|V_b|}} \quad & \sum_{k=1}^{|V_b|} t^{[k]} \\
\text{s.t.} \quad & \sum_{k=1}^{|V_b|} t^{[k]}\boldsymbol{c}^{[k]} \leq \boldsymbol{e}, \\
& \left(\boldsymbol{x}^{[k]}, \boldsymbol{y}^{[k]}, \boldsymbol{z}^{[k]}, \boldsymbol{c}^{[k]}\right) \in \mathcal{L}^{[k]}, \forall 1 \leq k \leq |V_b|, \\
& t^{[k]} \geq 0, \forall 1 \leq k \leq |V_b|.
\end{aligned} \tag{3.7}$$

MultiMove is more energy-efficient than the other schemes mentioned above. In the MultiFixed scheme, we select one configuration in the optimization space $\mathcal{L}$. In the MultiMove scheme, we select one configuration $(\boldsymbol{x}^{[k]}, \boldsymbol{y}^{[k]}, \boldsymbol{z}^{[k]}, \boldsymbol{c}^{[k]}) \in \mathcal{L}^{[k]}$ and a time duration $t^{[k]}$ for any $1 \leq k \leq 2^{|V_b|} - 1$. This virtually creates a new configuration whose average energy-consumption rates are $\sum_{k=1}^{2^{|V_b|}-1} \boldsymbol{c}^{[k]} t^{[k]} / \sum_{k=1}^{2^{|V_b|}-1} t^{[k]}$, a convex combination of $\boldsymbol{c}^{[1]}$, $\boldsymbol{c}^{[2]}$, $\cdots$, $\boldsymbol{c}^{[2^{|V_b|}-1]}$. Therefore, the optimization space of MultiMove for selecting configurations is expanded from $\mathcal{L}$ to its convex hull conv$(\mathcal{L})$. Similarly, in the OneMove scheme, the optimization space is expanded from $\bigcup_{k=1}^{|V_b|} \mathcal{L}^{[k]}$ (the subset of configurations where only one BS is active) to $\text{conv}(\bigcup_{\text{k}=1}^{|\text{V}_\text{b}|} \mathcal{L}^{[\text{k}]})$. We illustrate the optimization spaces of these different schemes in Figure 3.2 with an example of a WSN with three BSs.

3.4 Complexity Analysis

In this section, we will analyze the complexity of the virtually-moving BSs problem.

Unsurprisngly, the problem is "very hard" because (i) there is an exponential number of candidate subsets of active BSs, and (ii) active BSs have the fixed energy-consumption rate c_{lc} for using long-range communication, which makes the problem non-convex. To formally evaluate the hardness of the problem, we define the **decision version of the virtually-moving BSs problem** as follows: Given the topology of the WSN, the constant parameters c_{c}, c_{st}, c_{sr}, c_{lc}, c_{lt}, the data generating rate r, the initially available energy $\boldsymbol{e}$, the sets of possible configurations $\{\mathcal{L}^{[k]}\}_{k=1}^{2^{|V_b|}-1}$ defined by (3.1), (3.2), (3.3) and (3.4), and a number T, does there exist a configuration in each simplex $(\boldsymbol{x}^{[k]}, \boldsymbol{y}^{[k]}, \boldsymbol{z}^{[k]}, \boldsymbol{c}^{[k]}) \in \mathcal{L}^{[k]}$ and a time duration $t^{[k]}$ for each $1 \leq k \leq 2^{|V_b|} - 1$, such that the lifetime of the network $\sum_{k=1}^{2^{|V_b|}-1} t^{[k]} \geq T$ under the energy constraint $\sum_{k=1}^{2^{|V_b|}-1} t^{[k]}\boldsymbol{c}^{[k]} \leq \boldsymbol{e}$?

Theorem 3.1

The virtually-moving base-stations problem is NP-complete.

Proof.

First of all, the decision version of the problem is NP because we can verify a valid instance in polynomial time. Now we show that the problem is NP-hard. We reduce the 3-SAT problem [9] to the virtually-moving BSs problem. Consider a 3-SAT instance with l variables a_1, a_2, $\cdots$, a_l and m clauses b_1, b_2, $\ldots$, b_m (Notice that each clause is a 3-element subset of $\{a_i, \bar{a}_i\}_{1\leq i\leq l}$). We define an $2l+1$-th variable w other than $\{a_i, \bar{a}_i\}_{1\leq i\leq l}$ and define a few sets:

— $P_i = \{a_i, \bar{a}_i\} \cup \{b_j | b_j \text{ contains } a_i\}$, $\forall 1 \leq i \leq l$,

— $Q_i = \{a_i, \bar{a}_i\} \cup \{b_j | b_j \text{ contains } \bar{a}_i\}$, $\forall 1 \leq i \leq l$,

— $W_1 = \{w\} \cup \{a_1, a_2, \cdots a_l\} \cup \{b_1, b_2, \cdots b_m\}$,

— $W_2 = \{w\} \cup \{\bar{a}_1, \bar{a}_2, \cdots \bar{a}_l\}$.

We construct an instance of the virtually-moving BSs problem as follows. Let P_1, P_2, $\cdots$, P_l, Q_1, Q_2, $\cdots$, Q_l, W_1, W_2 be associated with a BS each, and let a_1, a_2, $\cdots$, a_l, b_1, b_2, $\ldots$, b_m, w be associated with a RSN each. Let each BS have a long-range communication link to the remote server S. Excluding the remote server S, the graph of the WSN is bipartite where every edge connects a BS to a RSN, and this edge only exists when the set associated with the BS contains the variable associated with the RSN. Let the initially available energy of each BS be 1 and let that of each RSN be 2. Let the constants for energy consumption be $c_c = 0$, $c_{\text{st}} = 1$, $c_{\text{sr}} = 0$, $c_{\text{lc}} = 1$, $c_{\text{lt}} = 0$, and let the data generating rate $r = 1$. Then, we set the tentative lifetime $T = 2$.

Because the data generating rate $r = 1$, $c_{\text{st}} = 1$ and because of (3.3), the energy-consumption rate of any RSN is at least 1. Remember that the initially available energy of any RSN is 2, the lifetime of the considered virtually-moving BSs problem is at most $T = 2$. In the following, we will show that if the constructed virtually-moving BSs problem achieves the lifetime $T = 2$, the original 3-SAT problem is satisfiable. We start with four deductions, given that the lifetime $T = 2$ is achievable:

(i) To be active, BSs W_1 and W_2 alternate, and each for a time duration 1. First, because of (3.4) and because $c_{\text{lt}} = 1$, the energy-consumption rate of any BS is at least 1, therefore any BS has an active time duration at most 1 given the initially available energy 1. Then, because BSs W_1 and W_2 are the only neighbours of RSN w and because they have to serve

w for a whole lifetime 2, both BSs W_1 and W_2 have to be active for a time duration 1 and they cannot be simultaneously active.

(ii) The energy-consumption rate of every RSN is 1 at any time. On the one hand, it is at least 1, as shown above. On the other hand, it cannot exceed 1, because every RSN has to sustain a lifetime 2 with an initially available energy 2.

(iii) Passive BSs do not forward data for other RSNs at any time, otherwise, at least one RSN would receive data from passive BSs. Using (3.3), we see that this would result in an energy-consumption rate exceeding 1, which would contradict (ii).

(iv) In the configuration where BS W_2 is active (BS W_1 is passive because of (i)), one and only one BS between BS P_i and BS Q_i is active for any $1 \le i \le l$. First, at least one of them has to be active in order to forward data from RSN a_i because they are the only neighbours of a_i. Then, they cannot be simultaneously active, because they have to forward data for RSN a_i for a lifetime 2 with the total available energy 2 and because $c_{\text{lt}} = 1$.

Because of deduction (i), we have a configuration in which BS W_2 is active and BS W_1 is passive. We use this configuration to construct a valid assignment for the original 3-SAT problem. Because of deduction (iv), one and only one BS between BSs P_i and Q_i is active for any $1 \le i \le l$ in that configuration. This enables us to construct an assignment for the 3-SAT problem by setting the variable $a_i = 1$ if BS P_i is active and setting $a_i = 0$ if BS Q_i is active. Then, because of deduction (iii), each RSN in $b_1, b_2, \cdots b_m$ connects to at least one active BS because passive BSs do not forward data from RSNs. Therefore, every clause b_j $(1 \le j \le m)$ in the 3-SAT problem is satisfied: If RSN b_j connects to the active BS P_i, clause b_j is satisfied because it contains variable a_i and $a_i = 1$; if RSN b_j connects to the active BS Q_i, clause b_j is also satisfied because it contains variable $\bar{a}_i$ and $a_i = 0$. In summary, if we can solve the constructed virtually-moving BSs problem, we can also solve the original 3-SAT problem. Therefore, the virtually-moving BSs problem is NP-hard as 3-SAT is known to be NP-hard. □

3.5 Scheduling Algorithm

In this section, we will propose a scheme for adaptively scheduling the virtual movement of multiple BSs. The proposed algorithm is computationally light, only requires easily available information as input, and guarantees, under mild conditions, a network lifetime at least 62% of the optimal one.

The adaptive scheduling scheme works as follows. Time is discretized into slots of length τ, during which we use only one configuration. Before the start of each time slot, the remote server collects the information about the current available energy of all BSs and all RSNs. Using only this information as input, the remote server selects the configuration to be used in the next time slot and notifies all BSs. As τ is usually much larger than the sampling interval of RSNs, this scheme has negligible overhead for collecting the required information and distributing the configuration determined by the server.

We denote the available energy of all RSNs and BSs at time n $(n \in \mathbb{N})$ by $\boldsymbol{e}^{(n)}$. In particular, the initially available energy $\boldsymbol{e}^{(0)} = \boldsymbol{e}$. Denote the configuration selected for time slot n by $(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}, \boldsymbol{z}^{(n)}, \boldsymbol{c}^{(n)}) \in \mathcal{L}$. The available energy evolves according to

$$\boldsymbol{e}^{(n)} = \boldsymbol{e}^{(n-1)} - \tau \boldsymbol{c}^{(n)}. \tag{3.8}$$

If we sum up the iterative equation (3.8) from time 0 to time n and use $\boldsymbol{e}^{(0)} = \boldsymbol{e}$, we have

$$\boldsymbol{e}^{(n)} = \boldsymbol{e} - \tau \sum_{t=1}^{n} \boldsymbol{c}^{(t)}. \tag{3.9}$$

Denote by N^* the maximum number of time slots before the first RSN depletes its energy,

$$N^* = \max\{n | \boldsymbol{e}^{(n)} \geq \boldsymbol{0}\}, \tag{3.10}$$

and the lifetime of the WSN is τN^*.

In the following, to find the optimal configurations, we first propose to use the constrained gradient method [24] and to separate the virtually-moving BSs problem into a sequence of subproblem called the *min-weight configuration problems*. Then, we will discuss how to solve the min-weight configuration problems.

3.5.1 The Constrained Gradient Method

First of all, we define auxiliary variables

$$p^{[k]} = t^{[k]} \Big/ \sum_{k=1}^{2^{|V_b|}-1} t^{[k]}, \tag{3.11}$$

$$\boldsymbol{\theta} = \sum_{k=1}^{2^{|V_b|}-1} p^{[k]} \boldsymbol{c}^{[k]}. \tag{3.12}$$

Here $p^{[k]}$ denotes the fraction of time for using configuration $(\boldsymbol{x}^{[k]}, \boldsymbol{y}^{[k]}, \boldsymbol{z}^{[k]}, \boldsymbol{c}^{[k]})$ for all $1 \leq k \leq 2^{|V_b|} - 1$ and $\boldsymbol{\theta}$ denotes the average energy decrease rates for all RSNs and BSs. Next, we recast problem (3.5) as the auxiliary problem

$$\begin{aligned} \min_{\{\boldsymbol{x}^{[k]}, \boldsymbol{y}^{[k]}, \boldsymbol{z}^{[k]}, \boldsymbol{c}^{[k]}, p^{[k]}\}_{k=1}^{2^{|V_b|}-1}, \boldsymbol{\theta}} \quad & F(\boldsymbol{\theta}) = \max_{i \in V_b \cup V_r} \theta_i / e_i \\ \text{s.t.} \quad & \boldsymbol{\theta} = \sum_{k=1}^{2^{|V_b|}-1} p^{[k]} \boldsymbol{c}^{[k]}, \\ & \left(\boldsymbol{x}^{[k]}, \boldsymbol{y}^{[k]}, \boldsymbol{z}^{[k]}, \boldsymbol{c}^{[k]}\right) \in \mathcal{L}^{[k]}, \forall 1 \leq k \leq 2^{|V_b|} - 1, \\ & \sum_{k=1}^{2^{|V_b|}-1} p^{[k]} = 1, \\ & p^{[k]} \geq 0, \forall 1 \leq k \leq 2^{|V_b|} - 1, \end{aligned} \tag{3.13}$$

where θ_i is the i-th element of $\boldsymbol{\theta}$. Note that the objective value of (3.13) is the inverse of the objective value in (3.5).

We relax the objective function $F(\boldsymbol{\theta})$ in problem (3.13) into a differentiable function

$$f(\boldsymbol{\theta}) = \frac{1}{\alpha} \log \left(\sum_{i \in V_b \cup V_r} \exp\left(\frac{\alpha \theta_i}{e_i} \right) \right) \tag{3.14}$$

Algorithm 3.1: The constrained gradient method for selecting configurations

```
Input: The parameters $c_c, c_{st}, c_{sr}, c_{lc}, c_{lt}, r$, the initially available energy $\boldsymbol{e}$, and a
       $\beta$-approximate algorithm for solving the min-weight configuration problem (3.18).
Output: $N^*$, $\{(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}, \boldsymbol{z}^{(n)}, \boldsymbol{c}^{(n)})\}_{1\le n\le N^*}$.
Initialize $\boldsymbol{\theta}^{(0)} \longleftarrow \boldsymbol{0}$, $\boldsymbol{e}^{(0)} \longleftarrow \boldsymbol{e}$, $n \longleftarrow 1$.
while $\boldsymbol{e}^{(n-1)} \ge \boldsymbol{0}$ do
    Calculate $\boldsymbol{\theta}^{(n-1)}$ from $\boldsymbol{e}^{(n-1)}$ and (3.16).
    Calculate the gradient $\boldsymbol{\nabla}^\top f(\boldsymbol{\theta}^{(n-1)})$ using (3.17).
    Select configuration $(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}, \boldsymbol{z}^{(n)}, \boldsymbol{c}^{(n)})$ by using a $\beta$-approximate algorithm to solve
    the min-weight configuration problem (3.18) where $\boldsymbol{\lambda} = \boldsymbol{\nabla}^\top f(\boldsymbol{\theta}^{(n-1)})$.
    Update $\boldsymbol{e}^{(n)} \longleftarrow \boldsymbol{e}^{(n-1)} - \boldsymbol{c}^{(n)}$.
    Update $n \longleftarrow n + 1$.
Set $N^* = n - 1$.
```

by introducing a real parameter $\alpha > 0$. We can check that

$$f(\boldsymbol{\theta}) - \frac{1}{\alpha}\log|V_b \cup V_r| \le F(\boldsymbol{\theta}) \le f(\boldsymbol{\theta}). \tag{3.15}$$

Therefore, using $f(\boldsymbol{\theta})$ as an approximation of $F(\boldsymbol{\theta})$ incurs an arbitrarily small loss of precision when α is large.

We define $\boldsymbol{\theta}^{(n)}$ as the average energy decrease rates during the first n time slots

$$\boldsymbol{\theta}^{(n)} = \frac{1}{n}\sum_{t=1}^{n} \boldsymbol{c}^{(t)} = \frac{\boldsymbol{e} - \boldsymbol{e}^{(n)}}{n\tau}, \tag{3.16}$$

where the second equality follows from (3.9). The purpose is to adaptively select the configurations of the WSN such that the sequence $\{\boldsymbol{\theta}^{(n)}\}_{n\in\mathbb{N}^+}$ gradually approaches the desired $\boldsymbol{\theta}$ in (3.13). Note that $\boldsymbol{\theta}^{(n-1)}$ is known at time slot n because $\boldsymbol{e}^{(n-1)}$ is already known and because of (3.16).

As shown in Algorithm 3.1, we iteratively take the following steps in each time slot $n \in \mathbb{N}^+$ as long as $\boldsymbol{e}^{(n-1)} \ge \boldsymbol{0}$:

(i) First, we calculate the energy decrease rates in the first $n-1$ time slots $\boldsymbol{\theta}^{(n-1)}$ using $\boldsymbol{e}^{(n-1)}$ and using (3.16), and we calculate the gradient of the relaxed objective function $\boldsymbol{\nabla}^\top f(\boldsymbol{\theta}^{(n-1)})$ whose i-th element ($i \in V_b \cup V_r$) is

$$\nabla_i^\top f\left(\boldsymbol{\theta}^{(n-1)}\right) = \frac{\exp\left(\alpha\theta_i^{(n-1)}/e_i\right)}{e_i \sum_{i\in V_b\cup V_r} \exp\left(\alpha\theta_i^{(n-1)}/e_i\right)}. \tag{3.17}$$

Here, e_i in (3.17) is the initially available energy of node i.

(ii) Then, we define a weight vector $\boldsymbol{\lambda} = \boldsymbol{\nabla}^\top f(\boldsymbol{\theta}^{(n-1)})$ and select a configuration to be used $(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}, \boldsymbol{z}^{(n)}, \boldsymbol{c}^{(n)})$ by solving

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{c}} \quad & \boldsymbol{\lambda}^\top \boldsymbol{c} \\ \text{s.t.} \quad & (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{c}) \in \mathcal{L}, \end{aligned} \tag{3.18}$$

which is termed as the *min-weight configuration problem.*

When the algorithm terminates, the lifetime of the network is τN^*, as defined in (3.10). Because N^* is dependent on τ and α, we use $N^*(\tau, \alpha)$ for N^* to show explicitly the dependencies in the following.

The min-weight configuration problem is still combinatorial because there could be an exponential number of candidate configurations in $\mathcal{L}$. Nontheless, in Theorem 3.2, we will show that if the min-weight configuration problem has a β-approximate algorithm (β can be either a constant or a big-O function of $|V_b \cup V_r|$), the virtually-moving BSs problem also has a β-approximate algorithm.

Theorem 3.2

Using Algorithm 3.1 for selecting the configurations $\{(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}, \boldsymbol{z}^{(n)}, \boldsymbol{c}^{(n)})\}_{n\in\mathbb{N}^+}$ *with a* β*-approximation algorithm for solving the min-weight configuration problem (3.18), the achieved lifetime* $\tau N^*(\tau, \alpha)$ *satisfies*

$$\frac{\tau N^*(\tau, \alpha) + \tau}{T^*} > \frac{1}{\beta + \gamma_1(\alpha) + \gamma_2(\tau, \alpha)},$$

where

$$\gamma_1(\alpha) = \frac{\beta T^* \ln |V_b \cup V_r|}{\alpha},$$
$$\gamma_2(\tau, \alpha) = \frac{\alpha\tau |V_b \cup V_r|^2 c_{\max}^3 T^* (\ln \frac{e_{\min}}{\tau c_{\max}} + 1)}{2e_{\min}^3},$$

with $e_{\min}$ *denoting the minimum element of* $\boldsymbol{e}$ *and* $c_{\max}$ *denoting an upper-bound on the energy-consumption rate of any BS or any RSN*

$$c_{\max} = c_{\text{c}} + c_{\text{lc}} + (c_{\text{st}} + c_{\text{sr}} + c_{\text{lt}})|V_b \cup V_r| r.$$

If we set $\tau \to 0$ *and then set* $\alpha \to \infty$*, we have*

$$\lim_{\alpha\to\infty} \lim_{\tau\to 0} \frac{\tau N^*(\tau, \alpha)}{T^*} \geq \frac{1}{\beta}.$$

Details of the proof can be found in Appendix 3.A.1.

3.5.2 The Min-Weight Configuration Problem

We propose an algorithm for solving the min-weight problem with a guaranteed approximation ratio $\beta = 1.61$ by building on previous results in the uncapacitated facility-location (UFL) problems [35].

The UFL problem can be stated as follows. Let $\mathcal{F}$ be a set of facilities and let $\mathcal{C}$ be a set of customers. Let $h_u > 0$ be the fixed cost for opening the facility $u \in \mathcal{F}$ and let $g_{uv} > 0$ be the service cost for each $u \in \mathcal{F}$ and $v \in \mathcal{C}$. We seek a subset of open facilities $\mathcal{F}_o \subseteq \mathcal{F}$, such that all

Table 3.1: Mapping from the min-weight configuration problem into an UFL problem

Facility set $\mathcal{F}$	BSs V_b
Open facility set $\mathcal{F}_o$	Active BSs $\{u \in V_b \vert z_u = 1\}$
Customer set $\mathcal{C}$	RSNs V_r
Facility cost h_u $(u \in \mathcal{F})$	$\lambda_u c_{\text{lc}}$, $u \in V_b$
Service cost g_{uv} $(u \in \mathcal{F}, v \in \mathcal{C})$	$L_{uv} + \lambda_v c_{\text{c}}$, where L_{uv} is the length of the shortest path from RSN $v \in V_r$ to S via BS $u \in V_b$ with the distance assignment (3.22).

customers are served and the sum of facility costs and service costs

$$\sum_{u \in \mathcal{F}_o} h_u + \sum_{v \in \mathcal{C}} \min_{u \in \mathcal{F}_o} g_{uv} \tag{3.19}$$

is minimized. A special type of UFL problems is called *metric uncapacitated facility-location* (metric-UFL) problems where the service costs are metric, that is,

$$g_{uv} + g_{u'v} + g_{u'v'} \geq g_{uv'}, \forall u, u' \in \mathcal{F}, \text{ and } v, v' \in \mathcal{C}. \tag{3.20}$$

In [35], Metric-UFL problems are solved by using an approximation algorithm with an approximation ratio 1.61.

We will convert the min-weight configuration problem (3.18) into an UFL problem with the mappings shown in Table 3.1:

(i) We set $\mathcal{F} = V_b$ and $\mathcal{C} = V_r$. The open facility set we are looking for is the set of active BSs

$$\mathcal{F}_o = \{u : z_u = 1, u \in V_b\}. \tag{3.21}$$

(ii) We assign a length d_{ij} for each edge of the WSN,

$$d_{ij} = \begin{cases} \lambda_i c_{\text{st}} r + \lambda_j c_{\text{sr}} r, & i, j \in V_b \cup V_r, \\ \lambda_i c_{\text{lt}} r, & i \in V_b, j = S, \\ \infty, & \text{othewise,} \end{cases} \tag{3.22}$$

where λ_i is the i-th element of the weight vector $\boldsymbol{\lambda}$.

(iii) For each RSN $v \in V_r$ and each BS $u \in V_b$, we calculate the shortest path from RSN v to the remote server S via BS u, which we denote by L_{uv}. We let the service cost g_{uv} for BS u forward the data packets from RSN $v \in V_r$ be

$$g_{uv} = L_{uv} + \lambda_v c_{\text{c}}, \tag{3.23}$$

and we let the facility cost h_u for activating BS $u \in V_b$ be

$$h_u = \lambda_u c_{\text{lc}}. \tag{3.24}$$

The UFL problem constructed from the mappings in Table 3.1 is not metric in general, because the length of a path under the distance assignment (3.22) is not symmetric: the length of the shortest path from RSN v to BS u is not equal to the reverse path from BS u to RSN v. In the following theorem, we show the conditions that ensure the UFL to be metric.

Theorem 3.3

Given the parameters c_{c}, c_{st}, c_{sr}, c_{lc}, c_{lt}, r, and a weight vector $\boldsymbol{\lambda}$, by using the mappings shown in Table 3.1, we construct an instance of the UFL problem from the min-configuration problem (3.18). The constructed UFL problem

- *has the same optimal objective value as that of the min-weight configuration problem (3.18), and*
- *is metric (3.20) under the condition that $c_{\mathrm{sr}} \le c_{\mathrm{st}} \le 2c_{\mathrm{lt}} + c_{\mathrm{sr}}$.*

Details of the proof can be found in Appendix 3.A.2.

We then use the 1.61-approximation algorithm proposed in [35] to solve the constructed UFL problem. For simplicity of discussion, we define the set of unconnected customers as $\mathcal{C}_u$. Let the two-tuple $(i, \mathcal{C})$ denote a star that is composed of a facility $i \in \mathcal{F}$ and a set of customers $\mathcal{D} \subseteq \mathcal{C}$ connected to facility i. The cost of the star $(i, \mathcal{D})$ is defined as the total cost divided by the number of unconnected customers in $\mathcal{D}$:

$$\mathrm{cost}(i, \mathcal{D}) = \begin{cases} \dfrac{h_i + \sum_{j \in \mathcal{D}} q_{ij}}{|\mathcal{C}_u \cap \mathcal{D}|}, & i \notin \mathcal{F}_o, \\ \dfrac{\sum_{j \in \mathcal{D}} q_{ij}}{|\mathcal{C}_u \cap \mathcal{D}|}, & i \in \mathcal{F}_o, \end{cases} \tag{3.25}$$

where

$$q_{ij} = \begin{cases} g_{ij}, & j \in \mathcal{C}_u, \\ \min(0, g_{ij} - \min_{k \in \mathcal{F}_o} g_{kj}), & j \notin \mathcal{C}_u. \end{cases}$$

The procedure of the algorithm is as follows:

(i) At the beginning, we set $\mathcal{F}_o = \emptyset$ and $\mathcal{C}_u = \mathcal{C}$.

(ii) As long as the set of unconnected customers $\mathcal{C}_u \neq \emptyset$, find the most cost-efficient star $(i, \mathcal{D})$ for all $i \in \mathcal{F}$ and $\mathcal{D} \subseteq \mathcal{C}$. Connect all customers in $\mathcal{D}$ to i and set $\mathcal{C}_u = \mathcal{C}_u \backslash \mathcal{D}$. If BS $i \notin \mathcal{F}_o$, activate BS i and $\mathcal{F}_o = \mathcal{F}_o \cup \{i\}$.

We note that in step (ii), although the number of stars is exponential with the number of customers, it is easy to find the most cost-effective one star. It is sufficient to consider stars $(i, \mathcal{D}_k^i)$ for $i \in \mathcal{F}$ and $k \in \{1, \cdots, |\mathcal{C}_u|\}$, where $\mathcal{D}_k^i$ denotes the set containing all connected customers $j \in \mathcal{C} \backslash \mathcal{C}_u$ whose $q_{ij} < 0$ and containing k unconnected customers $j \in \mathcal{C}_u$ with the smallest positive q_{ij}. Clearly, other stars cannot be more cost-effective (More details can be found in [35]).

We summarize the whole procedure for finding the approximate min-weight configuration as in Algorithm 3.2.

Theorem 3.4

Algorithm 3.2 guarantees to find a configuration $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{c}) \in \mathcal{L}$, such that the weight $\boldsymbol{\lambda}^\top \boldsymbol{c}$ is at most $\beta = 1.61$ times the optimal objective value of the min-weight configuration problem (3.18) under the condition that $c_{\mathrm{sr}} \le c_{\mathrm{st}} \le 2c_{\mathrm{lt}} + c_{\mathrm{sr}}$.

Algorithm 3.2: Approximation algorithm for solving the min-weight configuration problem

```
Input: The parameters $c_{\mathrm{c}}, c_{\mathrm{st}}, c_{\mathrm{sr}}, c_{\mathrm{lc}}, c_{\mathrm{lt}}, r$, and a weight vector $\boldsymbol{\lambda}$.
Output: A configuration $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{c}) \in \mathcal{L}$.
Assign the lengths of edges in the WSN as shown in (3.22).
Set $\mathcal{F} \longleftarrow V_b$ and set $\mathcal{C} \longleftarrow V_r$.
Calculate the shortest path $L_{uv}$ from each RSN $v \in \mathcal{C}$ to the remote server $S$ via each BS $u \in \mathcal{F}$.
Assign the service costs $g_{uv}$ $(u \in \mathcal{F}, v \in \mathcal{C})$ and the facility costs $h_u$ $(u \in \mathcal{F})$ as shown in (3.23) and (3.24), respectively.
Set $\mathcal{F}_o \longleftarrow \emptyset$ and set $\mathcal{C}_u \longleftarrow \mathcal{C}$.
while $\mathcal{C}_u \neq \emptyset$ do
    Find the most cost-effective star $(i, \mathcal{D})$ where $i \in \mathcal{F}$ and $\mathcal{D} \subseteq \mathcal{C}$ with the cost of the stars defined in (3.25).
    Connect all RSNs in $\mathcal{D}$ to BS $i$ along the shortest paths with edge lengths defined in (3.22). Set $\mathcal{C}_u \longleftarrow \mathcal{C}_u \backslash \mathcal{D}$.
    Enable BS $i$ to be active and set $\mathcal{F}_o \longleftarrow \mathcal{F}_o \cup \{i\}$.
```

In summary, the results in both Theorems 3.2 and 3.4 yield that our proposed algorithm achieves, under mild conditions, a lifetime at least 62% of the optimal one.

3.6 Simulations

In this section, we will evaluate the proposed scheme through extensive simulations.

We consider a 150 m × 150 m sensing field where we randomly generate a connected WSN with $|V_r| = 40$ and $|V_b| = 5$ (The default network size is small so that problem (3.5) can be solved without approximation). All RSNs generate data with a rate of $r = 1$ and BSs do not generate data. Data generated by RSNs are transmitted to active BSs by using short-range communication whose per-hop transmitting range is 40 m. Time is partitioned into slots with the length of a time slot $\tau = 1$. Let the constants of energy-consumption rates be $c_{\mathrm{sc}} = 1$, $c_{\mathrm{st}} = 1$, $c_{\mathrm{sr}} = 1$, $c_{\mathrm{lc}} = 10$ and $c_{\mathrm{lt}} = 1$ (We select these parameters to reflect the real energy consumptions of BSs and RSNs. More details can be found in our recent paper [69]). Let the initially available energy of all BSs be $e_i = 5000, \forall i \in V_b$ and let that of all RSNs be $e_i = 3000, \forall i \in V_r$. Let the parameter $\alpha = 10000$ in (3.14).

Here, we simulate five schemes: (i) *OneFixed*, the traditional scheme that uses one always-active BS, (ii) *MultiFixed* that uses multiple always-active BSs, where the locations of active BSs are optimized through (3.6), (iii) *OneMove* that schedules the virtual movement of one BS through (3.7), (iv) *MultiMove* that schedules the virtual movement of multiple BSs by solving (3.5) without approximation (considered as the optimum in the following), and (v) *MultiMove-A* that schedules the virtual movement of multiple BSs by using our proposed Algorithm 3.1 and Algorithm 3.2 to approximately solve (3.5). The details of MultiFixed, OneMove, and MultiMove are discussed in Section 3.3.3. Without the fixed energy-consumption of long-range communication, both MultiFixed and MultiMove are optimal; they will always activate all BSs and use the maximum lifetime routing [16]. Because problem (3.5) is computationally hard, MultiMove is only applicable when the network is small.

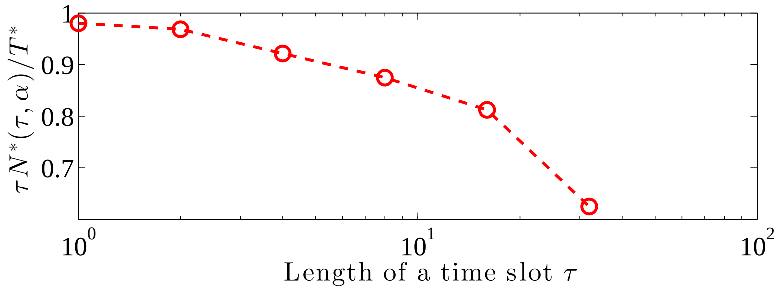


Figure 3.3: The ratio between the lifetime of MultiMove-A and that of MultiMove $\tau N^*(\tau, \alpha)/T^*$ versus the length of time slot τ. We see that the ratio decreases as τ increases. If τ is set small ($\tau < 10$), MultiMove-A yields a lifetime very close to the optimum achieved by MultiMove.

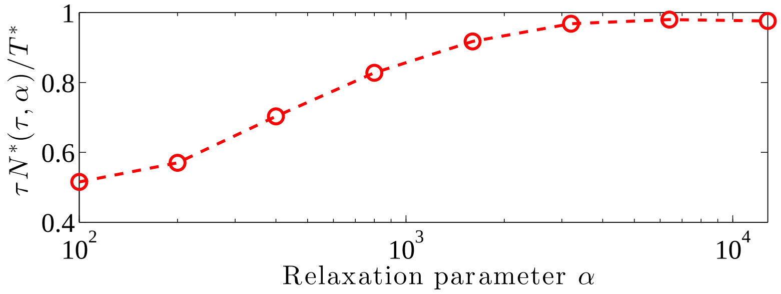


Figure 3.4: The ratio between the lifetime of MultiMove-A and that of MultiMove $\tau N^*(\tau, \alpha)/T^*$ versus the relaxation parameter α. We see that when α is sufficiently large ($\alpha > 1000$) , MultiMove yields a lifetime very close to the optimum.

3.6.1 Parameter Selection

As shown in Theorem 3.2, the performance of MultiMove-A depends on the parameters including the length of a time slot τ and the relaxation parameter α. We show how these parameters influence the ratio between the lifetime achieved by MultiMove-A and that by MultiMove $\tau N^*(\tau, \alpha)/T^*$.

In Figure 3.3, we show the ratio $\tau N^*(\tau, \alpha)/T^*$ versus the length of a time slot τ. When τ is small, MultiMove-A yields a lifetime very close to the optimum achieved by MultiMove, as the algorithm exploits the fine granularity of the time slot.

In Figure 3.4, we show the ratio $\tau N^*(\tau, \alpha)/T^*$ versus the relaxation parameter α in (3.14). When α is large, the relaxation of the objective function is precise, and MultiMove-A yields a lifetime very close to the optimum.

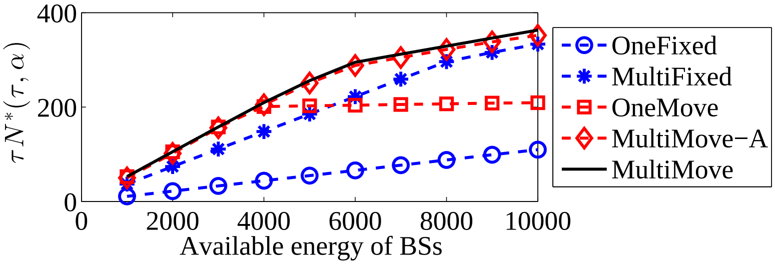


Figure 3.5: The network lifetime $\tau N^*(\tau, \alpha)$ versus the initially available energy of BSs when running different schemes..

3.6.2 Performance Comparison

To simulate different scenarios, we vary five parameters of the network: (i) the initially available energy of BSs, (ii) the number of BSs $|V_b|$, (iii) the number of RSNs $|V_r|$, (iv) the fixed energy-consumption rate of using long-range communication c_{lc}, and (v) the data rate of RSNs r. We summarize the simulation results as follows:

(i) In some scenarios, BSs cause bottlenecks in the network lifetime, *e.g.*, when the initially available energy of BSs is low, when $|V_b|$ is small, when $|V_r|$ is large, when c_{lc} is large and when r is small. In these scenarios, OneMove outperforms MultiFixed, because OneMove improves the energy efficiency of BSs by rotating one active BS among multiple BSs.

(ii) In other scenarios, RSNs cause bottlenecks in the network lifetime, for example, when the initially available energy of BSs is high, when $|V_b|$ is large, when $|V_r|$ is small, when c_{lc} is small and when r is large. In these scenarios, MultiFixed outperforms OneMove, because MultiFixed reduces the energy consumption of RSNs by reducing the hop-distance from RSNs to BSs.

(iii) In all scenarios, the lifetime achieved by MultiMove-A is always very close to the optimum achieved by MultiMove and is always longer than those achieved by both OneMove and MultiFixed.

Lifetime versus the initially available energy of BSs: In Figure 3.5, we show the lifetime of the WSN $\tau N^*(\tau, \alpha)$ versus the initially available energy of BSs. We fix the initially available energy of RSNs $e_i = 3000, \forall i \in V_r$, and we vary the initially available energy of BSs (all BSs have the same amount though). When the initially available energy of BSs is small, OneMove is very close to the optimum achieved by MultiMove and MultiMove-A. Therefore, in this scenario, virtually moving multiple BSs degrades into the scheme of virtually moving one BS as in [69]. When the initially available energy of BSs increases, the increase of lifetime of OneMove diminishes and Multi-Fixed outperforms OneMove, because the bottleneck in the network lifetime is shifted from BSs to RSNs.

Lifetime versus the number of BSs: In Figure 3.6, we show the lifetime of the WSN $\tau N^*(\tau, \alpha)$ versus the number of BSs $|V_b|$. We see similar results as that in Figure 3.5. When $|V_b|$ is small, OneMove outperforms MultiFixed and performs close to the optimum as MultiMove-A does. This is because when BSs cause bottlenecks in the network lifetime, rotating one active BS is the optimal solution. When $|V_b|$ is large, RSNs cause bottlenecks in the lifetime of the WSN,

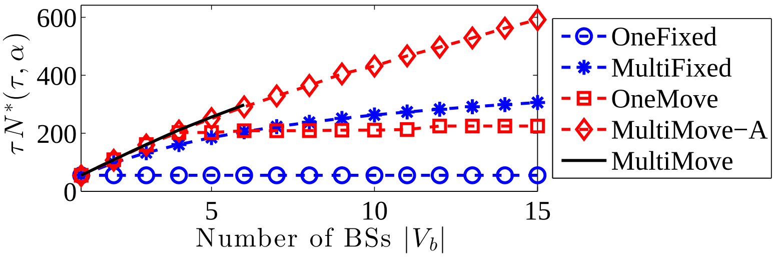


Figure 3.6: The network lifetime $\tau N^*(\tau,\alpha)$ versus the number of BSs $|V_b|$ when running different schemes..

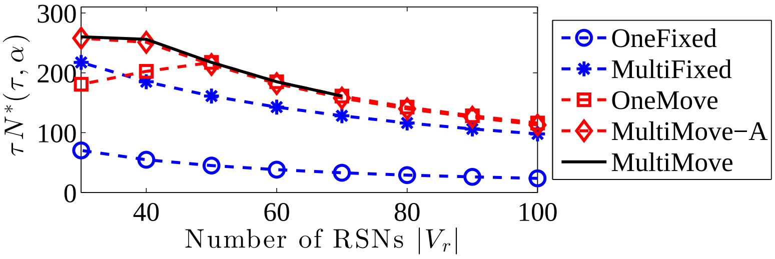


Figure 3.7: The network lifetime $\tau N^*(\tau,\alpha)$ versus the number of RSNs $|V_r|$ when running different schemes.

and MultiFixed outperforms OneMove.

Lifetime versus number of RSNs: In Figure 3.7, we show the lifetime of the WSN $\tau N^*(\tau,\alpha)$ versus the number of RSNs $|V_r|$. There is a clear trend that when $|V_r|$ increases, the lifetime of the network decreases. When $|V_r|$ is small, RSNs limit the lifetime of the WSN, and therefore OneMove performs poorly. When $|V_r|$ is large, BSs cause bottlenecks in the network lifetime, and OneMove performs close to the optimum achieved by MultiMove and MultiMove-A.

Lifetime versus the fixed energy-consumption of long-range communication: In Figure 3.8, we show the lifetime of the WSN $\tau N^*(\tau,\alpha)$ versus the fixed energy-consumption rate of long-range communication c_{lc}. There is a trend that when c_{lc} increases, the lifetime of the network decreases. Still, MultiMove-A and MultiMove only decrease mildly because they can evenly distribute the high energy-consumption load among the whole WSN. Meanwhile, we see that the lifetime curve of OneMove is flat when c_{lc} is small, because the bottleneck in lifetime is due to the RSNs rather than BSs in these scenarios.

Lifetime versus the data rate of RSNs: In Figure 3.9, we show the achieved lifetime $\tau N^*(\tau,\alpha)$ versus the data rate r using different schemes. For a better illustration, we set $e_i = 5000, \forall i \in V_b$ and $e_i = 1000, \forall i \in V_r$. We see results similar to that in Figure 3.8. When r is low, BSs cause bottlenecks in the lifetime, and OneMove outperforms MultiFixed. When r becomes high, RSNs cause the bottleneck in the network lifetime, and the lifetime achieved by MultiFixed is very close to the optimum.

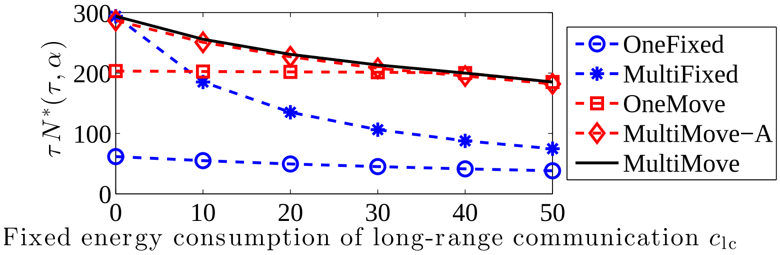


Figure 3.8: The network lifetime $\tau N^*(\tau, \alpha)$ versus the fixed energy-consumption of long-range communication c_{lc} when running different schemes.

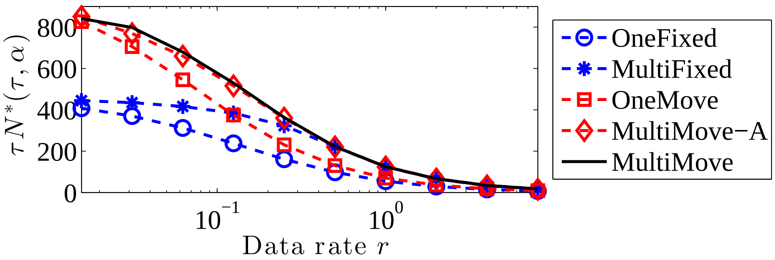


Figure 3.9: The network lifetime $\tau N^*(\tau, \alpha)$ versus the data rate of RSNs r when running different schemes.

3.7 Conclusion

In this chapter, we presented and evaluated the scheme of virtually moving multiple BSs in WSNs, where we adaptively re-elect an active subset of BSs. This scheme not only achieves a high energy-efficiency but also avoids the difficulty of physically moving the BSs. We showed that the general problem of virtually moving BSs is in fact NP-hard. We proposed an adaptive algorithm for scheduling the virtual movement by using the constrained gradient method and using previous results in uncapacitated facility-location problems. Under mild conditions, this algorithm guarantees to yield a lifetime at least 62% of the optimum. The idea of load balancing by virtually moving devices can be used in other networked systems, including super-node selection in peer-to-peer networks and cooperative beamforming in cellular networks. As a consequence, the proposed scheduling algorithm has merits beyond the WSNs.

3.A Appendix

3.A.1 Proof of Theorem 3.2

Consider problem (3.13) with its objective function $f(\cdot)$ relaxed as in (3.14). Let $\{(\boldsymbol{x}^{*[k]}$, $\boldsymbol{y}^{*[k]}$, $\boldsymbol{z}^{*[k]}$, $\boldsymbol{c}^{*[k]})\}_{k=1}^{2^{|V_b|}-1}$ be the optimal selection of configurations and let $\{p^{*[k]}\}_{k=1}^{2^{|V_b|}-1}$ be the optimal fractions of time for using these configurations. Let $\boldsymbol{\theta}^*$ be the optimal energy decrease rates

$$\boldsymbol{\theta}^* = \sum_{k=1}^{2^{|V_b|}-1} p^{*[k]} \boldsymbol{c}^{*[k]}. \tag{3.26}$$

Let $\{(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}, \boldsymbol{z}^{(n)}, \boldsymbol{c}^{(n)})\}_{n=1}^{N^*}$ be the selected sequence of configurations by using Algorithm 3.1 with a β-approximation algorithm for solving the min-weight configuration problem.

In the following, we will first derive an upper-bound of the objective value $f(\boldsymbol{\theta}^{(n)})$ for any $n \in \mathbb{N}^+$. Then, we will use that to derive a lower-bound of the lifetime.

The upper-bound of the objective value

First of all, we derive an upper-bound on the incremental change of the objective value $f(\boldsymbol{\theta}^{(n)}) - f(\boldsymbol{\theta}^{(n-1)})$. We take its Taylor expansion around $\boldsymbol{\theta}^{(n-1)}$ up to the second order,

$$\begin{aligned} f\left(\boldsymbol{\theta}^{(n)}\right) - f\left(\boldsymbol{\theta}^{(n-1)}\right) \le & \boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right) \cdot \left(\boldsymbol{\theta}^{(n)} - \boldsymbol{\theta}^{(n-1)}\right) \\ & + \frac{1}{2}|V_b \cup V_r|^2 \cdot \left\|\boldsymbol{\Delta} f\left(\boldsymbol{\theta}^{(n-1)}\right)\right\|_\infty \cdot \left\|\boldsymbol{\theta}^{(n)} - \boldsymbol{\theta}^{(n-1)}\right\|_\infty^2, \end{aligned} \tag{3.27}$$

where $\boldsymbol{\Delta} f(\boldsymbol{\theta}^{(n-1)})$ is the Hessian matrix of $f(\cdot)$ at the point $\boldsymbol{\theta}^{(n-1)}$.

The second term on the right-hand side of (3.27) can be easily upper-bounded. A little calculations give us an upper-bound of the infinite norm of $\boldsymbol{\Delta} f(\boldsymbol{\theta}^{(n-1)})$,

$$\left\|\boldsymbol{\Delta} f\left(\boldsymbol{\theta}^{(n-1)}\right)\right\|_\infty \le \frac{\alpha}{e_{\min}^2}. \tag{3.28}$$

From (3.16), we have

$$\boldsymbol{\theta}^{(n)} - \boldsymbol{\theta}^{(n-1)} = \frac{1}{n}\left(\boldsymbol{c}^{(n)} - \boldsymbol{\theta}^{(n-1)}\right). \tag{3.29}$$

And we have

$$\|\boldsymbol{c}^{(n)} - \boldsymbol{\theta}^{(n-1)}\|_\infty \le c_{\max} \tag{3.30}$$

because: (i) $0 \le \|\boldsymbol{c}^{(n)}\|_\infty < c_{\max}$ as $c_{\max}$ is the maximum energy-consumption rate of any BS or any RSN; (ii) $\|\boldsymbol{\theta}^{(n-1)}\|_\infty < c_{\max}$ since $\boldsymbol{\theta}^{(n-1)} = \sum_{t=1}^{n-1} \boldsymbol{c}^{(t)}/(n-1)$ and $0 \le \|\boldsymbol{c}^{(n)}\|_\infty < c_{\max}$. Combining (3.28), (3.29) and (3.30),

$$\frac{1}{2}|V_b \cup V_r|^2 \cdot \left\|\boldsymbol{\Delta} f\left(\boldsymbol{\theta}^{(n-1)}\right)\right\|_\infty \cdot \left\|\boldsymbol{\theta}^{(n)} - \boldsymbol{\theta}^{(n-1)}\right\|_\infty^2 \le \frac{\alpha|V_b \cup V_r|^2 c_{\max}^2}{2n^2 e_{\min}^2}. \tag{3.31}$$

It only remains to derive an upper-bound for the first term of the right-hand side of (3.27). Using (3.29),

$$\boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right) \cdot \left(\boldsymbol{\theta}^{(n)} - \boldsymbol{\theta}^{(n-1)}\right) = \frac{1}{n}\boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right) \cdot \boldsymbol{c}^{(n)} - \frac{1}{n}\boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right) \cdot \boldsymbol{\theta}^{(n-1)}, \tag{3.32}$$

where $\boldsymbol{\nabla}^\top f(\boldsymbol{\theta}^{(n-1)}) \cdot \boldsymbol{\theta}^{(n-1)}$ is independent of the selection of configuration at time n. It suffices to give an upper-bound for the term $\boldsymbol{\nabla}^\top f(\boldsymbol{\theta}^{(n-1)}) \cdot \boldsymbol{c}^{(n)}$. Remember that for any $1 \le n \le N^*$, we use a β-approximation algorithm for solving the min-weight configuration problem (3.18) with $\boldsymbol{\lambda} = \boldsymbol{\nabla}^\top f(\boldsymbol{\theta}^{(n-1)})$, we guarantee

$$\boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right) \cdot \boldsymbol{c}^{(n)} \le \beta f_s^{*(n)}, \tag{3.33}$$

where $f_s^{*(n)}$ denotes the optimal objective value of (3.18). Then, because for any $1 \leq k \leq 2^{|V_b|}-1$, $(\boldsymbol{x}^{*[k]}, \boldsymbol{y}^{*[k]}, \boldsymbol{z}^{*[k]}, \boldsymbol{c}^{*[k]}) \in \mathcal{L}^{[k]}$ is a feasible solution of the min-weight configuration problem (3.18),

$$f_s^{*(n)} \leq \boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right) \cdot \boldsymbol{c}^{*[k]}, \forall 1 \leq k \leq 2^{|V_b|}-1. \tag{3.34}$$

Combining (3.33) and (3.34), we have $\forall 1 \leq k \leq 2^{|V_b|}-1$,

$$\boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right) \cdot \boldsymbol{c}^{(n)} \leq \beta \boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right) \cdot \boldsymbol{c}^{*[k]}. \tag{3.35}$$

Multiplying (3.35) by $p^{*[k]}$ and summing it together for all $1 \leq k \leq 2^{|V_b|}-1$, we have

$$\boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right) \cdot \boldsymbol{c}^{(n)} \leq \beta \boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right) \cdot \boldsymbol{\theta}^{*}, \tag{3.36}$$

where we use $\sum_{k=1}^{2^{|V_b|}-1} p^{*[k]} = 1$ and (3.26).

Plugging (3.31), (3.32) and (3.36) into (3.27),

$$\begin{aligned} f\left(\boldsymbol{\theta}^{(n)}\right) - f\left(\boldsymbol{\theta}^{(n-1)}\right) &\leq \frac{1}{n} \boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right) \cdot \left(\beta \boldsymbol{\theta}^* - \boldsymbol{\theta}^{(n-1)}\right) + \frac{\alpha |V_b \cup V_r|^2 c_{\text{max}}^2}{2n^2 e_{\text{min}}^2} \\ &\leq \frac{1}{n}\left(\beta f(\boldsymbol{\theta}^*) - f\left(\boldsymbol{\theta}^{(n-1)}\right)\right) + \frac{\alpha |V_b \cup V_r|^2 c_{\text{max}}^2}{2n^2 e_{\text{min}}^2}, \end{aligned} \tag{3.37}$$

where the second inequality is because of the convexity of the objective function $f(\cdot)$.

Then, by multiplying n on both sides of (3.37) and by resorting,

$$n\left(f\left(\boldsymbol{\theta}^{(n)}\right) - \beta f(\boldsymbol{\theta}^*)\right) \leq (n-1)\left(f\left(\boldsymbol{\theta}^{(n-1)}\right) - \beta f(\boldsymbol{\theta}^*)\right) + \frac{\alpha |V_b \cup V_r|^2 c_{\text{max}}^2}{2n e_{\text{min}}^2}. \tag{3.38}$$

Summing it up from $n = 1$ to N ($N \in \mathbb{N}^+$), dividing it by N and using $\sum_{n=1}^{N} 1/n < \ln N + 1$, we have an upper-bound for $f(\boldsymbol{\theta}^{(N)})$,

$$f\left(\boldsymbol{\theta}^{(N)}\right) - \beta f(\boldsymbol{\theta}^*) < \frac{\alpha |V_b \cup V_r|^2 c_{\text{max}}^2 (\ln N + 1)}{2 e_{\text{min}}^2 N}. \tag{3.39}$$

The lower-bound of the lifetime

In the following, we will use (3.39) to derive a lower-bound of the lifetime τN^*. Let F^* denote the optimal objective value of problem (3.13) without approximations.

Using (3.15), one can easily check that $f(\boldsymbol{\theta}^{(N)}) \geq F(\boldsymbol{\theta}^{(N)})$ and that $f(\boldsymbol{\theta}^*) \leq F^* + \frac{1}{\alpha} \ln |V_b \cup V_r|$. Plugging them into (3.39), we see that

$$F\left(\boldsymbol{\theta}^{(N)}\right) - \beta F^* < \frac{\alpha |V_b \cup V_r|^2 c_{\text{max}}^2 (\ln N + 1)}{2 e_{\text{min}}^2 N} + \frac{\beta}{\alpha} \ln |V_b \cup V_r|. \tag{3.40}$$

Remember from (3.10) that $N = N^*$ is the maximum number of time slots that satisfies $\boldsymbol{e}^{(N)} \geq \boldsymbol{0}$. Therefore, $\boldsymbol{e}^{(N^*+1)} \geq \boldsymbol{0}$ does not hold true, which implies from (3.16) that $\boldsymbol{\theta}^{(N^*+1)} \leq \boldsymbol{e}/(\tau(N^*+1))$ does not hold true. Moreover, because $F(\boldsymbol{\theta}) = \max_{i \in V_b} \theta_i / e_i$, this is equivalent to

$$\tau(N^*+1) F\left(\boldsymbol{\theta}^{(N^*+1)}\right) > 1. \tag{3.41}$$

We will then use (3.40) and (3.41) to derive a lower-bound for τN^*.

Because $\boldsymbol{e}^{(N^*)} \geq \mathbf{0}$, because $0 \leq \|\boldsymbol{c}^{(n)}\|_\infty < c_{\max}$ for any $1 \leq n \leq N^*$ and because of (3.9), we have $N^* \geq e_{\min}/(\tau c_{\max})$, which in turn implies that $N^*+1 > e_{\min}/(\tau c_{\max})$. Then, because the term $(\ln N + 1)/N$ decreases,

$$\frac{\ln(N^*+1)+1}{N^*+1} < \frac{\tau c_{\max}}{e_{\min}} \left(\ln \frac{e_{\min}}{\tau c_{\max}} + 1 \right). \tag{3.42}$$

Setting $N = N^*+1$ in (3.40) and plugging in (3.42),

$$F\left(\boldsymbol{\theta}^{(N^*+1)}\right) < \beta F^* + \frac{\beta}{\alpha} \ln |V_b \cup V_r| + \frac{\alpha\tau |V_b \cup V_r|^2 c_{\max}^3}{2 e_{\min}^3} \left(\ln \frac{e_{\min}}{\tau c_{\max}} + 1 \right). \tag{3.43}$$

Using (3.41) and (3.43) together,

$$N^* + 1 > \frac{1}{\beta\tau F^* + \frac{\beta\tau \ln |V_b \cup V_r|}{\alpha} + \frac{\alpha\tau^2 |V_b \cup V_r|^2 c_{\max}^3}{2e_{\min}^3} \ln(\frac{e_{\min}}{\tau c_{\max}} + 1)}.$$

Multiplying τ/T^* on both sides,

$$\frac{\tau N^* + \tau}{T^*} > \frac{1}{\beta + \frac{\beta T^* \ln |V_b \cup V_r|}{\alpha} + \frac{\alpha\tau |V_b \cup V_r|^2 c_{\max}^3 T^* (\ln \frac{e_{\min}}{\tau c_{\max}} + 1)}{2e_{\min}^3}}, \tag{3.44}$$

where we use $T^* F^* = 1$.

Taking $\tau \to 0$ and then taking $\alpha \to \infty$ in (3.44), we have

$$\lim_{\alpha\to\infty} \lim_{\tau\to 0} \frac{\tau N^*(\tau,\alpha)}{T^*} \geq \frac{1}{\beta}.$$

3.A.2 Proof of Theorem 3.3

First, we will show that by using the mappings in Table 3.1, the constructed UFL has the same optimal objective value as that of the min-weight configuration problem (3.18).

We derive the equality from the side of the min-weight configuration problem (3.18). By plugging in (3.3) and (3.4), we transform the objective value of (3.18) into

$$\begin{aligned}
\boldsymbol{\lambda}^\top \boldsymbol{c} = & \sum_{i \in V_b \cup V_r} \lambda_i \left(c_{\mathrm{c}} + \sum_{(i,j)\in E_s} c_{\mathrm{st}} x_{ij} + \sum_{(j,i)\in E_s} c_{\mathrm{sr}} x_{ji} \right) \\
& + \sum_{i \in V_b} \lambda_i \left(c_{\mathrm{lt}} y_i + c_{\mathrm{lc}} z_i \right) \\
= & \sum_{i \in V_b \cup V_r} \lambda_i c_c + \sum_{(i,j)\in E_s} x_{ij} (\lambda_i c_{\mathrm{st}} + \lambda_j c_{\mathrm{sr}}) \\
& + \sum_{i \in V_b} \lambda_i c_{\mathrm{lt}} y_i + \sum_{i\in V_b} \lambda_i c_{\mathrm{lc}} z_i,
\end{aligned} \tag{3.45}$$

where the equality is due to the switch of terms.

We analyze the right hand side of the second equality of (3.45). The first term $\sum_{i \in V_r \cup V_b} \lambda_i c_{\mathrm{c}}$ is fixed and the last term $\sum_{i \in V_b} \lambda_i c_{\mathrm{lc}} z_i$ maps into the facility cost for selecting the active BSs

$$\sum_{i \in V_b} \lambda_i c_{\mathrm{lc}} z_i = \sum_{u \in \mathcal{F}_o} h_u \tag{3.46}$$

due to (3.21) and (3.24). Once the selection of active BSs $\boldsymbol{z}$ is fixed, problem (3.18) degrades into minimizing the second and third terms of (3.45) under constraints (3.1) and (3.2),

$$\begin{aligned} \min \quad & \sum_{(i,j) \in E_s} x_{ij} (\lambda_i c_{\mathrm{st}} + \lambda_j c_{\mathrm{sr}}) + \sum_{i \in V_b} \lambda_i c_{\mathrm{lt}} y_i \\ \text{s.t.} \quad & (3.1), (3.2). \end{aligned} \tag{3.47}$$

This problem boils down to routing a flow of rate r from every RSN $v \in V_r$ to the remote server S via the active BSs $\mathcal{F}_o = \{u : z_u = 1, u \in V_b\}$, such that the total length of routing paths is minimized given the distance assignments (3.22). Notice that L_{uv} denotes the length of the shortest path from $v \in V_r$ to S via active BS $u \in V_b$, and notice that active BSs are restricted by $\mathcal{F}_o = \{u : z_u = 1, u \in V_b\}$. Therefore, for any $v \in V_r$, its shortest path to S has a length $\min_{u \in \mathcal{F}_o} L_{uv}$. Hence, the optimal objective value of problem (3.47) is equal to $\sum_{v \in V_r} \min_{u \in \mathcal{F}_o} L_{uv}$. Using (3.23), (3.45) and (3.46), we see that the optimal objective value of (3.18) is equivalent to that of the constructed UFL problem.

Then, we will show that the service cost is metric. We denote the set of edges and the set of nodes along the shortest path from $v \in V_r$ to $u \in V_b$ by $P(u,v)$ and $N(u,v)$ (u, v are excluded from $N(u,v)$), respectively. We have

$$\begin{aligned} L_{uv} &= \sum_{(i,j) \in P(u,v)} (\lambda_i c_{\mathrm{st}} + \lambda_j c_{\mathrm{sr}}) r + \lambda_u c_{\mathrm{lt}} r \\ &= \sum_{i \in N(u,v)} (c_{\mathrm{st}} + c_{\mathrm{sr}}) \lambda_i r + (c_{\mathrm{lt}} + c_{\mathrm{sr}}) \lambda_u r + \lambda_v c_{\mathrm{st}} r, \end{aligned} \tag{3.48}$$

where we only change the costs on the path into the costs on nodes along that path.

Plugging (3.48) into the term $L_{uv} + L_{u'v} + L_{u'v'}$, we have

$$\begin{aligned}
&L_{uv} + L_{u'v} + L_{u'v'} \\
&\quad= \sum_{i \in N(u,v)} (c_{\text{st}} + c_{\text{sr}})\lambda_i r + (c_{\text{lt}} + c_{\text{sr}})\lambda_u r + \lambda_v c_{\text{st}} r \\
&\qquad+ \sum_{i \in N(u',v)} (c_{\text{st}} + c_{\text{sr}})\lambda_i r + (c_{\text{lt}} + c_{\text{sr}})\lambda_{u'} r + \lambda_v c_{\text{st}} r \\
&\qquad+ \sum_{i \in N(u',v')} (c_{\text{st}} + c_{\text{sr}})\lambda_i r + (c_{\text{lt}} + c_{\text{sr}})\lambda_{u'} r + \lambda_{v'} c_{\text{st}} r \\
&\quad= \sum_{\substack{i \in N(u,v) \\ \cup N(u',v) \\ \cup N(u',v')}} (c_{\text{st}} + c_{\text{sr}})\lambda_i r + 2\lambda_v c_{\text{st}} r + 2(c_{\text{lt}} + c_{\text{sr}})\lambda_{u'} r + (c_{\text{lt}} + c_{\text{sr}})\lambda_u r + \lambda_{v'} c_{\text{st}} r \\
&\quad\geq \sum_{\substack{i \in N(u,v) \\ \cup N(u',v) \\ \cup N(u',v')}} (c_{\text{st}} + c_{\text{sr}})\lambda_i r + (c_{\text{st}} + c_{\text{sr}})\lambda_v r + (c_{\text{st}} + c_{\text{sr}})\lambda_{u'} r + (c_{\text{lt}} + c_{\text{sr}})\lambda_u r + \lambda_{v'} c_{\text{st}} r \\
&\quad= \sum_{\substack{i \in N(u,v) \\ \cup \{v\} \cup N(u',v) \\ \cup \{u'\} \cup N(u',v')}} (c_{\text{st}} + c_{\text{sr}})\lambda_i r + (c_{\text{lt}} + c_{\text{sr}})\lambda_u r + \lambda_{v'} c_{\text{st}} r, \qquad (3.49)
\end{aligned}$$

where the inequality is because $c_{\text{sr}} \leq c_{\text{st}} \leq 2c_{\text{lt}} + c_{\text{sr}}$.

Moreover, from (3.48), we have

$$L_{uv'} = \sum_{i \in N(u,v')} (c_{\text{st}} + c_{\text{sr}})\lambda_i r + (c_{\text{lt}} + c_{\text{sr}})\lambda_u r + \lambda_{v'} c_{\text{st}} r.$$

Remembering that $L_{uv'}$ is the length of the shortest path from v' to u, it is not longer than the path through $v' \to N(u', v') \to u' \to N(u', v) \to v \to N(u, v) \to u$. Therefore $L_{uv} + L_{u'v} + L_{u'v'} \geq L_{uv'}$.

Using (3.23), we have

$$\begin{aligned}
&g_{uv} + g_{u'v} + g_{u'v'} - g_{uv'} \\
&\quad= L_{uv} + L_{u'v} + L_{u'v'} - L_{uv'} \\
&\qquad+ \lambda_v c_c + \lambda_v c_c + \lambda_{v'} c_c - \lambda_{v'} c_c \\
&\quad= (L_{uv} + L_{u'v} + L_{u'v'} - L_{uv'}) + 2\lambda_v c_{\text{c}} \\
&\quad\geq 0.
\end{aligned}$$

Chapter 4

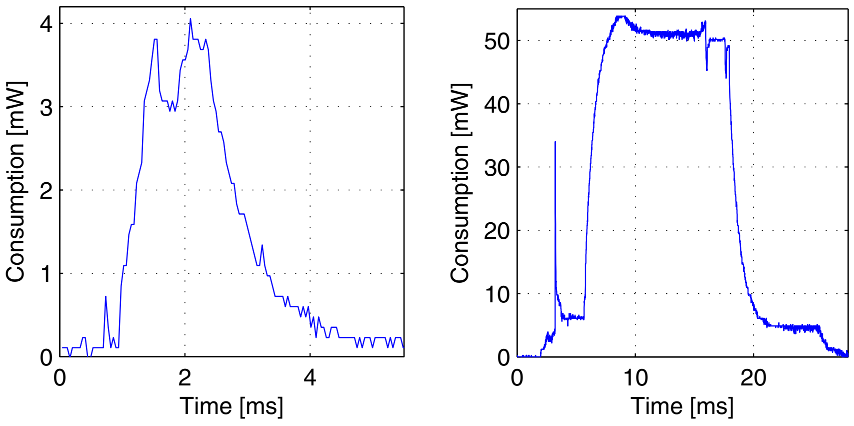
Sparse Sensor-Selection by Exploiting Temporal Correlations

> The scientist is not a person who gives the right answers, he's one who asks the right questions.
>
> *Claude Lévi-Strauss*

In the last two chapters, we studied the problem of adaptively selecting BSs in order to enhance the energy efficiency in WSNs. In this chapter, we will study a similar problem, that is, adaptively selecting the sparse sensing-samples for enhancing the energy efficiency in both WSNs and participatory sensing systems.

4.1 Introduction

Traditionally, in either a WSN or a participatory sensing system, the communication costs are assumed to dominate the overall energy-consumption of a node, whereas the sensing costs are considered negligible. This leads the research community of WSNs to optimize the communication costs alone. However, the gap between the energy costs for communication and that for sensing might not be as large as we thought. We measure these two parts of energy consumption on *Tmote-sky*, a low-power sensor node widely used in WSNs [64]; we program this sensor node to sample the ambient light intensity and transmit the samples through short-range radios. The measurement results are shown in Figure 4.1. The ratio between the energy consumption for taking one sample of the light intensity and that for transmitting one sample is 0.26. This ratio could be even larger in the following two scenarios:



(a) (b)

Figure 4.1: Energy consumptions of a *Tmote-sky* sensor: (a) while the node measures one sample of light intensity (two-bytes), $E_{sensor} = 7.5 \times 10^{-6}$J; (b) while the node transmits a packet with 24 samples of payload, $24 \cdot E_{radio} = 6.9 \times 10^{-4}$J. The ratio between E_{sensor} and E_{radio} is 0.26.

— Sensors with high energy-consumption: for example, an air pollution sensor consumes $30 \sim 50$ mW instead of the 3 mW of a *Tmote-sky*'s light sensor.

— Sensors with long sampling times: for example, the anemometer, a sensor that measures the direction and strength of the wind, requires $1 \sim 3$ seconds of continuous measurement per sample instead of the 4 ms of the *Tmote-sky*'s light sensor.

Therefore, optimizing the energy consumption for communication alone is not optimal. It is also desirable to reduce the energy consumption for sensing, or in other words, to reduce the number of sensing samples.

Sensing samples are usually temporally correlated. Therefore, researchers propose to use sparse patterns for sampling, so that the energy consumption for sensing is reduced while a certain sensing precision is maintained. For example, they use fixed or randomly generated patterns for scheduling the sparse sampling [48, 66]. However, these works are not well adapted to signal variations.

We propose a novel sparse-sensing scheme, that we call *"distributed adaptive sparse sensing"* (DASS). In this scheme, we learn the signal statistics on-the-fly from the captured measurements. Then, using the signal statistics, we find the optimized and generally irregular sparse-sampling pattern, which captures the most information of the original signal.

The key novelty of DASS is to find the optimal sampling pattern, that is, to decide when to measure for collecting as much information as possible. It deviates from the recent sparse-sensing schemes [48] [66] that have fixed or random sampling-patterns, in the sense that DASS dynamically adjusts the sampling pattern through online learning of signal statistics.

It is worth mentioning that DASS does not impose on-sensor computation or inter-node communication. Each sensor node simply takes measurements according to a designated sampling-schedule and transmits the data to a common server. The server collects all the data from one or multiple sensor nodes and performs signal reconstruction.

4.2 Related Works

Consider a continuous-time signal. Denote by $\boldsymbol{x} \in \mathbb{R}^N$ the discretization of such a signal with a sufficiently large number of discretized data points N. The target of the WSN is to sample $\boldsymbol{x}$ and then recover it with the maximum precision.

A traditional WSN collects data as frequent as possible, which is subsequently compressed and transmitted with the lowest possible rate. In other words, it collects a vector of samples $\boldsymbol{y}_0$ that is equal to the discretized physical signal $\boldsymbol{x}$ with some additive noise,

$$\boldsymbol{y}_0 = \boldsymbol{I}\boldsymbol{x} + \boldsymbol{\omega}, \tag{4.1}$$

where $\boldsymbol{I}$ is the identity matrix of size N and $\boldsymbol{\omega}$ represents the noise; see Figure 4.2a for an example.

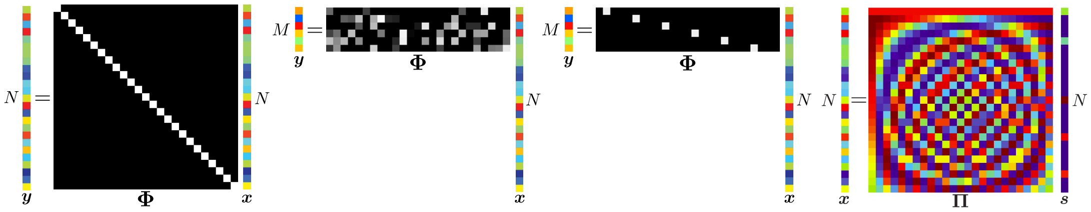
Ignoring the energy cost for sensing is sub-optimal, if sensing consumes a comparable amount of energy to communication and data processing. In fact, new sampling paradigms optimizing the overall energy-consumption have been proposed and show that further reductions of the energy consumption are possible. The basic idea involves a reduction of the number of collected samples and a reconstruction of the missing data by using algorithms that exploit the temporal structure available in the measured data. The reduction of the collected samples is done by designing a sampling operator $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$ $(M \ll N)$ instead of the identity matrix,

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{\omega}. \tag{4.2}$$

Note that $\boldsymbol{y}$ is significantly shorter than $\boldsymbol{x}$ and the reconstruction algorithm must estimate a significant amount of information from a limited amount of data. Therefore, regularization and constraints are added to the problem so that a stable solution can be obtained. Moreover, the reconstruction algorithm must be jointly designed with the sampling matrix $\boldsymbol{\Phi}$ for precise estimation of $\boldsymbol{x}$.

Pioneering work on sparse sampling used compressive sensing (CS) as a reconstruction scheme. CS attempts to recover $\boldsymbol{x}$ by solving a convex optimization problem, under the assumption that $\boldsymbol{x}$ is sparse in a known dictionary $\boldsymbol{\Pi}$. However, the solution is only approximate and it is exact if $\boldsymbol{\Pi}$ and $\boldsymbol{\Phi}$ satisfy certain requirements that are generally hard to check [12]. Initially, [21, 43, 62] proposed the use of a sampling matrix $\boldsymbol{\Phi}$ composed of random i.i.d. Gaussian entries. Note from Figure 4.2b that such $\boldsymbol{\Phi}$ has very few zero elements. Therefore, the number of sensing operations is not actually reduced, because we need to know all the values of $\boldsymbol{x}$ to compute $\boldsymbol{y}$. Moreover, if we adopt a distributed algorithm, a dense $\boldsymbol{\Phi}$ requires the sensor nodes to transmit their local samples to the other nodes, which would cause an excessive energy-consumption for communications.

To overcome such limitations, researchers proposed to use a sparse matrix $\boldsymbol{\Phi}$ which contains very few non-zero elements [48, 66] . More precisely, $\boldsymbol{\Phi}$ has only one non-zero element per row and the locations of such elements determine the temporal sampling pattern, see Figure 4.2c. However, the sampling patterns in these schemes are either fixed or randomly generated and thus not well adapted to the measured signal. Moreover, it is hard to guarantee the recovery of a faithful representation of $\boldsymbol{x}$, because the sparsity of dictionary $\boldsymbol{\Pi}$ usually changes over time and it might not satisfy the theoretical requirements of CS [10].



(a) Traditional Sensing (b) CS - Dense Matrix (c) CS - Sparse Matrix (d) Sparsity Dictionary

Figure 4.2: Comparison of various sensing schemes proposed in the literature (the noise term $\boldsymbol{\omega}$ is omitted for simplicity). We consider a discretized version of the sampled physical signal that is contained into a vector $\boldsymbol{x}$. In (a) we depict the traditional approach where we measure the signal in every time spot, thus the sampling operator is an identity matrix $\boldsymbol{I}$. In (b), we reduce the number of samples by taking random projections of the measurements. Note that we need to measure all the elements of $\boldsymbol{x}$ and we just reduce the number of stored samples. Whereas, in (c) we reduce the number of measured samples by using a sparse-sampling matrix $\boldsymbol{\Phi}$. Note that the methods in (b) and (c) require a set of conditions regarding $\boldsymbol{x}$ and $\boldsymbol{\Phi}$ to be satisfied [10]. Among these conditions, we note that, $\boldsymbol{x}$ must be sparse under a certain known dictionary $\boldsymbol{\Pi}$, see (d).

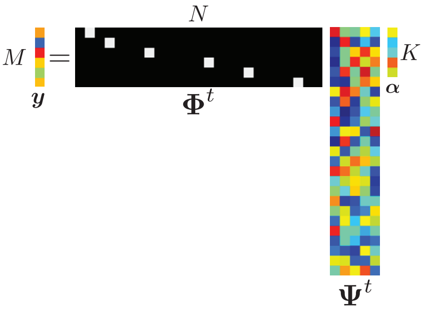


Figure 4.3: Graphical representation of the mathematical model of the proposed sensing scheme. The signal is modeled by an unknown time-varying linear K-dimensional model $\boldsymbol{\Psi}^t$ that is learned from the collected measurements. The sampling pattern $\boldsymbol{\Phi}^t$ is optimized at run-time according to the signal model and measures only M values out of the N available ones.

As the statistics of $\boldsymbol{x}$ are often unknown and varying over time, it may be advantageous to consider the decomposition

$$\boldsymbol{x} = \boldsymbol{\Psi}^t \boldsymbol{\alpha}, \tag{4.3}$$

where $\boldsymbol{\Psi}^t$ is the time-varying model and $\boldsymbol{\alpha} \in \mathbb{R}^K$ is a low-dimensional representation of $\boldsymbol{x}$ with $K \ll N$. The temporal model $\boldsymbol{\Psi}^t$ is non-stationary and has to be adaptively learned from the samples collected in the past. Using this model $\boldsymbol{\Psi}^t$, we adaptively change the sampling pattern $\boldsymbol{\Phi}^t$. The non-stationarity of $\boldsymbol{\Phi}^t$ is a feature diversifying our approach from the CS algorithms, where the sensing patterns are typically fixed as shown in Figure 4.2.

This new problem statement raises new challenges. While the model $\boldsymbol{\Psi}^t$ can be learned from the incomplete measurements $\boldsymbol{y}$ by using an online version of the principal component

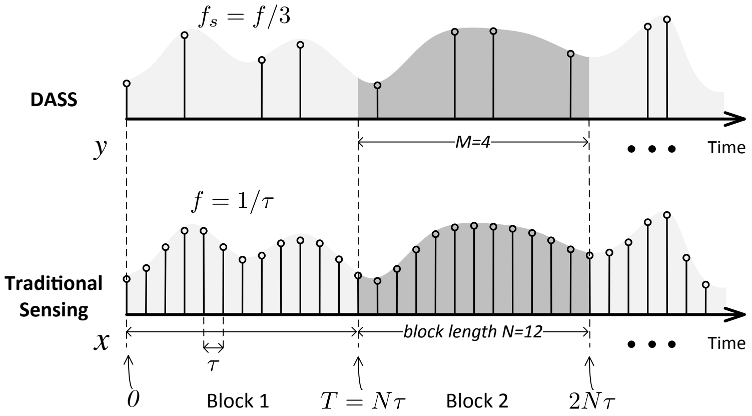


Figure 4.4: Upper plot: optimized temporal sampling pattern of DASS. Lower plot: traditional sensing scheme, where samples are collected regularly in time. The subsampling factor is $\gamma = 1/3$, since we collect 4 samples instead of 12 in each block.

analysis (PCA), selecting the sampling pattern $\mathbf{\Phi}^t$ for minimizing the reconstruction error is a combinatorial problem. In this chapter, we propose a generalized version of FrameSense, an algorithm that generates a near-optimal sensor placement for inverse problems [51]. Specifically, instead of optimizing the sensor placement, we optimize the temporal sampling pattern of the WSN. The obtained sampling pattern is generally irregular, time-varying and optimized to gather the maximum amount of information. See Figure 4.3 for a graphical illustration of the low-dimensional model and of the irregular sampling patterns.

Our method derives from and extends the sparse-sensing framework proposed by Quer *et al.* [48]: the signal is first approximated by a linear model $\mathbf{\Psi}^t$, and the sampling scheduling is defined in space and time by a sampling matrix $\mathbf{\Phi}^t$. Our major contribution is that we improve the way in which the temporal correlation is exploited, such that the sampling pattern is dynamically adapted to the low-dimensional model of the signal.

4.3 Problem Formulation

In this section, we first state the sampling-schedule problem for a single sensor. At the end of this section, we consider the general sampling-schedule problem for multiple sensors. We consider a block-based sensing strategy, meaning that the sensor takes sensing samples for a certain time T and at the end we reconstruct the signal in this block from the collected samples. Note that the block length is known and defined a-priori.

For each temporal block, let the discrete-time signal $\boldsymbol{x}$ be composed of N samples, and let τ be the temporal resolution (its inverse is the sampling frequency, $f = 1/\tau$). The temporal duration of a block is $T = N\tau$. See Figure 4.4 for a graphical representation of the signal to be sensed and its time-discretized version $\boldsymbol{x}$.

The objective is to select just a subset of samples of $\boldsymbol{x}$ and then reconstruct it using the captured samples. We denote the reconstructed signal by $\widetilde{\boldsymbol{x}}$. Let the number of sensing samples to be taken in each time-block be M ($M < N$). In each time block $t \in \mathbb{N}$, we denote by

$\boldsymbol{\chi}^t = \{\chi_i^t\}_{i=1}^M$ the set of indices of these M samples and it is adaptively chosen according to the previous measurements. The sampling pattern $\boldsymbol{\chi}^t$ determines the sampling matrix $\boldsymbol{\Phi}^t \in \mathbb{R}^{M\times N}$ in the t-th time block, where the (i,j)-th element of $\boldsymbol{\chi}^t$ is

$$\boldsymbol{\Phi}_{i,j}^t = \begin{cases} 1 & \text{if } j = \chi_i^t \\ 0 & \text{otherwise} \end{cases} .$$

The sensing matrix $\boldsymbol{\Phi}^t$ has exactly one non-zero element per row, and usually a maximum of one non-zero element per column. Here either $\boldsymbol{\Phi}_{i,j}^t$ or χ_i^t can be interpreted as a temporal selector that decides when the sensor i should take a sample – the index j indicates the time index within a block. It is important to underline that $\boldsymbol{\Phi}^t$ and $\boldsymbol{\chi}^t$ are time-varying and potentially changing at every block to adapt to the signal model $\boldsymbol{\Psi}^t$. Figure 4.4 shows an example of sampling patterns where $\boldsymbol{\chi}^t$ changes for each block.

We define $f_s = \frac{M}{N} \cdot f = \gamma f$ to be the average sampling frequency of the sensor node[4]. The subsampling rate $\gamma = f_s/f < 1$ is an important figure of merit for a sparse-sampling algorithm—the lower the γ, the lower the energy consumed for sensing.

The measured signal $\boldsymbol{y} \in \mathbb{R}^M$ is defined as

$$\boldsymbol{y} = \boldsymbol{\Phi}^t \boldsymbol{x} + \boldsymbol{\omega}, \tag{4.4}$$

where $\boldsymbol{\omega}$ represents the measurement noise that is modelled as an additive white Gaussian noise (AWGN), because the thermal effects [37] or/and quantization [65] are often the dominating terms[5]. Throughout this chapter, we mainly discuss the simple case of i.i.d. noises; later, we will discuss the extension to the generic case where the noises have a correlation matrix $\Sigma_{\boldsymbol{\omega}}$. We define the signal-to-noise ratio (SNR) of the measurement as following, which will be used in the evaluations:

$$\text{SNR (dB)} = 10 \log_{10} \left(\frac{\|\boldsymbol{x}\|_2^2}{\|\boldsymbol{\omega}\|_2^2} \right) . \tag{4.5}$$

The target of DASS is to optimize the sampling pattern $\boldsymbol{\Phi}^t$ at the t-th block according to $\boldsymbol{\Psi}^t$ such that we collect a small number of samples M and are still able to recover precisely the original signal. As we model the noise as a AWGN, we assess the quality of the recovered signal by using root-mean-square error (RMSE):

$$\epsilon = \frac{1}{\sqrt{N}} \|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\|_2 .$$

Multiple-node scenario: Although the above problem statement focuses on a single-sensor scenario for simplicity of notation, it is simple to generalize the statement to a WSN with more than one sensor node. We assume that the nodes are synchronized[6], so that we can concatenate all the measured blocks at different locations in a unique signal block $\boldsymbol{x}$. Figure 4.5 shows

---

[4] Note that we denote f_s as an average sampling frequency given the irregular and time-varying sampling pattern.

[5] Other noise models may be of interest for specific sensors; for example the noise term of a Geiger counter is usually modeled as a Poisson process.

[6] Note that the proposed method does not require a precise synchronization. In fact, variations of the model due to the lack of synchronization are handled by the proposed method thanks to the adaptive learning of the model.

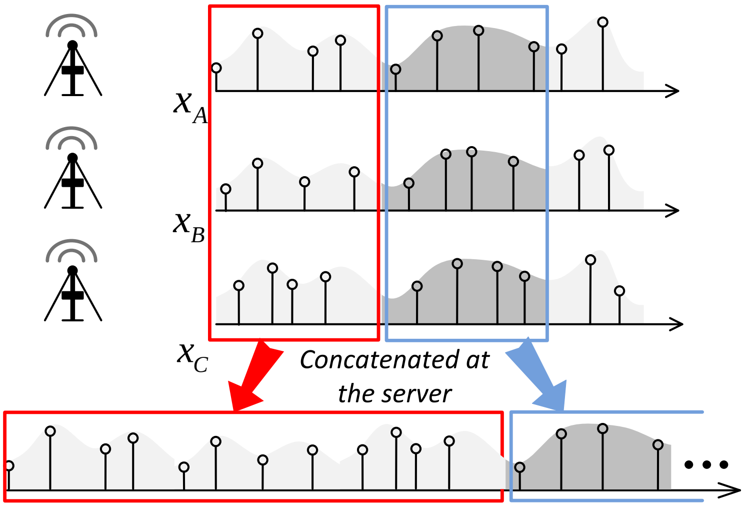


Figure 4.5: Signals of multiple distributed sensor nodes can be concatenated into a single signal stream at the server for recovery.

an example. $\boldsymbol{x}_A, \boldsymbol{x}_B, \boldsymbol{x}_C$ are signal blocks from three different locations; $\boldsymbol{\Phi}_A^t, \boldsymbol{\Phi}_B^t, \boldsymbol{\Phi}_C^t$ are the respective sampling matrices for each location, and $\boldsymbol{y}_A, \boldsymbol{y}_B, \boldsymbol{y}_C$ are the respective measurements. We can write

$$\begin{bmatrix} \boldsymbol{y}_A \\ \boldsymbol{y}_B \\ \boldsymbol{y}_C \end{bmatrix} = \boldsymbol{\Phi}^t \begin{bmatrix} \boldsymbol{x}_A \\ \boldsymbol{x}_B \\ \boldsymbol{x}_C \end{bmatrix} + \boldsymbol{\omega}, \text{ where } \boldsymbol{\Phi}^t = \begin{bmatrix} \boldsymbol{\Phi}_A^t & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi}_B^t & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Phi}_C^t \end{bmatrix}. \tag{4.6}$$

Here different sensors can take different number of samples, and $\boldsymbol{\Phi}_A^t, \boldsymbol{\Phi}_B^t, \boldsymbol{\Phi}_C^t$ can have different sizes. Thus, $\boldsymbol{\Phi}^t$ can be interpreted as a general spatio-temporal selector to choose *when and where* to sample such that we collect the maximum amount of information. Moreover, it is worth mentioning that $\boldsymbol{\Phi}^t$ is optimized for each block to adapt to the time-varying model of the signal.

4.4 Building Blocks

The proposed method is graphically represented in Figure 4.6 and is based on three building blocks:

- The desired signal $\widetilde{\boldsymbol{x}}$ is reconstructed using the collected measurements $\boldsymbol{y}$, the signal model $\boldsymbol{\Psi}^t$ and the estimated mean $\overline{\boldsymbol{x}}$ (Section 4.4.1).
- We use the measurements $\boldsymbol{y}$ to update the approximation model $\boldsymbol{\Psi}^t, \overline{\boldsymbol{x}}$ (Section 4.4.2).
- The sampling pattern for the subsequent temporal block $\boldsymbol{\chi}^{t+1}$ is optimized according to $\boldsymbol{\Psi}^t$ and is transmitted back to the sensor node(s) (Section 4.4.3).

The overhead of DASS on the sensor node is minimal in practice. First, the sampling pattern $\boldsymbol{\chi}^t$ has a sparse structure, hence it can be encoded efficiently with a few bytes per block. Therefore, the extra communication cost for receiving $\boldsymbol{\chi}^t$ is minimal. Second, all the algorithmic

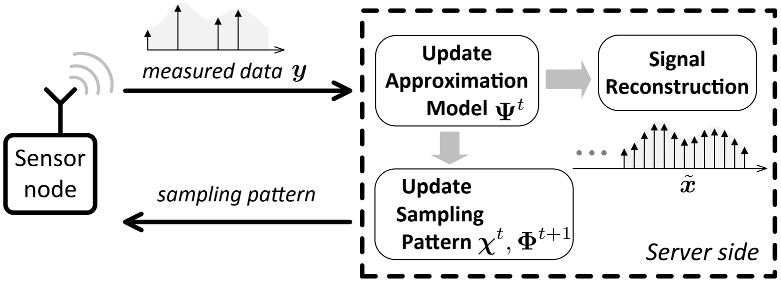


Figure 4.6: Representation of the operations of DASS in a WSN. The sensor node sends the measured data to the processing server and receives the sampling pattern for the next temporal block. The server uses the data to update the signal model $\boldsymbol{\Psi}^t$, reconstructs the discretized signal $\widetilde{\boldsymbol{x}}$ and optimizes the sampling pattern $\boldsymbol{\chi}^{t+1}$ for the sensor nodes. Note that $\boldsymbol{\chi}^{t+1}$ determines $\boldsymbol{\Phi}^{t+1}$.

complexity of DASS is at the server side, and the sensor nodes only need to sample and transmit the signal according to the sampling pattern received from the server. Therefore, the CPU and memory requirements of the sensor node are minimal.

In the following, we analyze each block by explaining the challenges and the proposed solutions.

4.4.1 Signal Approximation and Reconstruction

Due to the nature of most physical signals, a signal block is partially predictable by analyzing past data. In many cases, this predictability can be expressed by assuming that the signal belongs to a K-dimensional linear subspace $\boldsymbol{\Psi}^t \in \mathbb{R}^{N\times K}$. Such a subspace approximates $\boldsymbol{x}$ as

$$\widehat{\boldsymbol{x}} = \boldsymbol{\Psi}^t\boldsymbol{\alpha} + \overline{\boldsymbol{x}}, \tag{4.7}$$

where $\widehat{\boldsymbol{x}}$ is the approximated field, $\boldsymbol{\alpha} \in \mathbb{R}^K$ is the vector of the projection coefficients and $\overline{\boldsymbol{x}}$ is the mean of $\boldsymbol{x}$.

If the modeling subspace $\boldsymbol{\Psi}^t$ is well designed and K is sufficiently large compared to the complexity of $\boldsymbol{x}$, the signal realization $\boldsymbol{x}$ can be accurately expressed with just $K << N$ coefficients contained in $\boldsymbol{\alpha}$. To find such a subspace, we analyze all the past signal realizations and estimate at the t-th block the K-dimensional subspace $\boldsymbol{\Psi}^t$ that minimizes the expected approximation error

$$\epsilon_a = \frac{1}{\sqrt{N}}\mathbb{E}\left(\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|_2\right). \tag{4.8}$$

This is a dimensionality-reduction problem that can be solved by the well-known technique of *principal component analysis (PCA)*[7]. It has an analytic solution but requires the covariance matrix $\boldsymbol{C}_{\boldsymbol{x}}$.

Unfortunately, in our scenario it is hard to estimate $\boldsymbol{C}_{\boldsymbol{x}}$ because we have access only to M out of N elements of $\boldsymbol{x}$. However, if the M sampled elements vary at each temporal block t, we

---

[7] The CS-based sparse-sensing methods in [48] also used PCA for subspace learning.

can collect enough information to have a sufficiently precise estimate of $\boldsymbol{C}_{\boldsymbol{x}}$. We present a set of methods to estimate $\boldsymbol{C}_{\boldsymbol{x}}$ in Section 4.4.2.

Note that the approximation through $\boldsymbol{\Psi}^t$ exploits the temporal correlation among the elements of $\boldsymbol{x}$. The higher the correlation available in $\boldsymbol{x}$, the lower the dimensionality of the subspace $\boldsymbol{\Psi}^t$, the number of parameters K and the necessary measurements M. Hence, one of the key aspects is the choice of the signal block length T. In fact, it should be chosen such that the delay of the WSN respects the design specification while maximizing the correlation among the blocks. For example, a sensor measuring the outdoor light intensity naturally shows diurnal patterns. Therefore, if we choose a block length of one hour, the correlation within the signal block is usually weak. Whereas, if we choose a block length of one day, the correlation is stronger.

Once the approximation model $\boldsymbol{\Psi}^t$ is estimated, recovering the signal $\widetilde{\boldsymbol{x}}$ amounts to estimating $\boldsymbol{\alpha}$ from the measurements $\boldsymbol{y}$ when considering the approximated signal model,

$$\boldsymbol{y} \approx \boldsymbol{\Phi}^t \widehat{\boldsymbol{x}} + \boldsymbol{\omega} = \boldsymbol{\Phi}^t(\boldsymbol{\Psi}^t \boldsymbol{\alpha} + \overline{\boldsymbol{x}}) + \boldsymbol{\omega}. \tag{4.9}$$

If $\boldsymbol{\omega}$ is an i.i.d. Gaussian random noise, we can recover $\boldsymbol{\alpha}$ by solving an ordinary least square (OLS) problem [48]

$$\widetilde{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - \boldsymbol{\Phi}^t \overline{\boldsymbol{x}} - \boldsymbol{\Phi}^t \boldsymbol{\Psi}^t \boldsymbol{\alpha}\|_2^2, \tag{4.10}$$

which has the following analytic solution,

$$\widetilde{\boldsymbol{\alpha}} = (\boldsymbol{\Phi}^t \boldsymbol{\Psi}^t)^\dagger (\boldsymbol{y} - \boldsymbol{\Phi}^t \overline{\boldsymbol{x}}). \tag{4.11}$$

Here $(\boldsymbol{\Phi}^t \boldsymbol{\Psi}^t)^\dagger$ is the Moore-Penrose pseudoinverse of $\boldsymbol{\Phi}^t \boldsymbol{\Psi}^t$ that is defined for a generic matrix $\boldsymbol{A}$ as $\boldsymbol{A}^\dagger = (\boldsymbol{A}^* \boldsymbol{A})^{-1} \boldsymbol{A}^*$. We can slightly vary the reconstruction technique to account for more complicated noise models. For example, if the noise is distributed according a known covariance matrix $\Sigma_{\boldsymbol{\omega}}$, we can estimate $\boldsymbol{\alpha}$ as

$$\widetilde{\boldsymbol{\alpha}} = ((\boldsymbol{\Phi}^t \boldsymbol{\Psi}^t)^* \Sigma_{\boldsymbol{\omega}}^{-1} \boldsymbol{\Phi}^t \boldsymbol{\Psi}^t)^{-1} (\boldsymbol{\Phi}^t \boldsymbol{\Psi}^t)^* \Sigma_{\boldsymbol{\omega}}^{-1} (\boldsymbol{y} - \boldsymbol{\Phi}^t \overline{\boldsymbol{x}}). \tag{4.12}$$

We can generalize such an estimator to other additive noise models, see [39]. For the remainder of the chapter we keep considering the i.i.d. Gaussian noise case. Nonetheless, we show in Appendix 4.A that if the noises are correlated, we could still use the proposed scheme by whitening the measured data before the processing.

Once we have the estimator $\widetilde{\boldsymbol{\alpha}}$, the reconstruction algorithm is straightforward and is described in Algorithm 4.1. The following proposition states the necessary conditions to find a unique solution and provides an upper bound for the reconstruction error, that is going to be fundamental when optimizing the sampling pattern. Such a result is an adaption of two classic results of linear algebra [28].

```
Algorithm 4.1: Signal reconstruction
  Input: $\mathbf{\Psi}^t$, $\overline{\mathbf{x}}$, $\mathbf{\chi}^t$ and $\mathbf{\Phi}^t$
  Output: $\widetilde{\mathbf{x}}$
  1 Measure the signal $\mathbf{y}$ according to $\mathbf{\chi}^t$. $\widetilde{\mathbf{x}} = \mathbf{\Psi}^t(\mathbf{\Phi}^t\mathbf{\Psi}^t)^\dagger(\mathbf{y} - \mathbf{\Phi}^t\overline{\mathbf{x}}) + \overline{\mathbf{x}}$.
```

Proposition 4.1

Consider a sensor measuring a temporal signal as in (4.9) where the M measurements are corrupted by an i.i.d. Gaussian noise with variance σ^2. If $M \geq K$, $\mathbf{\Psi}^t \in \mathbb{R}^{N\times K}$ is formed by orthonormal columns and $\operatorname{rank}(\mathbf{\Phi}^t\mathbf{\Psi}^t) = K$*, then $\widetilde{\mathbf{x}}$ can be determined using Algorithm 4.1. The reconstruction error is bounded by*

$$\epsilon^2 = \frac{1}{N}\|\mathbf{x} - \widetilde{\mathbf{x}}\|_2^2 \leq \frac{1}{\lambda_K}\epsilon_a^2 + \sigma^2\sum_{i=1}^{K}\frac{1}{\lambda_i}, \tag{4.13}$$

where ϵ_a is the approximation error due to the signal model $\mathbf{\Psi}^t$ and λ_i $(1 \leq i \leq K)$ is the i-th largest eigenvalue of the matrix $\mathbf{\Psi}^{t*}\mathbf{\Phi}^{t*}\mathbf{\Phi}^t\mathbf{\Psi}^t$.

Proof.

Since the Gaussian noise is independent from the approximation error, we can treat it independently. Moreover, it is sufficient to compute the error on the estimation of $\boldsymbol{\alpha}$, given the assumed orthonormality of the columns of $\mathbf{\Psi}^t$.

For reconstruction error due to the approximation error ϵ_a, we look at the worst-case scenario with the following optimization problem,

$$\begin{aligned} \max \quad & \|(\mathbf{\Psi}^t\mathbf{\Psi}^t)^\dagger(\mathbf{x} - \widehat{\mathbf{x}})\|_2^2 \\ \text{subject to} \quad & \frac{1}{N}\|(\mathbf{x} - \widehat{\mathbf{x}})\|_2^2 = \epsilon_a, \end{aligned}$$

whose solution is proportional to the smallest eigenvalue of $(\mathbf{\Psi}^t\mathbf{\Psi}^t)^\dagger$. More precisely, it is possible to show that the approximation noise is upper bounded by $\frac{1}{\lambda_K}\epsilon_a^2$, where ϵ_a is the norm of the approximation error.

For the reconstruction error due to the white noise, we use a known result of frame theory, see [23]. We merge the two results to conclude the proof. ☐

The upper-bound of the total error ϵ is a function of both the approximation error ϵ_a and the measurement noise $\boldsymbol{\omega}$. The former term depends on the number of parameters K: when $K = N$, we have $\epsilon_a = 0$ and it grows when we decrease K. However, the rate at which the error increases depends on the spectrum of $C_{\mathbf{x}}$. In fact, if $\mathbf{x}$ has elements that are highly correlated, a small K could be sufficient to model $\mathbf{x}$ with a small approximation error. The latter term can be controlled directly by optimizing the sampling pattern.

Note that the part involving ϵ_a depends only on the smallest eigenvalue, because we are not guaranteed that the approximation error *spreads* over all the eigenvectors of $\mathbf{\Phi}^t\mathbf{\Psi}^t$. In fact, the worst-case scenario is represented by the approximation error that is in the same direction of the eigenvector with the smallest eigenvalue and ϵ_a is consequently maximally amplified.

Algorithm 4.2: Updating $\boldsymbol{\Psi}^t, \overline{\boldsymbol{x}}$ using a buffer

```
Input: $\boldsymbol{y}$, $L$
Output: $\boldsymbol{\Psi}^t, \overline{\boldsymbol{x}}$
Interpolate $\boldsymbol{y} \rightarrow \boldsymbol{x}_{\text{intep}}$. Insert $\boldsymbol{x}_{\text{intep}}$ into a buffer storing the most recent $L$ blocks.
Estimate $\mathbf{C}_{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}$ from the buffer. $\boldsymbol{\Psi}^t$ is formed by the eigenvectors corresponding to
the $K$ largest eigenvalues of the matrix $\mathbf{C}_{\boldsymbol{x}}$.
```

Algorithm 4.3: Updating $\boldsymbol{\Psi}^t, \overline{\boldsymbol{x}}$ using incremental PCA

```
Input: $\boldsymbol{y}$, $L$, $\boldsymbol{\Psi}^{t-1}$, $\boldsymbol{\lambda}^{t-1}, \overline{\boldsymbol{x}}^{t-1}$
Output: $\boldsymbol{\Psi}^t, \boldsymbol{\lambda}^t, \overline{\boldsymbol{x}}^t$
Interpolate $\boldsymbol{y} \rightarrow \boldsymbol{x}_{\text{intep}}$. $\mathbf{a} = \boldsymbol{\Psi}^{t-1*}(\boldsymbol{x}_{\text{intep}} - \overline{\boldsymbol{x}}^{t-1})$. $\mathbf{b} = \left(\boldsymbol{\Psi}^{t-1}\mathbf{a} + \overline{\boldsymbol{x}}^{t-1}\right)\big) - \boldsymbol{x}_{\text{intep}}$, and then
normalize $\mathbf{b}$. $c = \mathbf{b}^*(\boldsymbol{x}_{\text{intep}} - \overline{\boldsymbol{x}}^{t-1})$. $\mathbf{D} = \frac{1}{L+1}\begin{bmatrix} \text{diag}(\boldsymbol{\lambda}^{t-1}) & \mathbf{0} \\ \mathbf{0}^* & 0 \end{bmatrix} + \frac{L}{(L+1)^2}\begin{bmatrix} \mathbf{a}\mathbf{a}^* & c\mathbf{a} \\ c\mathbf{a}^* & c^2 \end{bmatrix}$.
Solve the eigenproblem: $\mathbf{D} = \mathbf{R} \cdot \text{diag}(\boldsymbol{\lambda}') \cdot \mathbf{R}^{-1}$, $\boldsymbol{\lambda}'$ is sorted in decreasing order.
$\boldsymbol{\Psi}' = \left[\boldsymbol{\Psi}^{t-1} \ \mathbf{b}\right] \cdot \mathbf{R}$. Update $\boldsymbol{\Psi}^t$ as the first $K$ columns of $\boldsymbol{\Psi}'$. Update $\boldsymbol{\lambda}^t$ as the first $K$
values of $\boldsymbol{\lambda}'$. Update $\overline{\boldsymbol{x}}^t$ as $\left(L\overline{\boldsymbol{x}}^{t-1} + \boldsymbol{x}_{\text{intep}}\right)/(L+1)$.
```

Compared to the methods based on CS, our approach based on a low-dimensional model and OLS has the following advantages: i) the solution is easy to compute and it requires a single matrix inversion, ii) it enables an analysis of the reconstruction error and a consequent optimization of the sampling pattern $\boldsymbol{\chi}^t$ such that the upper-bound of ϵ is minimized.

4.4.2 Learning from Incomplete Data over Time

In Section 4.4.1, we highlighted some challenges regarding the estimation of the covariance matrix $\boldsymbol{C}_{\boldsymbol{x}}$ — a fundamental step for determining the approximation model $\boldsymbol{\Psi}^t$. Most of the challenges derived from the lack of a sufficiently large set of realizations of $\boldsymbol{x}$, that are needed to estimate $\boldsymbol{C}_{\boldsymbol{x}}$. First, there is virtually no past data for a newly installed WSN. Second, $\boldsymbol{C}_{\boldsymbol{x}}$ is likely to vary over time. Third, a high ratio of data points $(1-\gamma)$ are not available for the estimation because we collect sparse measurements. Therefore, we need an on-line algorithm that estimates and adaptively updates the covariance matrix $\boldsymbol{C}_{\boldsymbol{x}}$ from incomplete data.

The main difficulty is the lack of complete realizations of $\boldsymbol{x}$. Two strategies are generally considered to overcome such a problem. The first strategy is to estimate from $\boldsymbol{y}$ an interpolation $\boldsymbol{x}_{\text{interp}}$ by using classic methods such as linear, polynomial or spline interpolation. The second strategy skips the estimation of $\boldsymbol{C}_{\boldsymbol{x}}$ and attempts to perform directly the PCA on the data with missing entries, see [49].

From our quantitative results, the second class of algorithms is less powerful for our purposes. Therefore, we focus our attention on the interpolation methods. More precisely, we analyze two different methods that implement an adaptive learning and updating of the approximation model $\boldsymbol{\Psi}^t$ from the interpolated signal $\boldsymbol{x}_{\text{intep}}$.

The first method, whose pseudocode is given in Algorithm 4.2, uses a FIFO buffer to store the most recent L blocks. Whenever a new block is added into the buffer, the oldest block in the buffer is excluded. As the approximation model is estimated according to the signal realizations in the buffer, this scheme is able to capture over time the variation of signal statistics.

The second method, described in Algorithm 4.3, adaptively updates the approximation model via a technique called incremental PCA [32]. It does not keep signal realizations in memory, instead, it stores the largest K eigenvalues of $\boldsymbol{C}_{\boldsymbol{x}}$, $\boldsymbol{\lambda} = \{\lambda_i\}$, for $i = 1, \cdots, K$. This method requires significantly less memory (K versus $N \times L$), and shows better performance when compared to Algorithm 4.2. Note that in both algorithms, the choice of L depends on the variability of the signal statistics for each specific application. In practice, we can cross-validate this parameter to find a suitable value (e.g., $L = 30$). We discuss and compare the performance of these two algorithms in the experimental results section

4.4.3 Sampling-Schedule Algorithm

According to Proposition 4.1, minimizing the overall error ϵ is equivalent to finding the optimal sampling pattern $\boldsymbol{\chi}$ that minimizes (4.13). We fix the values of K and M in the optimization process, hence the approximation error ϵ_a is fixed. We assume that the model $\boldsymbol{\Psi}^t$ is sufficiently precise and the dimensions K is large enough so that the term due to the white noise σ is dominant. Note that if the approximation error decays exponentially fast with K, there exists always a small K such that $\epsilon_a \ll \sigma$. We will show in the experimental part that meteorological data exhibits such an exponential decay of the approximation error.

To optimize the scheduling pattern, we want to find the sampling pattern that minimizes the following cost function,

$$\Theta\left(\widetilde{\boldsymbol{\Psi}}^t\right) = \sum_{k=1}^{K} \frac{1}{\lambda_k}, \tag{4.14}$$

where λ_k are the eigenvalues of $(\widetilde{\boldsymbol{\Psi}}^t)^* \widetilde{\boldsymbol{\Psi}}^t$, and $\widetilde{\boldsymbol{\Psi}}^t = \boldsymbol{\Phi}^t \boldsymbol{\Psi}^t$. Note that this optimization is equivalent to finding the M rows of $\boldsymbol{\Psi}^t$ that forms the submatrix $\widetilde{\boldsymbol{\Psi}}^t$ with the smallest $\Theta(\widetilde{\boldsymbol{\Psi}}^t)$. However, it has been already shown that such optimization is NP-hard [19, 20] and has a complexity $\mathcal{O}\left(\binom{N}{M}\right)$, which is prohibitively high in practice.

Therefore, for solving the scheduling problem, we investigate approximate solutions that can be implemented efficiently. These approximate solutions are usually hard to find because the cost function $\Theta(\widetilde{\boldsymbol{\Psi}}^t)$ has many local minima that are arbitrarily far away from the global minimum. Therefore, proxies of $\Theta(\widetilde{\boldsymbol{\Psi}})$ are usually chosen as a cost function for the approximated algorithm with a twofold aim: (i) inducing an indirect minimization of $\Theta(\widetilde{\boldsymbol{\Psi}}^t)$ and (ii) being efficiently optimized by standard techniques, as convex optimization or greedy algorithms.

In order to solve the sampling scheduling problem, we extend the recent work [51] on near-optimal sensor placement. In [51], a generic linear inverse problem

$$\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{\alpha} \tag{4.15}$$

is considered, where $\boldsymbol{x}$ contains all the possible sensors locations, $\boldsymbol{\Psi}$ is a known linear model and $\boldsymbol{\alpha}$ a set of parameters that we want to estimate. Then, we are allowed to measure only a subset of elements of $\boldsymbol{x}$. These elements are selected based on the greedy minimization of a proxy: the frame potential [15]. Such a proxy is defined as

$$\mathrm{FP}(\boldsymbol{\Psi}^t, \mathcal{S}) = \sum_{i,j \in \mathcal{S}} |\langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle|^2, \tag{4.16}$$

where $\boldsymbol{\psi}_i$ is the i-th row of $\boldsymbol{\Psi}^t$ and $\mathcal{S}$ denotes the set of selected locations for sensing. Under some mild solutions, we prove that such an algorithm is near-optimal with respect to the RMSE of the solution. The detailed proof is available in [51], and here we provide an intuitive explanation about why the frame potential is a good proxy: the frame potential favors the rows of $\boldsymbol{\Psi}^t$ that are close to be orthogonal to each other. Therefore, the algorithm selects the sensor locations containing a large amount of information about the sensing field.

In this work, we note that, the sensor-placement problem and the sampling scheduling problem are extremely similar: we have a linear inverse problem and we want to estimate a set of parameters $\boldsymbol{x}$ using the least number of measurements $\boldsymbol{y}$ without compromising the RMSE of the estimation. Nonetheless, there are a set of differences characterizing the latter:

— the signal model $\boldsymbol{\Psi}^t$ in our sparse-sensing scenario is generally not known a-priori and is time-variant,

— the sensor placement is optimized according to the spatial correlation, whereas here we select sparse sensing-samples according to the temporal correlation,

— the sensor placement is determined at design-time, whereas the sampling schedule is time-varying and optimized at run-time.

The sampling-schedule algorithm proposed here is based on an equivalent greedy “worst-out” procedure: as input we have the signal model $\boldsymbol{\Psi}^t$ and we initially consider the identity matrix of size N as the sampling matrix $\boldsymbol{\Phi}^{t+1}$. At each iteration, we remove the row of $\boldsymbol{\Phi}^{t+1}$ that maximizes the cost function (4.16). After $N - M + 1$ iterations we are left with an optimized $\boldsymbol{\Phi}^{t+1}$ that has only M elements different from zero and has near-optimal performance when reconstructing $\boldsymbol{x}$ from the measurements $\boldsymbol{y}$. Note that if $\boldsymbol{\Psi}^t$ satisfies the conditions given in [51], the obtained sampling matrix $\boldsymbol{\Phi}^{t+1}$ recovers $\boldsymbol{x}$ from the measurements $\boldsymbol{y}$ with a near-optimal RMSE.

In most of the scenarios, the sampling schedule optimized according to the proposed greedy minimization of the frame potential, gives state-of-the-art performance. However, there exists scenarios where a uniform sampling-schedule could be better [66], such as when the temporal measurements are characterized by a low-pass spectrum. Therefore, at the end of the greedy optimization, we compare the RMSE obtained by the greedy and the uniform schedule and opt for the one with the smaller reconstruction error. Note that the reconstruction error cannot be computed exactly given the uncertainty on the approximation error and we use the expression provided by Proposition 4.1, which bounds the RMSE for any given sampling matrix $\boldsymbol{\Phi}^t$.

A detailed description of the overall algorithm is given in Algorithm 4.4. Note that for the very first block of data upon system startup, the uniform sampling-schedule is used for initialization.

4.5 Comparisons with Baseline Methods

In this section, we briefly summarize the state-of-the-art methods for the sparse-sensing problem. They will serve as the baseline for comparisons in Section 4.6.

The first category of methods [48, 66] is based on compressive sensing (CS). With the notations introduced in Section 4.3, $\boldsymbol{x}$ is the unknown signal, $\boldsymbol{y}$ contains the incomplete measurements, and $\boldsymbol{\Phi}$ is a sparse-sampling matrix with only M elements different from zero. We assume $\boldsymbol{x}$ to be sparse with respect to a dictionary $\boldsymbol{\Pi}$. More precisely, we have $\boldsymbol{x} = \boldsymbol{\Pi}\boldsymbol{s}$ and $\boldsymbol{s}$ has just a few coefficients different from zero, that is $\|\boldsymbol{s}\|_0 \ll N$ (see [11] for more details). By approximating

Algorithm 4.4: Greedy sampling-schedule

```
Input: $\boldsymbol{\Psi}^t$, $M$
Output: $\boldsymbol{\chi}^{t+1}$ for the next temporal block
1 Initialize the set of removed sampling indices: $\mathcal{L} = \emptyset$. Initialize the set of selected sampling indices: $\mathcal{S} = \{1, \cdots, N\}$. Find the first two rows to eliminate, $\mathcal{L} = \arg\max_{i,j \in \mathcal{S}} |\langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle|^2$. Update $\mathcal{S} = \mathcal{S} \backslash \mathcal{L}$. repeat
2     Find the optimal row, $i^* = \arg\max_{i \in \mathcal{S}} \mathrm{FP}(\boldsymbol{\Psi}^t, \mathcal{S} \backslash i)$. Update the set of removed indices, $\mathcal{L} = \mathcal{L} \cup i^*$. Update the set of selected indices, $\mathcal{S} = \mathcal{S} \backslash i^*$.
3 until $|\mathcal{S}| = M$;
4 $\boldsymbol{\chi}^{t+1} = \arg\min_{\boldsymbol{\chi}} \left\{ \frac{\epsilon_a^2}{\lambda_K} + \sigma^2 \Theta(\widetilde{\boldsymbol{\Psi}}^t), \boldsymbol{\chi} \text{ is uniform pattern or } \mathcal{S} \right\}$.
```

Table 4.1: Summary of methods used in experiments

Abbreviation	Reconstruction Algorithm	Sampling Schedule
CS [48, 66]	(4.17)	uniform
CSN [12, 48]	(4.18)	uniform
OLS-random [48]	Alg. 4.1	random
OLS-uniform [48]	Alg. 4.1	uniform
DASS	Alg. 4.1	Alg. 4.4

the ℓ_0-norm with the ℓ_1-norm [12], the reconstruction method for the noiseless case is

$$\min_{\boldsymbol{s} \in \mathbb{R}^N} \|\boldsymbol{s}\|_1, \text{ s.t. } \boldsymbol{y} = \boldsymbol{\Phi \Pi s}, \tag{4.17}$$

whereas the one for the noisy case is

$$\min_{\boldsymbol{s} \in \mathbb{R}^N} \|\boldsymbol{s}\|_1, \text{ s.t. } \|\boldsymbol{y} - \boldsymbol{\Phi \Pi s}\|_2 \leq \xi, \tag{4.18}$$

where ξ measures the energy of the noise. Problem (4.17) and (4.18) are both convex and can be solved [12] in polynomial time using various solvers, in general iterative or based on convex optimization. In both methods, we use uniform sampling as the sampling scheduler — $\chi_j^t = \lfloor jN/M \rfloor$.

The second category of baseline methods [48] are based on learning the K-dimensional time-varying model $\boldsymbol{\Psi}^t$ and a reconstruction via an OLS as in Algorithm 4.1. We use two sampling schedulers, namely, a uniform sampling, and a random sampling where χ_j^t is randomly selected with a uniform distribution.

Table 4.1 lists all the methods (including DASS) that are evaluated in the experiments. To obtain a fair comparison, $\boldsymbol{\Pi}$ in CS-based methods and $\boldsymbol{\Psi}^t$ in OLS-based methods are both learned[8] by the incremental PCA described in Algorithm 4.3.

---

[8] The experimental results show that $K = M$ is the best choice for CS-based methods, and $K < M$ is a parameter that needs to be optimized for OLS-based methods, see Section 4.6.1.

Table 4.2: Summary of experimental datasets

Dataset name	Physical quantity	Number of nodes	Number of days
Payerne	temperature, solar radiation	1	1500
Valais	temperature	20	125

4.6 Evaluations of DASS and Sparse-Sensing Methods

In this section, we evaluate the performance of DASS and compare it with the state-of-the-art sparse-sensing methods. In addition to the experiments on the single-node case, we also verify DASS in the multi-node case where nearby sensor nodes measure spatially correlated signals. We use two real-world meteorological datasets as the ground truth, namely *Payerne* and *Valais*:

— *Payerne* is provided by MeteoSwiss [1]. This dataset contains 1500 days of continuous measurements for two physical quantities (temperature and solar radiation) [9], which are suitable for studying long-term performance of DASS. As MeteoSwiss only deployed a few observation stations across the whole nation, we use *Payerne* for evaluating the single-node case.

— *Valais* is provided by a microclimate-monitoring service provider [34]. A total of 20 stations are deployed in a mountain valley. Figure 4.7 shows six of them, covering an area of around 18 km^2. The deployments were started in March 2012 and collected 125 days of continuous temperature measurements. We use *Valais* for evaluating the multi-node case.

The two datasets are summarized in Table 4.2. For both datasets, there are 144 uniformly sampled data points for each day. We choose the day as the length of each block, that is, $N = 144$.

One of the targets of this section is to evaluate DASS and compare it with other algorithms when the sensing device induces measurement noise. As we do not know the groundtruth of the signal, we assume that *Payerne* and *Valais* represent the real value of the field $\boldsymbol{x}$. Then, we add white Gaussian noise to simulate the effect of noisy measurements. We evaluate the algorithms for the SNR of the measurement (as defined in (4.5)).

Note that the main merit figure considered in this section is the final reconstruction error under a fixed subsampling rate γ. Since all sparse-sensing schemes directly transmit the sensing samples without further data compression, two schemes with the same γ have the same amount of energy consumed for sensing and communication [10], regardless of which sensing platform is used.

4.6.1 Components of DASS

In this section, we evaluate the key components of DASS, including the optimal choice of K, the cost function $\Theta(\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t)$ in the sampling-schedule algorithm, and the performance of adaptive learning algorithms.

---

[9] We denote by *Payerne*-temperature the dataset of temperature measurements. The notation is similar for solar radiation.

[10] The processing costs of the considered sparse sensing methods are negligible.

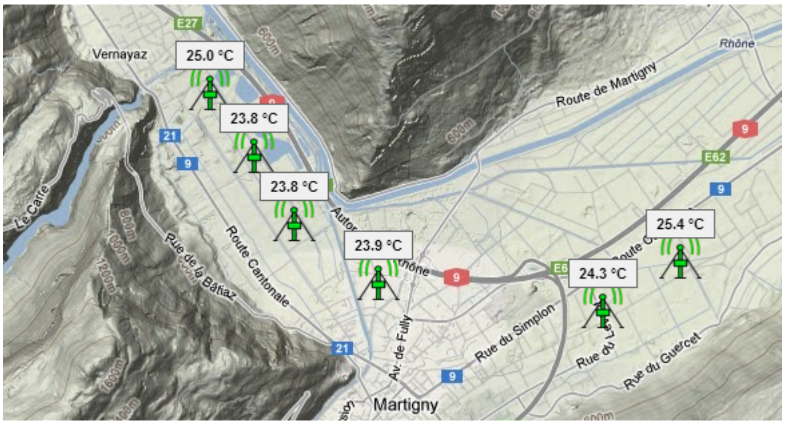


Figure 4.7: Locations of the sensor nodes that collected the data-set *Valais*.

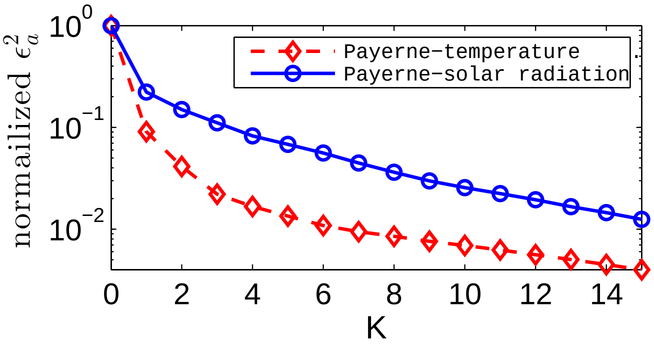


Figure 4.8: Normalized approximation error for the two considered datasets as a function of the model parameter K. Note how the error monotonically decreases with K given the optimality of PCA. Moreover, we highlight how the approximation error shows an exponential decay with K; thus confirming our assumption described in Section 4.4.3.

Optimal Choice of Dimension K: First, the larger the K, the smaller the approximation error for any dataset; the only difference being the decay rate of such an error. Such an aspect for the two considered dataset is depicted in Figure 4.8, where the data has been normalized for $K = 0$. Note that for both datasets we have an exponential decay of the approximation error as a function of K. Therefore, there exists a small K for which the approximation error is negligible with respect to the Gaussian noise corrupting the measurements, as we previously assumed.

As stated in Proposition 4.1, the overall reconstruction error ϵ is a function of both the approximation error ϵ_a (4.8) and the cost function $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ (4.14). Generally, ϵ_a decreases with K and $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ increases with K, hence there is an optimal choice of K for minimizing the overall error. The optimal K depends on the data statistics, the subsampling rate, and the SNR of the measurement. By cross-validation, Figure 4.9 shows the optimal ratio K/M for *Payerne*-temperature. We see that as DASS generally opts for a larger K when the SNR of measurement increases. This is intuitive since with better measurements we can afford a more complicated model with a weaker regularization.

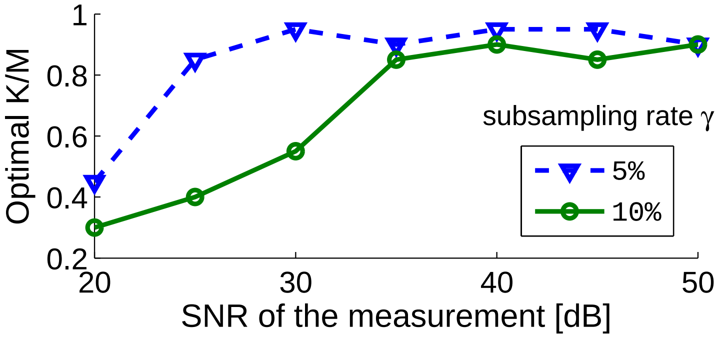


Figure 4.9: Optimal ratio K/M of DASS for a fixed subsampling rate with respect to the SNR of the measurement (*Payerne*-temperature dataset). First, we note that K/M must be smaller than 1 according to Proposition 4.1. Second, we note that for an increasing quality of the measurements we can collect only $M \approx K$ samples, meaning that the reconstruction algorithm is less influenced by the noise hence we need less samples. As a conclusion, we would expect the plots to be monotonically increasing. However, this is not the case due to the random nature of the noise model and to the near-optimality of scheduling algorithm.

Table 4.3: Average $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ achieved by different sampling-schedule algorithms ($\gamma = 10\%$, SNR of the measurement=30dB)

*Payerne*Method	uniform	random	Alg. 4.4
Temperature	0.56	4.9×10^{15}	0.54
Solar radiation	4.5×10^{5}	1.8×10^{15}	0.97

Sampling Schedule: The greedy algorithm proposed in Section 4.4.3 (Algorithm 4.4) finds an approximate solution of the sampling-schedule problem. By Proposition 4.1, $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ determines the reconstruction error. Table 4.3 shows the value of $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ achieved by different sampling-schedule algorithms for different datasets. Note that a high value indicates poor stability with respect to noise. We can see that the greedy algorithm achieves the best result for the two datasets. In particular, it is substantially better than uniform sampling for solar radiation data. For temperature data, as $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ of the uniform sampling strategy is already near the lower bound [11], the greedy algorithm provides little improvement. In the next section, we demonstrate how these improvements translate into an improved reconstruction performance of DASS.

Learning over Time: DASS is designed to learn the signal statistics from past data. In practical scenarios, a long backlog of data is usually unavailable and hence DASS should be designed to learn the model from scratch. We propose Algorithm 4.2 and Algorithm 4.3 for this task. Figure 4.10 shows the learning curves of these two algorithms over three years of data. As a benchmark, we consider an offline method that learns the model from the data of 600 days and is represented by the red-dotted curve. The offline method derives the transform matrix once and for all from the complete signal. However, using this matrix might provide poor results as the signal can have substantial variations over different time blocks.

---

[11] The lower bound of $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ is $\gamma = M/N$ if and only if $\mathbf{\Phi}^t\mathbf{\Psi}^t$ is a basis.

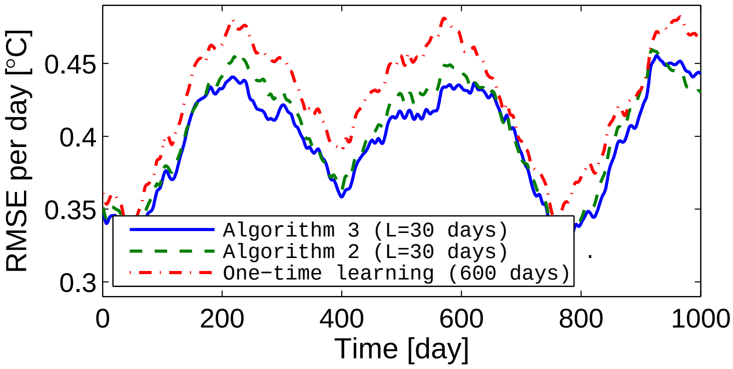


Figure 4.10: Learning curves of DASS (*Payerne*-temperature, $\gamma = 10\%$, SNR of the measurement=30dB): Comparisons of two online learning algorithms and an one-time learning algorithm with long backlog of past data. Note that Algorithm 4.3 achieves always the lowest error.

Note how Algorithm 4.2 and Algorithm 4.3 capture the signal statistics. In particular, even if they use less data—the last 30 days—they are generally better than the offline method that considers 600 days of data. It is clear that the non-stationary signal model $\mathbf{\Psi}^t$ is captured only by the adaptive on-line algorithms. Moreover, Algorithm 4.3 with incremental PCA performs better than the buffer-based Algorithm 4.2.

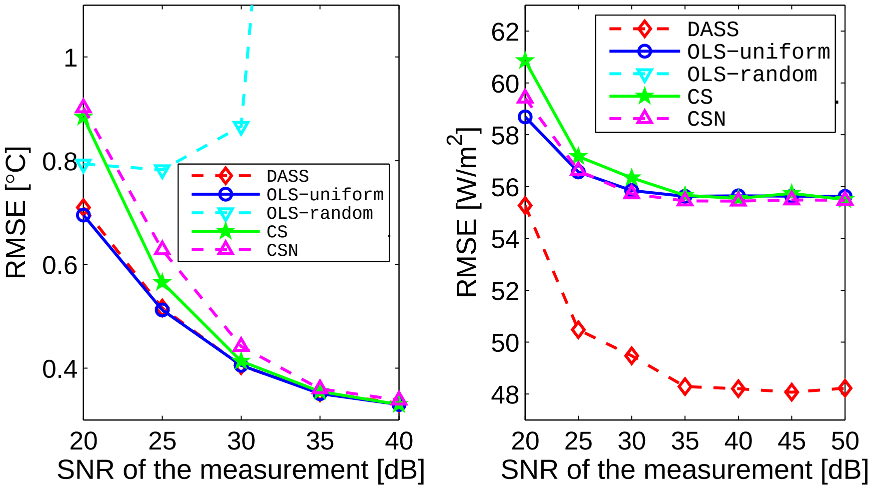
In the following experiments, we will consider only Algorithm 4.3 due to its good performance and low memory requirements.

4.6.2 DASS versus Baseline Methods

We compare DASS with the baseline methods introduced in Table 4.1, namely, CS, CSN, OLS-random, and OLS-uniform.

Known Noise Level: For DASS, we need to choose the optimal K according to the cross-validation studied in Figure 4.9. A similar parameter tuning is necessary for CSN, where ξ in equation (4.18) represents the noise level. Therefore, whenever we consider the case of noisy measurements, an estimate of the SNR of the measurement is necessary to avoid degradations of the reconstruction quality.

In the first experiment, we assume that the estimation of the SNR is exact. Figure 4.11 shows the comparison results of DASS, OLS-uniform, OLS-random, CS and CSN, for both temperature and solar radiation data. First, note that OLS-uniform generally performs better than the two CS-based schemes, especially in a low SNR regime. In a high SNR regime ($>$ 35dB), OLS-uniform, CS and CSN tend to perform similarly. Second, the bad performance of OLS-random indicates that random sampling is not a valid sampling strategy for either temperature or solar radiation signals. Third, although DASS and OLS-uniform perform almost equivalently for temperature data, we note that DASS is substantially better for solar radiation data. This fact is in accordance with the analysis of $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ given in Table 4.3: If $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ is large due to uniform sampling (see solar radiation data), then the sampling-schedule algorithm of DASS (Algorithm 4.4) significantly improves the effectiveness of sensing and preserves the average



(a) (b)

Figure 4.11: Reconstruction error (RMSE) with respect to SNR of the measurement, of DASS, OLS-uniform, OLS-random, CS and CSN, respectively ($\gamma = 10\%$). The SNR is assumed to be accurately estimated. (a) *Payerne*-temperature. (b) *Payerne*-solar radiation. DASS is either on par with the best method, see (a), or significantly better, see (b). Note that in (b) OLS-random is not visible in the plot because it is significantly worse than the other methods.

sampling rate.

Error in Noise Estimation: In practice, the estimation of the noise level might be inexact. Here, we study the performance deviation of the considered algorithms when there is an error in such estimates. More precisely, we fix all the parameters and we vary the estimation error of the SNR and then measure the performance of the algorithms in terms of the RMSE.

Figure 4.12 shows the reconstruction error with respect to the estimation error of the SNR, whereas the true SNR is 30dB. We can see that DASS performs the best, and generally DASS and OLS-uniform are both stable with respect to errors in the SNR estimation. However, the performance of CSN degrades severely when the SNR is underestimated. The reason behind this large gap is that DASS and OLS-uniform both solve a least square problem (4.10), which could automatically reveal the unknown noise variance after optimization. On the contrary, CSN requires a known noise variance in the objective function, hence it can be affected severely if the SNR is not correctly estimated a priori.

According to results given in Figure 4.11 and Figure 4.12, DASS is both more *accurate* and *robust* when compared to the state-of-the-art sparse-sensing methods.

4.6.3 DASS on Multiple Sensor Nodes

As discussed in Section 4.3, the concept of DASS can be extended to multiple sensor nodes by concatenating the collected samples into a single vector $\boldsymbol{y}$ and using the same strategy as for the single-node case.

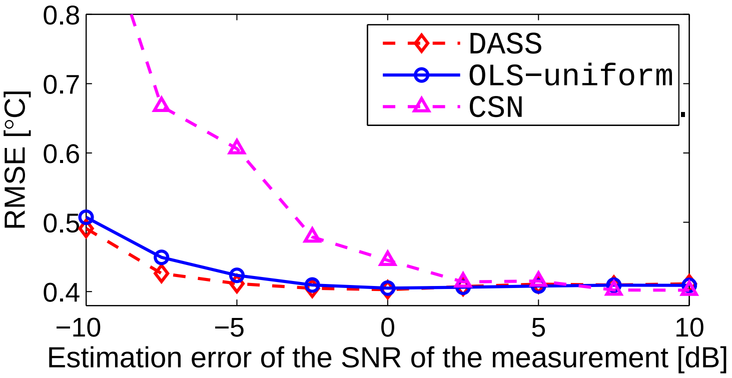


Figure 4.12: Reconstruction error (RMSE) with respect to estimation error of the SNR of the measurement, of OLS-uniform, DASS and CSN, respectively (*Payerne*-temperature, $\gamma = 10\%$). The true SNR is 30dB. Note that the proposed method is more robust to errors in the estimation of the noise power, when compared to other methods.

Merging the data of all the spatial nodes possibly augments the correlation; DASS may exploits such a correlation in order to reduce the sampling rate. In fact, if all the measurements collected by the sensors are linearly independent then DASS generates the same sampling-schedule that would have been optimized for each sensor individually. However, if there exists some correlation between the different sensor nodes, then DASS jointly optimizes the sensor scheduling so that the total average sampling rate is reduced.

We denote by *Joint-DASS* the scheme that reconstructs the signals of all sensors together (Figure 4.5), and *Independent-DASS* the scheme that independently reconstructs the signals of each sensor. Note that in both schemes, sensor nodes are operating in a purely distributed manner; the difference is that *Joint-DASS* aggregates the sensed data of all nodes and jointly processes them.

Figure 4.13 shows the ratio between the subsampling rates of *Joint-DASS* and *Independent-DASS*, using the data-set *Valais*. We only show up to six sensors because the benefit stabilized at 30% with more than 4 sensors in the experiments. We can see that as the number of sensors increases, the required sampling rate of *Joint-DASS* gradually decreases. In particular, with 4 sensors we can reduce the number of samples by 70% with *Joint-DASS*. Therefore, exploiting the spatial correlation further enhances the energy reduction of DASS. On the other hand, the benefit flattens out when we consider 5 or more sensors. The intuition behind this phenomenon is that as the number of sensors increases, there are more sensors far apart from each other and hence the spatial correlations reduce accordingly.

4.7 Energy Saving over Traditional Data Collection Schemes

In Section 4.6, we show that DASS achieves better sensing precision with respect to the state-of-the-art *sparse-sensing schemes*. In this section, we study the *overall energy saving* of DASS with respect to the *traditional data collection schemes* [52, 71]. The energy saving is particularly significant on platforms where the energy consumed for sensing is more pronounced. This is

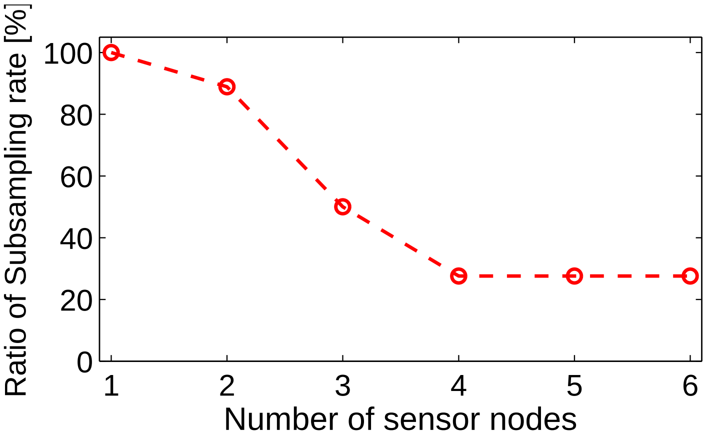


Figure 4.13: Ratio of sampling rate between *Joint-DASS* and *Independent-DASS*, such that both schemes have the same reconstruction error (*Valais*, SNR of the measurement=20dB). Note that the joint scheme always reduces the number of samples required, this is due to the spatial correlation available in the sampled data.

intuitive since DASS can substantially reduce the number of sensing samples. Nevertheless, our analysis shows that this saving is also noticeable on platforms with small sensing cost, e.g. a *Tmote-sky* node [64].

The traditional data collection schemes typically sample the signal at a high frequency f as in (4.1) and then compress the samples to reduce the communication rate, see Figure 1.5a. In contrast, DASS collects measurements by using an optimized sampling pattern and a reduced average sensing frequency $\gamma \cdot f$, where $\gamma < 1$. Then, each sensor node transmits the raw data points without any compression, see Figure 1.5b. In both traditional schemes and DASS, we aim at precisely reconstructing the signal $\boldsymbol{x}$. Furthermore, we restrict the discussion to the single-node scenario.

It is clear that DASS reduces the energy consumption for the sensing operations over the traditional scheme. However, DASS might not necessarily consume less communication energy, because the compression ratio $\mathbf{r}_c$ [12] used in traditional sensing is generally better than $1/\gamma$. In fact, existing data-compression schemes can achieve a compression ratio $\mathbf{r}_c$ of $1.5 \sim 5$ for lossless coding [52], and $5 \sim 50$ for lossy coding [71], whereas a typical value of γ used in DASS is 0.1. Hence, there is a tradeoff between the energy saved on sensing and communications.

Such a tradeoff between the different energy-consumption depends on platform-specific parameters. In particular, we denote the energy consumption for collecting and transmitting one sample as E_{sensor} and E_{radio}, respectively. An interesting figure is the ratio between the two energy values, that we denote as $\mathbf{r}_s = E_{sensor}/E_{radio}$. Intuitively, the larger $\mathbf{r}_s$, the larger the energy savings obtained by DASS. For the traditional data collection schemes, we assume that the compression step has a negligible energy cost. For DASS we use a subsampling rate of $\gamma = 0.1$, which means that 10% of the original signal is sampled and transmitted.

Under these assumptions, we can quantitatively analyze the relative energy savings of DASS with respect to the traditional sensing as a 2-D function of the platform parameter $\mathbf{r}_s$ and the

---

[12] $\mathbf{r}_c$ equals uncompressed size / compressed size.

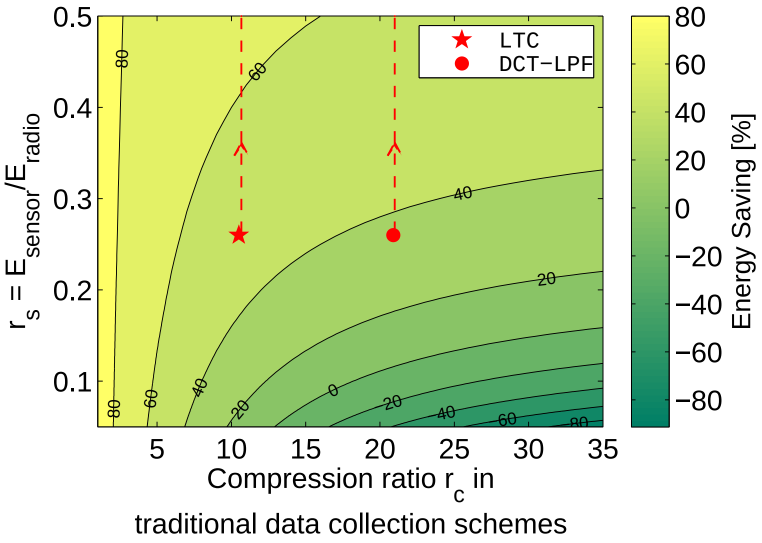


Figure 4.14: Relative energy saving of DASS ($\gamma = 10\%$) with respect to traditional data collection schemes. The saving depends on the sensing platform (value of $\mathbf{r}_s$) and the compression ratio $\mathbf{r}_c$ in traditional sensing. The "star" and "circle" markers represent the energy saving on *Tmote-sky*, when DASS achieves the same reconstruction error as traditional sensing using LTC and DCT-LPF compression methods [71] (on dataset *Payerne*-temperature) . The dashed lines indicate further savings when $\mathbf{r}$ increases, that is for sensors with higher energy costs.

compression ratio $\mathbf{r}_c$ achieved by the compression stage of the traditional scheme. Such function representing the energy saving is plotted in Figure 4.14. We see that there is a line, indicated by the zero value, that defines where DASS is more energy-efficient than the traditional schemes. Above the line, a WSN consumes less energy if it uses DASS and vice versa. Note that DASS is only less efficient in the scenarios where the compression ratio $\mathbf{r}_c$ is very high and the platform parameter $\mathbf{r}_s$ is very low.

As discussed in Section 4.1, we consider *Tmote-sky*, a low-power sensing platform widely used in WSNs [64]; it has a photodiode sensor that measures the light intensity of the surroundings and communicates with others through short-range radio. We measured the two energy consumptions E_{sensor} and E_{radio} of *Tmote-sky* in a set of experiments, and the measurements are given in Figure 4.1. The experiments indicate that $\mathbf{r}_s = 0.26$. To evaluate the energy consumption of a traditional scheme, we need to choose a specific compression algorithm and measure the achieved $\mathbf{r}_c$. Zordan et al. [71] have recently compared various lossy compression algorithms and showed that DCT-LPF [71] achieves the best performance in terms of compression ratio. However, it is also a complex algorithm and has a significant energy-consumption on a resource-limited platform such as *Tmote-sky*. Therefore, we also consider a lightweight algorithm, LTC [53], that achieves the lowest energy-consumption on WSN nodes if the energy cost for compression is considered.

Here, we ignore the energy cost of compression and we compare both algorithms with DASS. Note that, if we consider computational energy cost, the benefit of DASS will be even larger since it requires minimal on-board computation. We implement and evaluate the two algorithms on the dataset *Payerne*-temperature, and record the corresponding compression ratio $\mathbf{r}_c$ when their reconstruction errors are the same as those achieved by DASS.

The “star” and “circle” markers in Figure 4.14 show the energy savings of DASS over a *Tmote-sky* that compresses the data with LTC and DCT-LPF, respectively. The energy savings for the two cases are equal to 50% and 35%. It is worth mentioning that the compression ratios achieved in Figure 4.14 (“star” and “circle” markers) are specific of the considered meteorological datasets. There might be extreme cases where traditional compression schemes achieve a very high compression ratio (e.g., $r_c = 100$), and the respective saving falls below zero. However, we observe in Figure 4.14 that the energy savings can still be obtained in such cases, if $\mathbf{r}_s$ increases due to a higher energy cost for sensing, as denoted by the dashed lines. This scenario could be realistic for many scenarios, for example, when the sensors are air pollution sensors or anemometers as discussed in Section 4.1.

4.8 Conclusions

In this chapter, we proposed DASS, a novel approach for sparse sampling that optimizes sparse-sampling patterns for precisely recovering temporal signals. DASS is based on three main blocks. First, it adaptively learns the signal statistics from past data. Second, it dynamically adjusts the sampling pattern according to the time–varying signal statistics. Third, it recovers the signal from the limited amount of collected samples and according to the learned signal statistics.

We demonstrated the effectiveness of DASS through extensive experiments using two real-world meteorological datasets. The results show significant improvements over the state-of-the-art methods. These improvements are more pronounced in the presence of significant spatial and/or temporal correlation in the sampled data by WSN.

We evaluated DASS on static WSNs; however, DASS is flexible and can be applied to other sensing scenarios such as mobile WSNs. For instance, sensors are installed on top of buses for collecting various environmental data along their trajectories [2]. The collected samples show strong temporal-correlations due to the fixed routes periodically taken by the buses, and thus, DASS is likely to be efficient in such a case as well.

4.A Appendix

In this appendix, we show how DASS deals with a the scenario where the sensing noises are correlated.

We assume the noise $\boldsymbol{\omega}$ to have zero mean and a correlation matrix $\Sigma_{\boldsymbol{\omega}}$ and recall that our measurements are defined as $\boldsymbol{x} + \boldsymbol{\omega}$, where $\boldsymbol{x}$ is the actual physical signal that we are measuring.

We define $\boldsymbol{\Omega} = (\Sigma_{\boldsymbol{\omega}})^{-0.5}$ and we whiten the measurements according as

$$\boldsymbol{\Omega}(\boldsymbol{x} + \boldsymbol{\omega}) = \boldsymbol{x}' + \boldsymbol{\omega}',$$

where the noise $\boldsymbol{\omega}'$ is now i.i.d. Therefore, we note that, DASS can be easily applied to sensing scenarios where the noise is not i.i.d. by whitening the measured data before the processing.

Chapter 5

Sparse Sensor-Selection by Exploiting Spatial Correlations

> When I'm working on a problem, I never think about beauty. I think only how to solve the problem. But when I have finished, if the solution is not beautiful, I know it is wrong.
>
> *R. Buckminster Fuller*

In the previous chapter, we proposed a sparse-sensing scheme that exploits temporal correlations. Although this scheme can be extended to exploit spatial correlations, it uses a block-based selection algorithm and implicitly assumes that the spatio-temporal model of the signal is stationary in each time block. However, this assumption might not hold when the signal changes rapidly or the sensors themselves move fast. In this chapter, we will discuss a general sparse-sensing framework that exploits spatial correlations of the signal to be sensed.

5.1 Introduction

In both static WSNs or participatory sensing systems, sensors take samples of a sensing field and transmit the captured samples to a remote server for reconstructing the original field. Because taking each sample costs some resources on the sensors, *e.g.*, energy or communication bandwidth, it is desirable to take few of them (sparse sampling). We observe that the sensing samples usually have a strong inter-node correlation, in other words, a high spatial correlation. By exploiting this correlation, the number of sensing samples to be taken can be largely reduced, whereas a certain level of reconstruction precision can be ensured.

Most existing works that exploit spatial correlations aim at minimizing the number of sensing samples at a single time instant. In these works, an application-specific utility function is defined on the subset of the selected sensors. The utility function might be defined as the sensing coverage [41], the mutual information [31], the frame potential [51], or the log determinant of a confidence ellipsoid [38]. The objective is to minimize the number of selected sensors to ensure a certain utility in that time instant. This problem is usually NP-hard [40, 46]. Fortunately, all the aforementioned utility functions are submodular. Leveraging on this submodular property, we can employ greedy algorithms to select the sensors and guarantee that the number of the selected sensors is at most $\log M$ times the optimal number where M denotes the number of all candidate sensors.

However, in most practical WSNs or participatory sensing systems, the objective is not to select a minimal subset of sensors for sensing in a single time instant. Instead, the objective is to adaptively select the subsets of sensors in a continuous sensing period, such that a certain sensing utility is always satisfied and the length of this sensing period is maximized given a limited amount of resources available to all sensors. The idea is shown as in Figure 5.1. Compared to the traditional sparse-sensing problem in a single time instant, selecting the subsets of sensors in a continuous sensing period is more complicated due to the increased size of the optimization space. We have to adaptively decide when, how many, and which sensors need to be activated for sensing.

In the rest of this chapter, we will propose a generic framework for adaptively re-selecting a sparse subset of sensors that guarantee a certain utility. This framework is compatible with different application-specific utility functions and is easy-to-implement in both static WSNs and participatory sensing systems. In static WSN scenarios, we show that the proposed framework guarantees a lifetime at least $1/\log M$ of the optimum (M denotes the number of all sensors). In participatory sensing scenarios where sensors move in an uncoordinated manner, it is hard to guarantee a certain approximation-ratio; still, we verify the effectiveness of the proposed framework through extensive simulations.

5.2 Related Works

The problem of selecting sparse sensors by exploiting spatial correlations is studied in both static WSNs and participatory sensing systems. We will review the related works in these two types of systems separately.

In static WSN systems, most works on sparse sensor-selection aim at minimizing the number of sensing samples in a single time instant [31, 38, 41, 51]. Other works that aim at maximizing lifetime given limited amounts of resources [5, 8, 13, 14] only focus on the scenarios where the utility function is the coverage. In contrast, we consider a general framework for sparse sensor-selection where application-specific utility functions can be used. Moreover, the algorithms we propose are adaptive, which perfectly fit the WSNs in dynamic environments.

In participatory sensing systems, sensors move in an uncoordinated manner. The sparse sensor-selection problem becomes more complicated. In [57], an opportunistic algorithm is used, where the probability for selecting each sensor is adaptively changed based on the number of detected sensors in the same neighborhood. In [63], the authors propose a mechanism for automated mapping of urban areas that provide a virtual sensor abstraction to different applications. Sensors are adaptively selected by using mobility predictions, so that both the spatial coverage

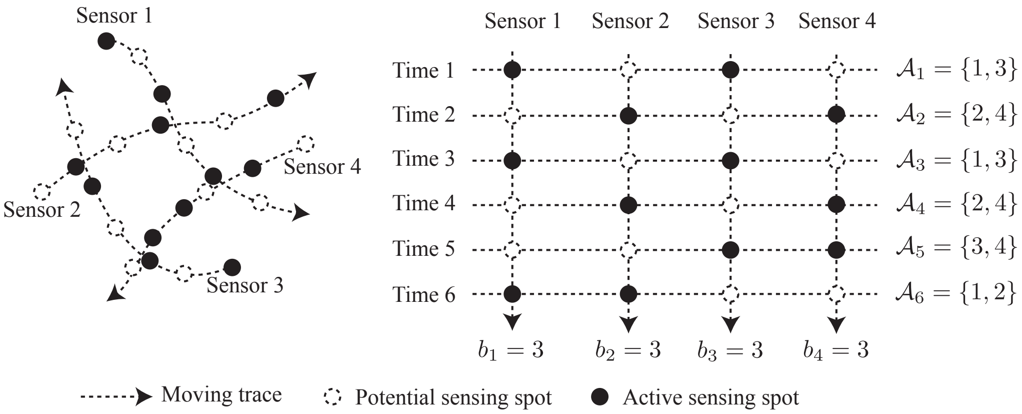


Figure 5.1: A participatory sensing system with 4 mobile sensors. We enhance its resource efficiency by exploiting spatial correlations of the sensing field. We do not control the mobility traces of sensors, but we adaptively decide when and which sensors should be activated for sensing. In this example, every sensor has a limited amount of resources and can be chosen at most three times for actively sensing, *i.e.*, $e_m = 3$ $(\forall 1 \leq m \leq 4)$. The set of all sensors is $\mathcal{M} = \{1, 2, 3, 4\}$. The set of selected sensors at time n is denoted by $\mathcal{A}_n$. The objective is to select a sequence of active subsets of sensors $\{\mathcal{A}_n\}_{n \in \mathbb{N}}$ to satisfy a given sensing-utility and to maximize sensing lifetime. In this example, the selected sequence is $\{\mathcal{A}_n\}_{n=1}^{6}$ and the sensing lifetime is 6.

and the temporal coverage are maintained. In [54], researchers solve the problem in the special case where the sensing field is one-dimensional on a straight road and where the mobility traces of sensors are known in advance. The performance of the proposed algorithm is limited when there exist scarce mobile sensors in the sensing field. In [67], the authors use an empirical method for sensor selection by analyzing the previous data samples. The data samples are first partitioned into segments which are assumed to be generated from different models. Then, within each segment, the optimal sensing points are selected. However, this method is only useful when both the mobility trajectories of sensors and the signal to be sensed are periodic.

5.3 Problem Formulations

We will use a generic formulation for sparse sensor-selection in both static WSNs and participatory sensing systems. Let the set of sensors be $\mathcal{M} = \{1, 2, \cdots, M\}$ where M denotes the number of sensors. In Figure 5.1, we show an example of a participatory sensing system with $M = 4$ mobile sensors. We partition time into slots while $n \in \mathbb{N}^+$ denotes the slot index and τ denotes the length of a time slot. For simplicity of discussion, we assume that the sensors do not move within each time slot. We denote the selected subset of sensors by $\mathcal{A}_n \subseteq \mathcal{M}$ in time slot n and we denote by $g_n(\mathcal{A}_n)$ the utility function of this set of sensors. The function $g_n(\cdot)$ might be defined in various ways depending on different applications, *e.g.*, sensing coverage [41], mutual information [31], frame potential [51], or log determinant of a confidence ellipsoid [38]. Notice that these utility functions $g_n(\cdot)$ enjoy the submodular property, that is,

$$g_n(\mathcal{A} \cup \{m\}) - g_n(\mathcal{A}) \geq g_n(\mathcal{B} \cup \{m\}) - g_n(\mathcal{B}), \forall \mathcal{A} \subset \mathcal{B} \subset \mathcal{M}, \forall m \in \mathcal{M} \backslash \mathcal{B}.$$

Also, notice that $g_n(\cdot)$ depends on the index of time slot n because the locations of the sensors might change over time.

Let q be the desired utility level for the selected sensors. We define $u_n(\mathcal{A}) = \min(g_n(\mathcal{A}), q)$ for any $\mathcal{A} \subseteq \mathcal{M}$. In each time slot n, the desired subset of selected sensors $\mathcal{A}_n$ has to satisfy

$$u_n(\mathcal{A}_n) = u_n(\mathcal{M}), \tag{5.1}$$

which amounts to enforcing that: (i) if the utility of all sensors $g_n(\mathcal{M}) < q$, all sensors in $\mathcal{M}$ need to be selected; (ii) otherwise, only a subset $\mathcal{A}_n \subseteq \mathcal{M}$ needs to be selected to ensure $g_n(\mathcal{A}_n) \geq q$. Notice that if $g_n(\cdot)$ is submodular, so is $u_n(\cdot)$. Likewise, since $g_n(\cdot)$ depends on $n \in \mathbb{N}^+$, so does $u_n(\cdot)$.

Denote the initially available resource of each sensor node $m \in \mathcal{M}$ by e_m. Without loss of generality, we assume that every sensor node has unit resource-consumption rate when activated for sensing; and it does not have any resource consumptions when inactivated. The objective of the sparse-sensing problem is to select a sequence of subsets of sensors $\{\mathcal{A}_n\}_{n=1}^N$, such that the lifetime of the network τN is maximized, that is,

$$\begin{aligned} \max_{\{\mathcal{A}_n\}_{n=1}^N} \quad & \tau N \\ \text{s.t.} \quad & u_n(\mathcal{A}_n) = u_n(\mathcal{M}), \forall 1 \leq n \leq N, \\ & \tau \cdot \sum_{n=1}^{N} \mathbb{I}(m \in \mathcal{A}_n) \leq e_m, \forall m \in \mathcal{M}, \end{aligned} \tag{5.2}$$

where the second constraint states that the total resource consumptions should not exceed the initially available resources. Let the optimal lifetime in problem (5.2) be τN_{opt}. Because N_{opt} depends on τ, we will also write $N_{\text{opt}}(\tau)$ for N_{opt} to show explicitly this dependency.

As far as we know, this problem is studied only in static WSN scenarios where the utility function is the coverage [5, 8, 13, 14]. In the following, we will propose an unified framework for solving problem (5.2) in both static WSN and participatory sensing systems where different utility functions can be employed.

5.4 The Proposed Framework

The proposed framework is generic in both static WSNs and participatory sensing systems. Before the start of each time slot, every sensor sends beacons to a server. These beacons contain the information about the remaining resources, the GPS location of the sensor and the routing path to the server. The server collects all beacons, uses all the available information to select a subset of sensors and notifies the selected sensors to be active. Then, the selected sensors start sensing and continue to upload the captured sensing samples during the whole time slot. Finally, the server collects all the sensing samples and uses these samples for reconstructing the original sensing field. This process is repeated in the next time slot. The overall process is shown in Figure 5.2.

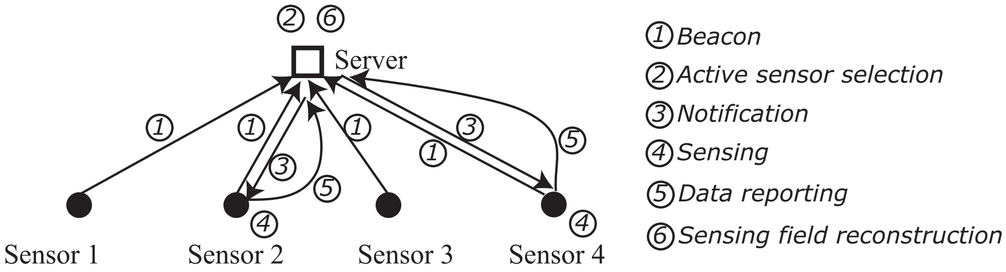


Figure 5.2: An example of the spatial sparse-sensing framework. In step 1, all sensors periodically send beacons to the server in order to signal their remaining resources and their GPS locations. In step 2, the server collects all the beacons and decides which sensors should be activated in the next time slot. In step 3, notifications are sent to the selected sensors. Then, the selected sensors start sensing in step 4 and continue to upload the sensing samples to the server in step 5. Finally, in step 6, the server collects all the sensing samples and reconstructs the original sensing field. In this example, four sensors participate in the sensing task, but only sensors 2 and 4 are activated for sensing in this time slot.

In the following, we will focus on the algorithm running on the server for selecting the sparse sensors. To start with, we introduce a few notations. We denote the available resource of sensor node m at time slot $n \in \mathbb{N}^+$ by $e_m^{(n)}$. In particular, the initially-available resource $e_m^{(0)} = e_m$. Because the available resource at time n equals the initially available resource minus the resource consumed in the first n time slots,

$$e_m^{(n)} = e_m - \tau \sum_{t=1}^{n} \mathbb{I}(m \in \mathcal{A}_t). \tag{5.3}$$

We define $\theta_m^{(n)}$ as the average resource-consumption rate of sensor node $m \in \mathcal{M}$ during the first $n \in \mathbb{N}^+$ time slots

$$\theta_m^{(n)} = \frac{1}{n} \sum_{t=1}^{n} \mathbb{I}(m \in \mathcal{A}_t) = \frac{e_m - e_m^{(n)}}{n\tau}, \tag{5.4}$$

where the second equality follows from (5.4).

The objective is to select a sequence of subset of sensors $\{\mathcal{A}_n\}_{n \in \mathbb{N}^+}$ that fulfils (5.1) in each time slot $n \in \mathbb{N}^+$, such that in the long run, the resources of all sensors are efficiently utilized. Our proposed method is to penalize the sensors that are frequently used, so that they get less selected afterwards.

The proposed algorithm is shown as Algorithm 5.1. It iteratively does the following in each time slot $n \in \mathbb{N}^+$ as long as the available resources $e_m^{(n-1)} \geq 0$ for all $m \in \mathcal{M}$:

(i) First, the server collects the information about the available resource of all sensors $\{e_m^{(n-1)}\}_{m \in \mathcal{M}}$, and calculates the resource-consumption rates in the first $n-1$ time slots $\{\theta_m^{(n-1)}\}_{m \in \mathcal{M}}$ using (5.4). We define the **penalty** for selecting each sensor node $m \in \mathcal{M}$ in time slot n as

$$\lambda_m^{(n-1)} = \frac{\exp\left(\alpha \theta_m^{(n-1)} / e_m\right)}{e_m \sum_{m \in \mathcal{M}} \exp\left(\alpha \theta_m^{(n-1)} / e_m\right)} \tag{5.5}$$

Algorithm 5.1: The proposed method for selecting subsets of sensors

```
Input: The initially available resource $\{e_m\}_{m\in\mathcal{M}}$, and a $\log M$-approximate algorithm for
       solving the min-penalty set-cover problem (5.6).
Output: $N^*$, $\{\mathcal{A}_n\}_{1\leq n\leq N^*}$.
Initialize $n \longleftarrow 1$.
while $e_m^{(n)} \geq 0, \forall m \in \mathcal{M}$ do
    Measure the available resource $e_m^{(n)}, \forall m \in \mathcal{M}$.
    Calculate the penalties $\{\lambda_m^{(n-1)}\}_{m\in\mathcal{M}}$ using (5.4) and (5.5).
    Select the subset of sensors $\mathcal{A}_n$ by using a $\log M$-approximate algorithm to solve the
    min-penalty set-cover problem (5.6).
    Update the available resource $e_m^{(n)}, \forall m \in \mathcal{M}$ following (5.3).
    Update $n \longleftarrow n+1$.
Set $N^* = n-1$.
```

Algorithm 5.2: The $\log M$-approximate algorithm for min-penalty set-cover problem

```
Input: The available resource of all sensors $\{e_m^{(n-1)}\}_{m\in\mathcal{M}}$
Output: A set of selected sensors $\mathcal{A}_n \subseteq \mathcal{M}$
Calculate the penalties $\{\lambda_m^{(n-1)}\}_{m\in\mathcal{M}}$ by using $\{e_m^{(n-1)}\}_{m\in\mathcal{M}}$, (5.4) and (5.5)
Initialize $\mathcal{A}_n \leftarrow \emptyset$
while $u_n(\mathcal{A}_n) \neq u_n(\mathcal{M})$ do
    Choose $m \in \mathcal{M}\backslash\mathcal{A}_n$ minimizing the weight per utility $\frac{\lambda_m^{(n-1)}}{u_n(\mathcal{A}_n \cup m) - u_n(\mathcal{A}_n)}$.
    Set $\mathcal{A}_n \leftarrow \mathcal{A}_n \cup m$.
```

where $\alpha > 0$ is a constant parameter. Notice that for any sensor $m \in \mathcal{M}$, the larger the average resource-consumption rate $\theta_m^{(n-1)}$ is , the larger the penalty $\lambda_m^{(n-1)}$ is. More details why we select this penalty function will be discussed in Section 5.5.

(ii) Then, the server selects the subset of sensors $\mathcal{A}_n$ that minimizes the sum of all penalties for the selected sensors

$$\begin{aligned} \min_{\mathcal{A}_n} \quad & \sum_{m\in\mathcal{A}_n} \lambda_m^{(n-1)} \\ \text{s.t.} \quad & u_n(\mathcal{A}_n) = u_n(\mathcal{M}). \end{aligned} \tag{5.6}$$

Problem (5.6) is a well-known min-penalty set-cover problem [18] and it can be solved by using a $\log M$-approximation algorithm, as repeated in Algorithm 5.2.

Denote by N^* the maximum number of time slots before the first sensor depletes its resource,

$$N^* = \max\{n | e_m^{(n)} \geq 0, \forall m \in \mathcal{M}\}. \tag{5.7}$$

Because N^* depends on τ and α, we will also write $N^*(\tau, \alpha)$ for N^* to show explicitly this dependency.

5.5 Analyses in the Static WSN Scenarios

In this section, we will analyze the performance of the proposed sparse-sensing framework only in static WSNs. The analysis of this framework in the participatory sensing systems is left for future work.

We will derive a lower-bound for the lifetime achieved by using our proposed algorithms in these scenarios. First of all, we will introduce an auxiliary problem that approximates the problem (5.2) with arbitrarily high precision when τ is small.

Remember that in static WSN scenarios, the GPS locations of all sensors are fixed, and the utility function $u_n(\cdot)$ does not change in different time slots $n \in \mathbb{N}^+$. Denote the constant utility function by $u(\cdot) = u_n(\cdot), \forall n \in \mathbb{N}^+$. Denote by $\mathcal{M}(q) \subseteq 2^{\mathcal{M}}$ the set of all subsets of $\mathcal{M}$ that satisfies (5.1),

$$\mathcal{M}(q) = \{\mathcal{A} | u(\mathcal{A}) = u(\mathcal{M}), \mathcal{A} \subseteq \mathcal{M}\}.$$

For each subset of sensors $\mathcal{A} \in \mathcal{M}(q)$, denote by $p_{\mathcal{A}}$ the fraction of time that $\mathcal{A}$ is used. Denote the average resource-consumption rate of each sensor node $m \in \mathcal{M}$ by

$$\theta_m = \sum_{\mathcal{A} \in \mathcal{M}(q)} p_{\mathcal{A}} \cdot \mathbb{I}(m \in \mathcal{A}).$$

Notice that the lifetime of the WSN is $\min_{m \in \mathcal{M}} e_m/\theta_m$ which is the inverse of $\max_{m \in \mathcal{M}} \theta_m/e_m$, and that $\max_{m \in \mathcal{M}} \theta_m/e_m$ can be approximated by

$$f(\boldsymbol{\theta}) = \frac{1}{\alpha} \log \left(\sum_{m \in \mathcal{M}} \exp \left(\frac{\alpha \theta_m}{e_m} \right) \right) \tag{5.8}$$

where $\boldsymbol{\theta}$ denotes the vector which lists $\{\theta_m\}_{m \in \mathcal{M}}$, and α is a constant parameter. The approximation is arbitrarily precise when α is large.

Therefore, we have the auxiliary problem

$$\begin{aligned} \min_{\{p_{\mathcal{A}}\}_{\mathcal{A} \in \mathcal{M}(q)}, \boldsymbol{\theta}} \quad & f(\boldsymbol{\theta}) = \frac{1}{\alpha} \log \left(\sum_{m \in \mathcal{M}} \exp \left(\frac{\alpha \theta_m}{e_m} \right) \right) \\ \text{s.t.} \quad & \theta_m = \sum_{\mathcal{A} \in \mathcal{M}(q)} p_{\mathcal{A}} \cdot \mathbb{I}(m \in \mathcal{A}), \forall m \in \mathcal{M} \\ & \sum_{\mathcal{A} \in \mathcal{M}(q)} p_{\mathcal{A}} = 1, \\ & p_{\mathcal{A}} \geq 0, \forall \mathcal{A} \in \mathcal{M}(q). \end{aligned} \tag{5.9}$$

Denote the optimal objective value of (5.9) by f_{opt}. The link between f_{opt} and the optimal value N_{opt} in problem (5.2) is given in the following lemma.

Lemma 5.1

The optimal objective value f_{opt} of the auxiliary problem (5.9) is bounded by

$$\frac{1}{\tau N_{\text{opt}} + \tau} \leq f_{\text{opt}} \leq \frac{1}{\tau N_{\text{opt}}} + \frac{\log M}{\alpha},$$

where N_{opt} is the optimal objective value of problem (5.2).

The proof can be found in Appendix 5.A.1.

Remember that in our proposed framework, we use Algorithm 5.1 for selecting the active sensors. In each time slot $n \in \mathbb{N}^+$, the penalty $\lambda_m^{(n-1)}$ of sensor node $m \in \mathcal{M}$ is indeed the gradient of $f(\boldsymbol{\theta}^{(n-1)})$, that is,

$$\lambda_m^{(n-1)} = \frac{\partial f(\boldsymbol{\theta}^{(n-1)})}{\partial \theta_m^{(n-1)}},$$

where $\boldsymbol{\theta}^{(n-1)}$ denotes the vector that lists $\{\theta_m^{(n-1)}\}_{m\in\mathcal{M}}$. Therefore, solving the min-penalty set-cover problem in each time slot n is equivalent to updating the objective value $f(\boldsymbol{\theta}^{(n-1)})$ in the steepest descent direction. Remember also that we use the $\log M$-approximate Algorthm 5.2 for solving this problem. Relying on the convexity of the function $f(\boldsymbol{\theta}^{(n-1)})$, we derive an upper-bound for $f(\boldsymbol{\theta}^{(N)})$ after running Algorithm 5.1 and Algorithm 5.2 for N ($N \in \mathbb{N}^+$) time slots.

Theorem 5.1

Using Algorithm 5.1 for selecting the subsets of sensors $\{\mathcal{A}_n\}_{n\in\mathbb{N}^+}$ with a β-approximation algorithm for solving the min-penalty set-cover problem (5.6), the achieved objective value $f(\boldsymbol{\theta}^{(N)})$ in time slot N satisfies

$$f\left(\boldsymbol{\theta}^{(N)}\right) - \beta f_{\text{opt}} < \frac{\alpha M^2(\log N + 1)}{2e_{\min}^2 N}. \tag{5.10}$$

where $\beta \geq 1$ denotes the guaranteed approximation ratio for solving the min-penalty set-cover problem (5.6), f_{opt} denotes the optimal objective value of (5.9) and $e_{\min}$ denotes $e_{\min} = \min_{m\in\mathcal{M}}\{e_m\}$.

The proof can be found in Appendix 5.A.2.

Using the results in Theorem 5.1 and plugging in $\beta = \log M$ (the approximation ratio for solving the min-penalty set-cover problem guaranteed by Algorithm 5.2), we derive a lower-bound for the achieved lifetime by using our proposed framework.

Theorem 5.2

Using Algorithm 5.1 for selecting the subsets of sensors $\{\mathcal{A}_n\}_{n\in\mathbb{N}^+}$ with Algorithm 5.2 for solving the min-penalty sensor-selection problem (5.6), the achieved number of time slots $N^*(\tau, \alpha)$ satisfies

$$\lim_{\alpha\to\infty}\lim_{\tau\to 0}\frac{N^*(\tau,\alpha)}{N_{\text{opt}}(\tau)} \geq \frac{1}{\log M},$$

where $N_{\text{opt}}(\tau)$ is the optimal number of time slots defined in problem (5.2).

The proof can be found in Appendix 5.A.3.

5.6 Performance Evaluation

In this section, we will evaluate the proposed framework that selects sparse sensors by exploiting spatial correlations. As discussed in Section 5.4, we partition time into slots and run certain algorithms to select the sensors in each time slot. We compare the performance of the following three algorithms for selecting the sparse sensors:

- The first algorithm is a randomized algorithm. In each time slot, we iteratively select the sensor nodes uniformly at random and add them to the set of active sensors, until the coverage constraint is satisfied. We call this algorithm **Random**.
- The second algorithm is a greedy algorithm. In each time slot, we iteratively select the sensors with the highest amount of remaining resources and add them to the set of active sensors, until the coverage constraint is satisfied. We call this algorithm **MaxRe**.
- The third algorithm is our proposed algorithm, as discussed in Section 5.4. We adaptively update a penalty for each sensor node and solve a min-penalty set-cover problem. We call this algorithm **MinPenalty**.

First, we consider two ideal sensing scenarios where the sensors are either static or moving according to a Manhattan mobility model (the details will be discussed shortly). Then, we consider two practical sensing scenarios where the mobile sensors are installed on either taxis or public transportation vehicles.

5.6.1 Ideal Scenarios

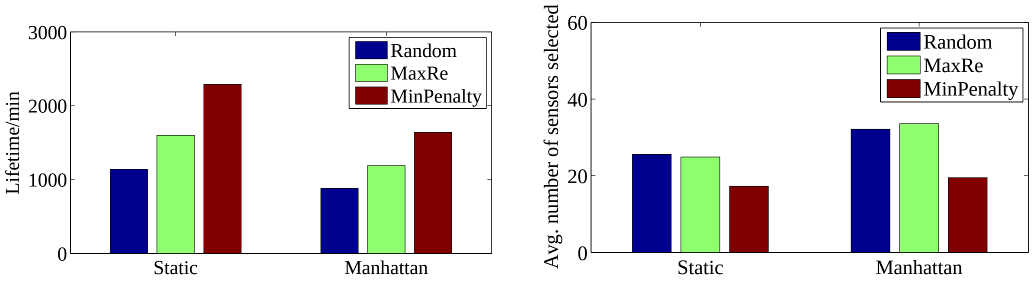
Consider a 10km × 10km sensing field. It is discretized into a 50 × 50 grid where the size of each cell is 200m × 200m. Assume that there are $M = 100$ sensor nodes. At the very beginning, we position all sensor nodes uniformly at random in the sensing field. Time is discretized into slots with length $\tau = 10$min. Depending on two different mobility models of the sensor nodes, we have the following two scenarios:

- The first ideal scenario is called **"Static"**. In this scenario, all the sensor nodes have fixed locations and they do not move throughout the sensing lifetime.
- The second ideal scenario is called **"Manhattan"**. In this scenario, all the sensor nodes move according to a Manhattan mobility model: in each time slot, each sensor node has an equal probability 1/9 to move to the adjacent 8 cells, or to stay at its current location.

The sensing radius of each sensor node is assumed to be 1km. The utility function $u_n(\mathcal{A})$ ($n \in \mathbb{N}^+$, $\mathcal{A} \subseteq \mathcal{M}$) is defined to be the coverage, *i.e.*, the number of cell centers that could be sensed by the selected sensors $\mathcal{A}$. Because there are 2500 cell centers, $0 \leq u_n(\mathcal{A}) \leq 2500$, $\forall n \in \mathbb{N}^+, \forall \mathcal{A} \subseteq \mathcal{M}$. Let the desired sensing coverage to be $q = 2000$ cells. Notice that the utility function could be defined in other ways, and our framework can easily adapt to those definitions. Let the initially available resource on each sensor node be $e_m = 400, \forall m \in \mathcal{M}$. Let the selected sensors consume one unit resource per minute, whereas the unselected sensors do not consume any resource.

In Figure 5.3, we show the performances of the three algorithms with the aforementioned settings. In particular, in Figure 5.3a, we show the lifetime achieved by these algorithms. We see that our proposed algorithm MinPenalty performs the best among the three. In the "Static" scenario, it results in a lifetime 101% longer than that of Random and 43% longer than that of MaxRe; in the "Manhattan" scenario, it results in a lifetime 86% longer than that of Random and 35% longer than that of MaxRe. In Figure 5.3b, we show the average number of sensors selected for satisfying the coverage constraint in each time slot. We see that MinPenalty selects the smallest number of sensors on average, and therefore it is the most efficient.

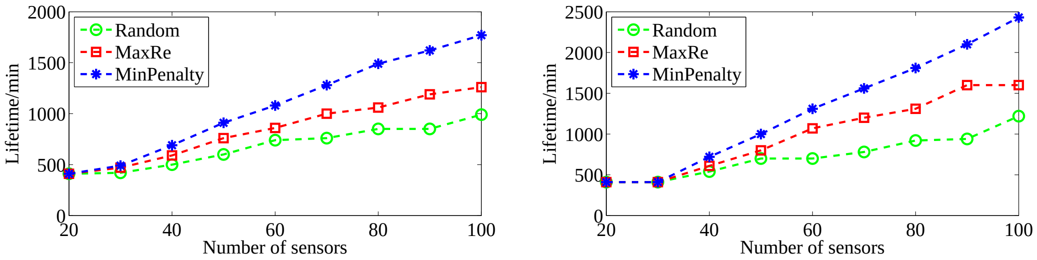
In Figure 5.4, we run simulations in both the scenarios "Static" and "Manhattan", and we vary the number of participating sensors M. We see that when M increases, the lifetime of the



(a) (b)

Figure 5.3: The performances of the three mentioned algorithms in both scenarios "Static" and "Manhattan". In Figure 5.3a, we show the lifetime incurred by these algorithms. In Figure 5.3b, we show the average number of sensors selected for satisfying the coverage constraint in each time slot. We see that MinPenalty is the most resource-efficient algorithm among the three.

sensing system increases because the total amount of available resources increases. Still, our proposed algorithm MinPenalty always performs the best among the three simulated algorithms.



(a) (b)

Figure 5.4: The performances of the three mentioned algorithms when we vary the number of sensors. In Figure 5.4a, we simulate the scenario "Static" and in Figure 5.4b, we simulate the scenario "Manhattan".

5.6.2 Practical Scenarios

We also consider two practical scenarios that are representative in participatory sensing systems.

- The first practical scenario is called **"TaxiSense"**. In this scenario, mobile sensors are installed on top of taxis for sensing certain environmental parameters. We use a real trajectory dataset of $M = 423$ taxis in San-Francisco in 3 weeks. The average speed of these taxis are 22.6 km/h in this dataset. The sensing field covers an area of 6km × 6km.
- The second practical scenario is called **"BusSense"**. In this scenario, mobile sensors are installed on top of public transportation vehicles in the city of Lausanne. We synthesize the mobility trajectories of $M = 327$ public transportation vehicles by using their timetable

and the GPS locations of the bus stops and metro stations. In this scenario, the sensing field covers an area of 9.6km × 6km.

In both scenarios, we partition the sensing field into grid cells and partition time into slots. Each sensor node has a sensing radius. The utility function of a selected subset of sensor nodes is still defined as the coverage, or more precisely, the number of cell centers that could be covered. We still assume that the selected sensors consume unit resource per minute and the unselected sensors do not consume any resource. The detailed settings are specified in Table 5.1.

Table 5.1: Parameters of the two practical scenarios

Parameters	TaxiSense	BusSense
Area of the sensing field	6km × 6km	9.6km × 6km
Size of a cell	120m × 120m	240m × 240m
Total number of cells	50 × 50	25 × 40
Sensing radius	600m	720m
Length of a time slot	10min	10min
Initially available resource	400	400
Desired coverage	1600	600

In Figure 5.5, we run simulations in both scenario “TaxiSense” and “BusSense” where we vary the desired coverage q. We see that when q increases, the lifetime of the sensing system decreases because more sensors have to be activated for sensing in each time slot.

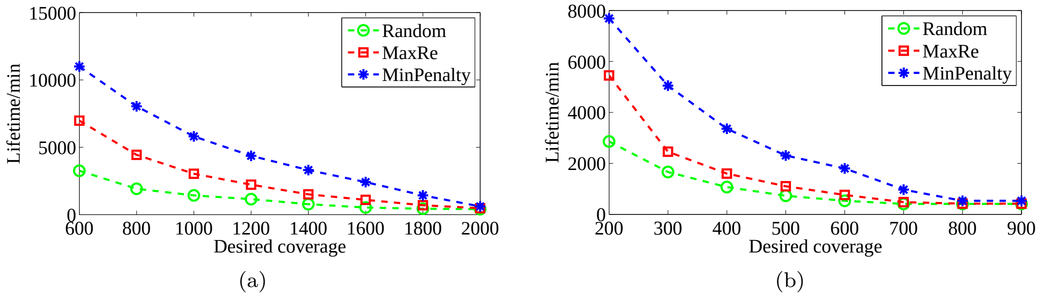


Figure 5.5: The performances of the three mentioned algorithms when we vary the desired coverage q. In Figure 5.5a, we simulate the scenario “TaxiSense” and in Figure 5.5b, we simulate the scenario “BusSense”.

5.7 Conclusions

In this chapter, we addressed the sparse sensor-selection problem by exploiting spatial correlations in both static WSNs and participatory sensing systems. In order to efficiently use the limited resources available to all sensor nodes, we proposed a generic framework that adaptively selects a subset of sensors for actively sensing. Using this framework, a certain level of application-specific utility is guaranteed at all time and the lifetime of the sensing system is prolonged. In static WSN scenarios, we rigorously showed that the proposed algorithms guarantee a lifetime at least $1/\log M$ (M is the number of sensors) of the optimal one. We ran extensive simulations in both static WSNs and participatory sensing systems and the results show the effectiveness of the proposed framework.

5.A Appendix

5.A.1 Proof of Lemma 5.1

Denote by F_{opt} the optimal objective value of problem (5.9) without the log-sum-exp approximation, where the objective value is $F(\boldsymbol{\theta}) = \max_{m\in\mathcal{M}} \theta_m/e_m$. We first use (5.8) to establish the relation between f_{opt} and F_{opt}

$$F_{\text{opt}} \leq f_{\text{opt}} \leq F_{\text{opt}} + \frac{\log M}{\alpha}. \tag{5.11}$$

Then, remember that F_{opt} is the inverse of the optimal lifetime, we have

$$\tau N_{\text{opt}} \leq \frac{1}{F_{\text{opt}}} \leq \tau(N_{\text{opt}}+1). \tag{5.12}$$

Plugging (5.12) into (5.11), we have $\frac{1}{\tau N_{\text{opt}}+\tau} \leq f_{\text{opt}} \leq \frac{1}{\tau N_{\text{opt}}} + \frac{\log M}{\alpha}$.

5.A.2 Proof of Theorem 5.1

First of all, we derive an upper-bound on the incremental change of the objective value $f(\boldsymbol{\theta}^{(n)}) - f(\boldsymbol{\theta}^{(n-1)})$. We take its Taylor expansion around $\boldsymbol{\theta}^{(n-1)}$ up to the second order,

$$\begin{aligned} f\left(\boldsymbol{\theta}^{(n)}\right) - f\left(\boldsymbol{\theta}^{(n-1)}\right) \leq& \boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right)\cdot\left(\boldsymbol{\theta}^{(n)}-\boldsymbol{\theta}^{(n-1)}\right) \\ &+\frac{M^2}{2}\cdot\left\|\boldsymbol{\Delta} f\left(\boldsymbol{\theta}^{(n-1)}\right)\right\|_\infty\cdot\left\|\boldsymbol{\theta}^{(n)}-\boldsymbol{\theta}^{(n-1)}\right\|_\infty^2, \end{aligned} \tag{5.13}$$

where $\boldsymbol{\nabla} f(\boldsymbol{\theta}^{(n-1)})$ and $\boldsymbol{\Delta} f(\boldsymbol{\theta}^{(n-1)})$ denote the gradient and the Hessian matrix of $f(\cdot)$ at the point $\boldsymbol{\theta}^{(n-1)}$, respectively.

The second-order term of the right-hand side of (5.13) is easy to upper bound. From (5.4), we deduce that $0 \leq \theta_m^{(n-1)} \leq 1$ and that

$$\theta_m^{(n)} - \theta_m^{(n-1)} = \frac{1}{n}\left(\mathbb{I}(m\in\mathcal{A}_n) - \theta_m^{(n-1)}\right). \tag{5.14}$$

It follows that $-1/n \leq \theta_m^{(n)} - \theta_m^{(n-1)} \leq 1/n, \forall m\in\mathcal{M}$, which is equivalent to $\left\|\boldsymbol{\theta}^{(n)}-\boldsymbol{\theta}^{(n-1)}\right\|_\infty \leq 1/n$. Also, a little calculation gives us $\left\|\boldsymbol{\Delta} f\left(\boldsymbol{\theta}^{(n-1)}\right)\right\|_\infty \leq \alpha/e_{\min}^2$. Therefore, the second-order term

$$\frac{M^2}{2}\cdot\left\|\boldsymbol{\Delta} f\left(\boldsymbol{\theta}^{(n-1)}\right)\right\|_\infty\cdot\left\|\boldsymbol{\theta}^{(n)}-\boldsymbol{\theta}^{(n-1)}\right\|_\infty^2 \leq \frac{\alpha M^2}{2n^2e_{\min}^2}. \tag{5.15}$$

It only remains to derive an upper-bound for the first-order term of the right-hand side of (5.13). We check from (5.5) and (5.8) that $\nabla_m f\left(\boldsymbol{\theta}^{(n-1)}\right) = \lambda_m^{(n-1)}$, and plug it into the first-order term

$$\begin{aligned}\boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right)\cdot\left(\boldsymbol{\theta}^{(n)}-\boldsymbol{\theta}^{(n-1)}\right) &= \sum_{m\in\mathcal{M}}\lambda_m^{(n-1)}\cdot\left(\theta_m^{(n)}-\theta_m^{(n-1)}\right)\\ &= \frac{1}{n}\sum_{m\in\mathcal{M}}\lambda_m^{(n-1)}\cdot\mathbb{I}\left(m\in\mathcal{A}_n\right)-\frac{1}{n}\sum_{m\in\mathcal{M}}\lambda_m^{(n-1)}\cdot\theta_m^{(n-1)}\\ &= \frac{1}{n}\sum_{m\in\mathcal{M}}\lambda_m^{(n-1)}\cdot\mathbb{I}(m\in\mathcal{A}_n)-\frac{1}{n}\boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right)\cdot\boldsymbol{\theta}^{(n-1)},\end{aligned}\tag{5.16}$$

where we use (5.14) on the second line. Remember that for any $n\in\mathbb{N}^+$, we use a β-approximation algorithm for solving the min-penalty set-cover problem (5.6), we guarantee

$$\sum_{m\in\mathcal{M}}\lambda_m^{(n-1)}\cdot\mathbb{I}(m\in\mathcal{A}_n) = \sum_{m\in\mathcal{A}_n}\lambda_m^{(n-1)} \le \beta f_s^{*(n)},\tag{5.17}$$

where $f_s^{*(n)}$ denotes the optimal objective value of (5.6). Because $f_s^{*(n)}$ is optimal, for any $\mathcal{A}\in\mathcal{M}(q)$,

$$f_s^{*(n)} \le \sum_{m\in\mathcal{A}}\lambda_m^{(n-1)} = \sum_{m\in\mathcal{M}}\lambda_m^{(n-1)}\cdot\mathbb{I}(m\in\mathcal{A}).\tag{5.18}$$

Let $\{p_{\mathcal{A}}^*\}_{\mathcal{A}\in\mathcal{M}(q)}$, $\boldsymbol{\theta}^*$ be the optimal solution of the problem (5.9) where $f(\boldsymbol{\theta}^*) = f_{\text{opt}}$ and the m-th element of $\boldsymbol{\theta}^*$ is $\theta_m^* = \sum_{\mathcal{A}\in\mathcal{M}(q)} p_{\mathcal{A}}^*\cdot\mathbb{I}(m\in\mathcal{A})$. Multiplying $p_{\mathcal{A}}^*$ on both sides of (5.18), summing it together for all $\mathcal{A}\in\mathcal{M}(q)$ and using $\sum_{\mathcal{A}\in\mathcal{M}(q)} p_{\mathcal{A}}^* = 1$,

$$\begin{aligned}\sum_{\mathcal{A}\in\mathcal{M}(q)} p_{\mathcal{A}}^* f_s^{*(n)} = f_s^{*(n)} &\le \sum_{\mathcal{A}\in\mathcal{M}(q)} p_{\mathcal{A}}^*\cdot\sum_{m\in\mathcal{M}}\lambda_m^{(n-1)}\cdot\mathbb{I}(m\in\mathcal{A})\\ &= \sum_{m\in\mathcal{M}}\lambda_m^{(n-1)}\cdot\sum_{\mathcal{A}\in\mathcal{M}(q)} p_{\mathcal{A}}^*\cdot\mathbb{I}(m\in\mathcal{A})\\ &= \sum_{m\in\mathcal{M}}\lambda_m^{(n-1)}\cdot\theta_m^* = \boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right)\cdot\boldsymbol{\theta}^*,\end{aligned}\tag{5.19}$$

where the first equality on the third line is because of the definition of θ_m^* in problem (5.9). Combining (5.16), (5.17) and (5.19), we have

$$\boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right)\cdot\left(\boldsymbol{\theta}^{(n)}-\boldsymbol{\theta}^{(n-1)}\right) \le \frac{1}{n}\boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right)\cdot\left(\beta\boldsymbol{\theta}^*-\boldsymbol{\theta}^{(n-1)}\right).\tag{5.20}$$

Plugging the upper-bounds (5.15) and (5.20) into (5.13),

$$\begin{aligned}f\left(\boldsymbol{\theta}^{(n)}\right)-f\left(\boldsymbol{\theta}^{(n-1)}\right) &\le \frac{1}{n}\boldsymbol{\nabla}^\top f\left(\boldsymbol{\theta}^{(n-1)}\right)\cdot\left(\beta\boldsymbol{\theta}^*-\boldsymbol{\theta}^{(n-1)}\right)+\frac{\alpha M^2}{2n^2e_{\text{min}}^2}\\ &\le \frac{1}{n}\left(\beta f(\boldsymbol{\theta}^*)-f\left(\boldsymbol{\theta}^{(n-1)}\right)\right)+\frac{\alpha M^2}{2n^2e_{\text{min}}^2},\end{aligned}\tag{5.21}$$

where the second inequality is because of the convexity of the objective function $f(\cdot)$.

Then, by multiplying n on both sides of (5.21) and by resorting,

$$n\left(f\left(\boldsymbol{\theta}^{(n)}\right)-\beta f(\boldsymbol{\theta}^*)\right)\leq(n-1)\left(f\left(\boldsymbol{\theta}^{(n-1)}\right)-\beta f(\boldsymbol{\theta}^*)\right)+\frac{\alpha M^2}{2ne_{\text{min}}^2}.$$

Summing it up from $n=1$ to N ($N\in\mathbb{N}^+$), dividing it by N and using $\sum_{n=1}^{N}1/n<\log N+1$, we have an upper-bound for $f(\boldsymbol{\theta}^{(N)})$,

$$f\left(\boldsymbol{\theta}^{(N)}\right)-\beta f(\boldsymbol{\theta}^*)<\frac{\alpha M^2(\log N+1)}{2e_{\text{min}}^2N}. \tag{5.22}$$

5.A.3 Proof of Theorem 5.2

Let the function $F(\boldsymbol{\theta})=\max_{m\in\mathcal{M}}\theta_m/e_m,\forall\boldsymbol{\theta}\in\mathbb{R}^M$. Remember from (5.7) that $N=N^*$ is the maximum number of time slots that satisfies $e_m^{(N)}\geq0,\forall m\in\mathcal{M}$. This implies that $\exists m\in\mathcal{M},e_m^{(N^*+1)}<0$. Using (5.4), we deduce that $\exists m\in\mathcal{M},\theta_m^{(N^*+1)}/e_m>1/(N^*+1)\tau$. Moreover, because $F(\boldsymbol{\theta}^{(N^*+1)})=\max_{m\in\mathcal{M}}\theta_m^{(N^*+1)}/e_m$,

$$F\left(\boldsymbol{\theta}^{(N^*+1)}\right)>\frac{1}{\tau(N^*+1)}. \tag{5.23}$$

We will then use (5.23) to derive a lower-bound for τN^*.

From (5.8), we have

$$F(\boldsymbol{\theta})\leq f(\boldsymbol{\theta})\leq F(\boldsymbol{\theta})+\frac{\log M}{\alpha},\forall\boldsymbol{\theta}\in\mathbb{R}^M. \tag{5.24}$$

Using (5.24), one can easily check that $f(\boldsymbol{\theta}^{(N^*+1)})\geq F(\boldsymbol{\theta}^{(N^*+1)})$ and that $f_{\text{opt}}\leq F_{\text{opt}}+\frac{\log M}{\alpha}$ where F_{opt} denotes the optimal objective value of problem (5.9) where the objective function $f(\cdot)$ is replaced by $F(\cdot)$. Plugging them into (5.10), we see that

$$F\left(\boldsymbol{\theta}^{(N^*+1)}\right)-\beta F_{\text{opt}}<\frac{\alpha M^2(\log(N^*+1)+1)}{2e_{\text{min}}^2(N^*+1)}+\frac{\beta\log M}{\alpha}. \tag{5.25}$$

Because active sensors have unit resource-consumption rates, each sensor can sustain at least $N^*\geq e_{\text{min}}/\tau$ time slots, which implies that $N^*+1>e_{\text{min}}/\tau$. Then, because $(\log N+1)/N$ is a diminishing term,

$$\frac{\log(N^*+1)+1}{N^*+1}<\frac{\tau}{e_{\text{min}}}\left(\log\frac{e_{\text{min}}}{\tau}+1\right). \tag{5.26}$$

Plugging (5.26) into (5.25),

$$F\left(\boldsymbol{\theta}^{(N^*+1)}\right)<\beta F_{\text{opt}}+\frac{\beta\log M}{\alpha}+\frac{\alpha\tau M^2}{2e_{\text{min}}^3}\left(\log\frac{e_{\text{min}}}{\tau}+1\right). \tag{5.27}$$

Using (5.23) and (5.27) together,

$$\tau(N^*+1)>\frac{1}{\beta F_{\text{opt}}+\frac{\beta\log M}{\alpha}+\frac{\alpha\tau M^2}{2e_{\text{min}}^3}\left(\log\frac{e_{\text{min}}}{\tau}+1\right)}. \tag{5.28}$$

Because F_{opt} is the inverse of the optimal lifetime, $F_{\text{opt}} \leq 1/(\tau N_{\text{opt}})$. Multiplying F_{opt} on both sides of (5.28) and using $F_{\text{opt}} \leq 1/(\tau N_{\text{opt}})$,

$$\frac{N^*}{N_{\text{opt}}} > \frac{1}{\beta + \frac{\beta \log M}{\alpha F_{\text{opt}}} + \frac{\alpha \tau M^2}{2e_{\min}^3 F_{\text{opt}}} \left(\log \frac{e_{\min}}{\tau} + 1\right)} - \tau F_{\text{opt}}. \tag{5.29}$$

To show explicitly the dependencies, we use $N^* = N^*(\tau, \alpha)$ and $N_{\text{opt}} = N_{\text{opt}}(\tau)$. Taking $\tau \to 0$ and then taking $\alpha \to \infty$ in (5.29), we have

$$\lim_{\alpha \to \infty} \lim_{\tau \to 0} \frac{N^*(\tau, \alpha)}{N_{\text{opt}}(\tau)} \geq \frac{1}{\beta}.$$

Plugging in $\beta = \log M$, we conclude the proof.

Chapter 6

Conclusion and Future Work

> We feel that even if all possible scientific questions be answered, the problems of life have still not been touched at all.
>
> *Ludwig Wittgenstein*

In this thesis, we discussed a set of adaptive selection problems in networked systems. Here, we review the results and raise some open problems in future work.

1. **Active Base-Station Selection:** we introduced a novel scheme for organizing WSNs, in which multiple BSs are deployed, but only one BS is adaptively selected to be active. By using the proposed scheme, we efficiently utilize the temporally and spatially varying energy resources available to all BSs. Therefore, the large batteries and energy harvesting devices of individual BSs can be substantially reduced. To adaptively choose the active BS, we proposed a simple yet powerful algorithm HEF. We proved its asymptotic optimality under mild conditions.

   In future work, we should consider the following problems:

   - We assumed that the energy recharge-rates of sensor nodes from solar panels are constant conditioned on the previous information. We might further relax this assumption in the analyses of the HEF algorithm.
   - We focused on the outdoor WSN scenarios where the energy recharged from solar panels is continuous. However, in many indoor WSN scenarios, the energy recharged from solar panels is intermittent. In these scenarios, our proposed scheme has to be adapted, so as to make full use of unstable energy resources.

2. **Joint Selection of Active Base-Stations and Routing:** we discussed the scheme of virtually moving multiple BSs in WSNs, where we adaptively re-elect an active subset of

BSs. This scheme not only achieves a high energy-efficiency but also avoids the difficulty of physically moving the BSs. We showed that the general problem of virtually moving BSs is in fact NP-hard and proposed an adaptive algorithm for solving it. Under mild conditions, the proposed algorithm guarantees to yield a lifetime at least 62% of the optimum.

In future work, we should consider the following problems:

- There might be some implementation issues when we apply the proposed scheme to a large-scale WSN. We have to consider how to initiate the WSN, how to maintain the network with multiple BSs, how to enrol new BSs, and how to deal with a BS failure.
- The idea of load balancing by virtually moving devices can be useful in other networked systems, including super-node selection in peer-to-peer networks and cooperative beamforming in cellular networks. We might adapt the proposed scheme and use it in the aforementioned scenarios.

3. **Sparse Sensor-Selection by Exploiting Temporal Correlations:** we proposed DASS, a novel scheme for sparse sampling that optimizes sparse-sampling patterns for precisely recovering temporal signals. This scheme adaptively learns the signal statistics, dynamically adjusts the sampling pattern, and recovers the signal from the limited amount of collected samples. We demonstrated the effectiveness of DASS through extensive experiments using two real-world meteorological datasets. The results show significant improvements over the state-of-the-art methods.

   In future work, we should consider the following problems:

   - We assumed that the signal to be sensed has a strong correlation among different time blocks. However, in many practical sensing scenarios, it is usually not clear how long each block is and whether there is a strong correlation across different blocks. Therefore, we might consider other temporal models of the signal, for example, an Auto Regressive Moving-Average (ARMA) model. In this example, the signal model has to be learnt in real time, rather than be learnt after a long time-block.
   - We did not consider the multi-hop communication costs for transmitting the sensing samples. In a real WSN, sensor nodes not only have to take the sensing samples, they also have to forward the sensing samples using the multi-hop communications. The problem will be more interesting if we incorporate the costs for routing the sensing samples.

4. **Sparse Sensor-Selection by Exploiting Spatial Correlations:** we discussed the sparse sensor-selection problem by exploiting the spatial correlation in both static WSNs and participatory sensing systems. We proposed a generic framework to adaptively select a subset of sensors for actively sensing. Using this framework, a certain level of application-specific utility is always guaranteed and the lifetime of the sensing system is prolonged.

   In future work, we should consider the following problems:

   - We only analyzed the performance of the proposed algorithms in the static WSN scenarios. We might also analyze it in participatory sensing systems, by assuming certain mobility models of the mobile sensors, *e.g.*, Markovian.
   - We might also consider exploiting simultaneously both temporal correlations and spatial correlations. In this scheme, we have to adaptively learn the spatio-temporal model of the signal to be sensed and select sparse sensors both in space and time.

Although we focused on the adaptive selection problems in WSNs and participatory sensing systems, we believe that the proposed algorithms have merits beyond those systems. For example, in smart grids, there is an increasing trend in integrating the intermittent renewable energy sources into the current grid. This poses many adaptive selection problems to maintain the stable operation and reliable control of the grid. We envisage that our proposed algorithms will be adapted to solve those problems.

Bibliography

[1] MeteoSwiss: The Federal Office of Meteorology and Climatology of Switzerland. http://www.meteoswiss.admin.ch.

[2] K. Aberer, S. Sathe, D. Chakraborty, A. Martinoli, G. Barrenetxea, B. Faltings, and L. Thiele, "Opensense: Open community driven sensing of environment," in Proc. *ACM SIGSPATIAL International Workshop on GeoStreaming*, IWGS '10, pp. 39–42, 2010.

[3] G. Barrenetxea, B. Berefull-Lozano, and M. Vetterli, "Lattice networks: capacity limits, optimal routing, and queueing behavior," *Networking, IEEE/ACM Transactions on*, vol. 14, no. 3, pp. 492–505, June 2006.

[4] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z. M. Wang, "Controlled sink mobility for prolonging wireless sensor networks lifetime," *Wirel. Netw.*, vol. 14, no. 6, pp. 831–858, Dec. 2008.

[5] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, "Power efficient monitoring management in sensor networks," in Proc. *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 4, pp. 2329–2334 Vol.4, March 2004.

[6] Y. Bi, L. Sun, J. Ma, N. Li, I. Khan, and C. Chen, "Hums: An autonomous moving strategy for mobile sinks in data-gathering sensor networks," *Eurasip Journal on Wireless Communications and Networking*, 2007.

[7] A. Bogdanov, E. Maneva, and S. Riesenfeld, "Power-aware base station positioning for sensor networks," in Proc. *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, pp. –585, 2004.

[8] B. Bollobás, D. Pritchard, T. Rothvoss, and A. Scott, "Cover-decomposition and polychromatic numbers," *SIAM Journal on Discrete Mathematics*, vol. 27, no. 1, pp. 240–256, 2013.

[9] A. Z. Broder, A. M. Frieze, and E. Upfal, "On the satisfiability and maximum satisfiability of random 3-cnf formulas," *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 322–330, 1993.

[10] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, 2008.

[11] E. Candès, "Compressive sampling," in Proc. *the International Congress of Mathematicians: invited lectures*, pp. 1433–1452, 2006.

[12] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[13] M. Cardei, M. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in Proc. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, March 2005.

[14] M. Cardei and D.-Z. Du, "Improving wireless sensor network lifetime through power aware organization," *Wirel. Netw.*, vol. 11, no. 3, pp. 333–340, May 2005.

[15] P. G. Casazza, M. Fickus, J. Kovačević, M. Leon, and J. Tremain, "A physical interpretation of tight frames," in *Harmonic analysis and applications.* Springer, 2006, pp. 51–76.

[16] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *Networking, IEEE/ACM Transactions on*, vol. 12, no. 4, pp. 609–619, Aug 2004.

[17] Z. Chen, J. Ranieri, R. Zhang, and M. Vetterli, "Dass: Distributed adaptive sparse sensing," *Wireless Communications, IEEE Transactions on*, vol. 14, no. 5, pp. 2571–2583, May 2015.

[18] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.

[19] A. Das and D. Kempe, "Algorithms for subset selection in linear regression," in Proc. *ACM Symposium on Theory of Computing (STOC)*, pp. 45–54, July 2009.

[20] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive Approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[21] M. F. Duarte, M. B. Wakin, D. Baron, and R. G. Baraniuk, "Universal distributed sensing via random projections," in Proc. *International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 177–185, Nashville, TN, Apr. 2006.

[22] A. El-Hoiydi and J.-D. Decotignie, "Wisemac: an ultra low power mac protocol for the downlink of infrastructure wireless sensor networks," in Proc. *Computers and Communications (ISCC), 2004, ninth International Symposium on*, vol. 1, pp. 244–251 Vol.1, June 2004.

[23] M. Fickus, D. G. Mixon, and M. J. Poteet, "Frame completions for optimally robust reconstruction," *arXiv:1107.1912v1*, July 2011.

[24] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[25] S. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy efficient schemes for wireless sensor networks with multiple mobile base stations," in Proc. *GLOBECOM '03. IEEE Global Telecommunications Conference*, pp. 377–381.

[26] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing-sync protocol for sensor networks," in Proc. *ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2003.

[27] N. Garg and J. Könemann, "Faster and simpler algorithms for multicommodity flow and other fractional packing problems," *SIAM J. Comput.*, vol. 37, no. 2, pp. 630–652, May 2007.

[28] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.).* Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[29] P. Gonzalez-Brevis, J. Gondzio, Y. Fan, H. Poor, J. Thompson, I. Krikidis, and P.-J. Chung, "Base station location optimization for minimal energy consumption in wireless networks," *IEEE Vehicular Technology Conference*, 2011.

[30] G. Grimmett and D. Stirzaker, *Probability and Random Processes, Third Edition.* Oxford University Press, 2001.

[31] C. Guestrin, A. Krause, and A. P. Singh, "Near-optimal sensor placements in gaussian processes," in Proc. *The 22rd International Conference on Machine Learning*, ICML '05, pp. 265–272, 2005.

[32] P. M. Hall, D. Marshall, and R. R. Martin, "Incremental eigenanalysis for classification," in Proc. *The British Machine Vision Conference*, pp. 286–295, 1998.

[33] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "Cartel: A distributed mobile sensor computing system," in Proc. *The Fourth International Conference on Embedded Networked Sensor Systems (Sensys)*, pp. 125–138, 2006.

[34] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope: Application-specific sensor network for environmental monitoring," *ACM Transactions on Sensor Networks*, 2010.

[35] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani, "Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP," *J. ACM*, vol. 50, no. 6, pp. 795–824, Nov. 2003.

[36] D. Johnson, "Routing in ad hoc networks of mobile hosts," in Proc. *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, Dec. 1994.

[37] J. Johnson, "Thermal agitation of electricity in conductors," *Physical Review*, vol. 32, no. 1, p. 97, 1928.

[38] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *Signal Processing, IEEE Transactions on*, vol. 57, no. 2, pp. 451–462, 2009.

[39] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory.* Prentice-Hall, Inc., Mar. 1993.

[40] C.-W. Ko, J. Lee, and M. Queyranne, "An exact algorithm for maximum entropy sampling," *Operations Research*, vol. 43, no. 4, pp. 684–691, 1995.

[41] A. Krause and C. Guestrin, "Optimizing sensing: From water to the web," *Computer*, vol. 42, no. 8, pp. 38–45, 2009.

[42] N. D. Lane, Y. Chon, L. Zhou, Y. Zhang, F. Li, D. Kim, G. Ding, F. Zhao, and H. Cha, "Piggyback CrowdSensing (PCS): Energy Efficient Crowdsourcing of Mobile Sensor Data by Exploiting Smartphone App Opportunities," in Proc. *The 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pp. 7:1–7:14, 2013.

[43] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in Proc. *The ACM International Conference on Mobile Computing and Networking (MobiCom)*, pp. 145–156. ACM, 2009.

[44] J. Luo and J.-P. Hubaux, "Joint sink mobility and routing to maximize the lifetime of wireless sensor networks: The case of constrained mobility," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 871–884, June 2010.

[45] J. Luo and J. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," *2005. 24th Annual Joint Conference of the INFOCOM*, 2005.

[46] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions i," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[47] H. N. Pham, D. Pediaditakis, and A. Boulis, "From Simulation to Real Deployments in WSN and Back," in Proc. *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pp. 1–6, 2007.

[48] G. Quer, R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi, "Sensing, Compression, and Recovery for WSNs: Sparse Signal Modeling and Monitoring Framework," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 10, pp. 3447–3461, October 2012.

[49] T. Raiko, A. Ilin, and J. Karhunen, "Principal component analysis for sparse high-dimensional data," in Proc. *International Conference on Neural Information Processing (ICONIP)*, pp. 566–575. Springer, 2008.

[50] R. Rana, C. Chou, S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: An end-to-end participatory urban noise mapping system," in Proc. *The 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN '10*, pp. 105–116, 2010.

[51] J. Ranieri, A. Chebira, and M. Vetterli, "Near-Optimal Sensor Placement for Linear Inverse Problems," *Signal Processing, IEEE Transactions on*, vol. 62, no. 5, pp. 1135–1146, 2014.

[52] C. M. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," in Proc. *ACM International Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2006.

[53] T. Schoellhammer, B. Greenstein, E. Osterweil, M. Wimbrow, and D. Estrin, "Lightweight temporal compression of microclimate datasets," in Proc. *The 29th Annual IEEE International Conference on Local Computer Networks*, pp. 516–524, 2004.

[54] X. Sheng, J. Tang, and W. Zhang, "Energy-efficient collaborative sensing with mobile phones," in Proc. *the IEEE Conference on Computer Communications (INFOCOM)*, pp. 1916–1924, 2012.

[55] Y. Shi and Y. Hou, "Some fundamental results on base station movement problem for wireless sensor networks," *IEEE/ACM Transactions on Networking*, 2011.

[56] Y. Tanaka, S. Imahori, M. Sasaki, and M. Yagiura, "An lp-based heuristic algorithm for the node capacitated in-tree packing problem," *Computers and Operations Research*, vol. 39, no. 3, pp. 637–646, 2012.

[57] N. Thepvilojanapong, S. Konomi, Y. Tobe, Y. Ohta, M. Iwai, and K. Sezaki, "Opportunistic collaboration in participatory sensing environments," in Proc. *The fifth ACM international workshop on Mobility in the evolving internet architecture*, MobiArch '10, pp. 39–44, 2010.

[58] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in Proc. *The 1st international conference on Embedded networked sensor systems*, SenSys '03, pp. 171–180, 2003.

[59] Z. Vincze, R. Vida, and A. Vidacs, "Deploying multiple sinks in multi-hop wireless sensor networks," in Proc. *Pervasive Services, IEEE International Conference on*, pp. 55 –63, july 2007.

[60] K. Viswanatha, S. Ramaswamy, A. Saxena, and K. Rose, "Error-resilient and complexity-constrained distributed coding for large scale sensor networks," in Proc. *The ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 293–304. ACM, 2012.

[61] L. Wang, D. Zhang, and H. Xiong, "EffSense: Energy-efficient and Cost-effective Data Uploading in Mobile Crowdsensing," in Proc. *The 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, UbiComp '13 Adjunct, pp. 1075–1086, 2013.

[62] W. Wang, M. Garofalakis, and K. Ramchandran, "Distributed sparse random projections for refinable approximation," in Proc. *The ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 331–339, 2007.

[63] H. Weinschrott, F. Durr, and K. Rothermel, "Streamshaper: Coordination algorithms for participatory mobile urban sensing," in Proc. *Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on*, pp. 195–204, 2010.

[64] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, "Deploying a wireless sensor network on an active volcano," *Internet Computing, IEEE*, vol. 10, no. 2, pp. 18–25, March 2006.

[65] B. Widrow and I. Kollár, *Quantization noise.* Cambridge University Press, 2008.

[66] X. Wu and M. Liu, "In-situ soil moisture sensing: measurement scheduling and estimation using compressive sensing," in Proc. *The ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN).* ACM, 2012.

[67] Z. Yan, J. Eberle, and K. Aberer, "Optimos: Optimal sensing for mobile sensors," in Proc. *Mobile Data Management (MDM), 2012 IEEE 13th International Conference on*, pp. 105–114, July 2012.

[68] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in Proc. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1567–1576 vol.3, 2002.

[69] R. Zhang, F. Ingelrest, G. Barrenetxea, P. Thiran, and M. Vetterli, “The beauty of the commons: Optimal load sharing by base station hopping in wireless sensor networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 33, no. 8, pp. 1480–1491, Aug 2015.

[70] R. Zhang, P. Thiran, and M. Vetterli, “Virtually moving base stations for energy efficiency in wireless sensor networks,” in Proc. *The 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc ’15, pp. 357–366, 2015.

[71] D. Zordan, B. Martinez, I. Vilajosana, and M. Rossi, “On the performance of lossy compression schemes for energy constrained sensor networking,” *ACM Transactions on Sensor Networks*, vol. 11, no. 1, pp. 15:1–15:34, Aug. 2014.

[72] G. Zussman and A. Segall, “Energy efficient routing in ad hoc disaster recovery networks,” in Proc. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 1, pp. 682–691 vol.1, March 2003.

Curiculum Vitæ

Runwei Zhang

Audiovisual Communications Laboratory (LCAV)
École Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland

Email: `runwei.zhang@epfl.ch`
Web-page: `http://lcav.epfl.ch/people/runwei.zhang`

Personal

Date of Birth	July 21st, 1988
Marital Status	Single

Education

Sep. 2010–present	Ph.D. candidate in School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
Sep. 2006–Aug. 2010	B.Sc. in Automation, Tsinghua University, Beijing, China • Honored undergraduate student • Got schneider electric scholarship in 2009 • Got first-rank chinese national scholarship in 2007 and 2008
Jun. 2009–Jul. 2009	Research intern. in Engineering School of Viterbi, University of Southern California (USC), Los Angeles, United States

Research Interests

Adaptive Algorithms, Network Optimization and Signal Processing

Publications

Conference papers

1. **R. Zhang**, P. Thiran, M. Vetterli, *Virtually moving base stations for energy efficiency in wireless sensor networks*, accepted to the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing (**MobiHoc**), 2015

Journal papers

1. **R. Zhang**, F. Ingelrest, G. Barrenetxea, P. Thiran, M. Vetterli, *The beauty of the commons: optimal load sharing by base station hopping in wireless sensor networks*, accepted in IEEE Journal on Selected Areas in Communications (**JSAC**), 2015
2. Z. Chen, J. Ranieri, **R. Zhang** and M. Vetterli, *DASS: Distributed adaptive sparse sensing*, accepted in IEEE Transactions on Wireless Communications (**TWC**), 2015.

Skills

Mathematics	Convex Optimization, Signal Processing, Probability and Statistics
Programming	C/C++, Matlab, Java/Android, Python, JavaScript, Bootstrap, Jquery, Nodejs, Linux Shell Scripting, SQL, PHP, HTML/CSS, Latex, C#, Nes-C • Got the ninth place at Helvetic Coding Contest in Switzerland, 2012 • Got the best runners-up at Facebook Hackathon in Switzerland, 2015
Language	Chinese(Native), English(Advanced C1-C2), French(Intermediate, A2-B1)
Presentations	Gave presentations in top-tiered international conferences and received positive feedbacks

Other Activities

1. President of Chinese Students and Scholar Association (CSSA) in Lausanne
   Organize spectacles, seminars and sport matches to enrich the life of the local community and to promote cultural exchanges between Switzerland and China
2. Sports
   Basketball, Jogging, Hiking, Skiing