Boston University

OpenBU **http://open.bu.edu**

Theses & Dissertations Boston University Theses & Dissertations

2017

Control and optimization approaches for energy-limited systems: applications to wireless sensor networks and battery-powered vehicles

https://hdl.handle.net/2144/20849

Boston University

BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

CONTROL AND OPTIMIZATION APPROACHES FOR ENERGY-LIMITED SYSTEMS: APPLICATIONS TO WIRELESS SENSOR NETWORKS AND BATTERY-POWERED VEHICLES

by

SEPIDEH POURAZARM

B.S., K. N. Toosi University of Technology, Tehran, Iran, 2004
M.S., K. N. Toosi University of Technology, Tehran, Iran, 2007

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

2017

© 2017 by
SEPIDEH POURAZARM
All rights reserved

Approved by

First Reader

Christos G. Cassandras, PhD
Professor of Electrical and Computer Engineering
Professor of Systems Engineering

Second Reader

Ioannis Ch. Paschalidis, PhD
Professor of Electrical and Computer Engineering
Professor of Systems Engineering
Professor of Biomedical Engineering

Third Reader

Michael Caramanis, PhD
Professor of Mechanical Engineering
Professor of Systems Engineering

Fourth Reader

Pirooz Vakili, PhD
Associate Professor of Mechanical Engineering
Associate Professor of Systems Engineering

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Christos G. Cassandras, without his trust and tremendous support I would have never been able to reach where I am right now. I would like to thank Prof. Cassandras for his unwavering support, immense knowledge and patience as an advisor and mentor during my PhD career. This research would not have been possible without the thoughtful advice and mentorship of my advisor.

It was an honor for me to have Professor Paschalidis, Professor Vakili and Professor Caramanis on my dissertation committee. I would like to thank them each for their time, constructive feedback, and valuable ideas and questions. I also thank Professor Olshevsky for accepting to serve as the chair for my PhD defense.

I would like to thank my hardworking collaborator, Jing Zhang, for his collaboration in developing chapter 6 of this dissertation. I am also thankful to my current and past fellow lab researchers, Yasaman, Julia, Xinmiao, Yue, Nan, Rui, and Rebecca for their friendship and support during this journey. I would like to thank the rest of my friends at BU, specially, Setareh, Soudeh, Armin, Mohammad, Morteza, Elli, Eran, and Selin. It has been a pleasure to share this experience with them.

Last but not least, I would like to thank my dear family. My parents, my sister and my brother made this journey possible with their unconditional love and support.

Finally, but most importantly, I wish to thank Saber for his patience, love, and faith in me. I could not have completed this journey without Saber by my side. Thank you with all my heart and soul. I love you and am forever indebted to you for giving me your love and your heart.

CONTROL AND OPTIMIZATION APPROACHES FOR ENERGY-LIMITED SYSTEMS: APPLICATIONS TO WIRELESS SENSOR NETWORKS AND BATTERY-POWERED VEHICLES

SEPIDEH POURAZARM

Boston University, College of Engineering, 2017

Major Professor: Christos G. Cassandras, PhD
Professor of Electrical and Computer Engineering
Professor of Systems Engineering

ABSTRACT

This dissertation studies control and optimization approaches to obtain energy-efficient and reliable routing schemes for battery-powered systems in network settings.

First, incorporating a non-ideal battery model, the lifetime maximization problem for static wireless sensor networks is investigated. Adopting an optimal control approach, it is shown that there exists a time-invariant optimal routing vector in a fixed topology network. Furthermore, under very mild conditions, this optimal policy is robust with respect to the battery model used. Then, the lifetime maximization problem is investigated for networks with a mobile source node. Redefining the network lifetime, two versions of the problem are studied: when there exist no prior knowledge about the source node's motion dynamics vs. when source node's trajectory is known in advance. For both cases, problems are formulated in the optimal control framework. For the former, the solution can be reduced to a sequence of nonlinear programming problems solved on line as the source node trajectory evolves. For the latter, an explicit off-line numerical solution is required.

Second, the problem of routing vehicles with limited energy through a network with inhomogeneous charging nodes is studied. The goal is to minimize the total elapsed time, including traveling and recharging time, for vehicles to reach their destinations. Adopting a game-theoretic approach, the problem is investigated from two different points of view: user-centric vs. system-centric. The former is first formulated as a mixed integer nonlinear programming problem. Then, by exploiting properties of an optimal solution, it is reduced to a lower dimensionality problem. For the latter, grouping vehicles into subflows and including the traffic congestion effects, a system-wide optimization problem is defined. Both problems are studied in a dynamic programming framework as well.

Finally, the thesis quantifies the Price Of Anarchy (POA) in transportation networks using actual traffic data. The goal is to compare the network performance under user-optimal vs. system-optimal policies. First, user equilibria flows and origin-destination demands are estimated for the Eastern Massachusetts transportation network using speed and capacity datasets. Then, obtaining socially-optimal flows by solving a system-centric problem, the POA is estimated.

Contents

1 Introduction **1**

1.1 Routing in Wireless Sensor Networks 2

1.1.1 Non-Ideal Battery Dynamics 3

1.1.2 Lifetime Maximization for Static Wireless Sensor Networks . . 5

1.1.3 Lifetime Maximization of Wireless Sensor Networks with a Mobile Source Node 8

1.2 Optimal Routing for Battery-Powered Vehicles 10

1.2.1 Scope of work 12

1.3 The Price of Anarchy in Transportation Networks 13

1.3.1 Scope of work 14

1.4 Analytical Tools 15

1.4.1 Optimal control approach and optimization methods 15

1.5 Contributions of This Work 19

1.5.1 Optimal routing of static wireless sensor networks 19

1.5.2 Optimal routing of wireless sensor networks with a mobile source node 21

1.5.3 Optimal routing of energy-limited vehicles 21

1.5.4 Price of Anarchy in transportation networks using real traffic data 23

1.6 Thesis Outline 24

2 Lifetime Maximization for Static Wireless Sensor Networks **26**

2.1 Introduction 26
2.2 Optimal control problem formulation 31
  2.2.1 Network model 31
  2.2.2 Non-ideal battery dynamics 32
  2.2.3 Energy consumption model 35
  2.2.4 Optimal control problem formulation 37
2.3 Optimal control problem solution 38
  2.3.1 Analysis of scenario S_i 39
  2.3.2 Algorithm for solving the optimal control problem 45
  2.3.3 A robustness property of the optimal routing policy 48
  2.3.4 Simulation examples 50
2.4 A joint optimal routing and initial energy allocation problem 53
  2.4.1 Simulation examples 59
2.5 Lifetime maximization problem under a more general nonlinear battery model 60
  2.5.1 Optimal control problem formulation 64
  2.5.2 Optimal control problem solution 65
  2.5.3 Analysis of scenario S_i 66
  2.5.4 A Robustness Property of the Optimal Routing Policy 69
  2.5.5 Optimal routing by solving a single NLP 71
  2.5.6 Simulation examples 72
  2.5.7 Joint optimal routing and initial energy allocation 75
  2.5.8 Simulation examples 78
2.6 Network Performance Under Security Threats 79
2.7 Summary 85

3 Lifetime Maximization for Wireless Sensor Networks with a Mobile Source Node 86

3.1 Introduction 86

3.2 Network model 88

3.3 Energy Consumption Model 90

3.4 Optimal control problem formulation 90

3.4.1 Optimal Control Problem - I 91

3.4.2 Optimal Control Problem - II 96

3.4.3 Optimal Control Problem - III 99

3.4.4 Numerical examples 104

3.5 Optimal Control Formulation when source node trajectory is known in advance 108

3.5.1 Numerical Examples 111

3.6 Summary 118

4 Optimal Routing and Charging of Energy-Limited Vehicles in Traffic Networks 119

4.1 Introduction 119

4.2 Single Vehicle Routing 123

4.2.1 Properties 125

4.2.2 Determination of optimal recharging amounts r_i^* 131

4.3 Multiple Vehicle Routing 132

4.3.1 Properties 135

4.3.2 Flow control formulation 138

4.3.3 Objective function selection 141

4.3.4 Numerical Examples 143

4.4 Selection of the Number of Subflows 151

4.4.1 Numerical Example 154

4.5 Multiple-Vehicle Routing Problem in the Presence of Non-Electric Vehicle Flows 158
4.6 Routing of energy-aware vehicles in networks with inhomogeneous charging nodes 160
    4.6.1 Single Vehicle Routing 161
    4.6.2 Multiple Vehicle Routing 167
    4.6.3 Properties 168
    4.6.4 Flow control formulation 171
    4.6.5 Objective function selection using actual traffic data 173
    4.6.6 Numerical examples for the Eastern Massachusetts transportation network. 177
4.7 Summary 184

5 Optimal Routing of Electric Vehicles in Networks with Charging Nodes: A Dynamic Programming Approach 185
5.1 Introduction 185
5.2 Single Vehicle Routing 186
    5.2.1 Dynamic Programming Formulation 187
    5.2.2 Numerical Example 189
5.3 Multiple Vehicle Routing 190
    5.3.1 Dynamic Programming Formulation 191
    5.3.2 Numerical Examples 196
5.4 Conclusions and future work 197

6 The Price of Anarchy in Transportation Networks Using Actual Traffic Data 199
6.1 Introduction 199
6.2 Models and Methods 201

6.2.1 Transportation network model 201
6.2.2 Selfish routing 202
6.2.3 Socially optimal routing 203
6.2.4 Price of Anarchy 203
6.3 Data Set Description 204
6.3.1 Speed dataset description 204
6.3.2 Capacity dataset description 204
6.3.3 Matching capacity data with speed data 206
6.4 Data Processing 206
6.4.1 Preprocessing 206
6.4.2 Estimating the O-D demand matrix 211
6.4.3 Estimating cost functions 212
6.5 Numerical Results 213
6.6 Summary 217

7 Conclusions and Future Directions 218

7.1 Future Directions 222
7.1.1 Extensions for lifetime maximization problem for static wireless sensor networks 222
7.1.2 Extensions for lifetime maximization problem for wireless sensor networks with mobile source nodes 222
7.1.3 Extensions to Optimal Routing of Energy-limited Vehicles . . 223
7.1.4 Extensions to data-driven estimation of Origin-Destination demand matrices 224

References 226

Curriculum Vitae 234

List of Tables

2.1 Optimal routing probs., 7-node network, ideal batteries 51

2.2 Lifetimes under routing policy given in Table 2.1 52

2.3 Optimal routing probs., 7-node network, *non-ideal* batteries ($k = 0.001$) 53

2.4 Lifetimes under routing policy given in Table 2.3 and $k = 0.001$ . . . 53

2.5 Optimal routing probabilities and network lifetime for a 7-node network (Fig.2·2) 54

2.6 Lifetimes under routing policy given in Table 2.5 54

2.7 Optimal routing probabilities and network lifetime for a 7-node network with different initial energies 54

2.8 Lifetimes under routing policy given by Table 2.7 55

2.9 Optimal routing probabilities, initial battery energy and network lifetime for a 7-node network 60

2.10 Optimal routing probabilities and network lifetime for a 7-node network with different diffusion coefficients 74

2.11 Node Lifetimes under $w^*(0)$ when $\delta_m = (0.273)^2 m^2$ 74

2.12 CPU time under different battery dynamics using Single NLP formulation compared to **Algorithm A1** 75

2.13 Optimal routing probs., 7-node network, *non-ideal* batteries 80

2.14 Optimal routing probs., 7-node network, ideal batteries 81

3.1 Network lifetime using OCP-II for different values of ϵ 108

4.1 d_{ij} values for network of Fig. 4·1 (*miles*) 143

4.2 Numerical results for sample problem 144
4.3 CPU time for sample problems 148
4.4 Numerical results for different values of eg for network of Fig. 4·1 . . 148
4.5 Traveling time on each link for the network shown in Fig. 4·1 under system-optimal flows 149
4.6 Normalized Nash-equilibrium flows 150
4.7 Optimal Normalized flow on each link (x_{ij}) obtained by solving N-NLP problem 155
4.8 Estimates for the MINLP solution for different values of N 156
4.9 Critical number of subflows, N^*, for different values of δ 157
4.10 Classification of charging stations (Bai et al., 2010) 161
4.11 Uncontrolled NEV flow on each link during AM period [No. veh/hr] . 178
4.12 Distance [mile] of all links in the sub-network in Fig. 4·6 178
4.13 Numerical results for sample problem 180
4.14 Effect of flow rate, R, on Optimal routes 182
4.15 CPU time for sample problem 183

5.1 Subflow-level graph size for different number of subflows for road network shown in Fig. 5·1 192
5.2 Numerical results for sample problem 197

6.1 Traveling time (hr) on each link under system-optimal flows. 217

List of Figures

1·1 Battery operation: a) Charged battery; b) Before recovery; c) After recovery d) Discharged battery 4

2·1 Kinetic Battery Model including recharging 34
2·2 Network topology-1 52
2·3 Network topology 74
2·4 Network topology 79
2·5 Routing probability updates under EAR policy 82
2·6 Routing probability updates under EAR policy when node 2 is under attack 83
2·7 Normalized throughput vs prob. of broadcasting faked-cost 84

3·1 6-node network with mobile source (node 1) 105
3·2 (a) Routing vector ; (b) Residual energies over time during the network lifetime (Problem II) 106
3·3 (a) Routing vector ; (b) Residual energies over time during the network lifetime (Problem III) 107
3·4 5-node network with mobile source 111
3·5 (a) Residual energies over time during the network lifetime; (b) Optimal routing vector 112
3·6 (a) Residual energies over time during the network lifetime; (b) Routing vector under greedy policy 114
3·7 5-node network with mobile source 115

3·8 (a) Residual energies over time during the network lifetime; (b) Optimal routing vector for the sinusoidal trajectory 116
3·9 (a) Residual energies over time during the network lifetime; (b) Routing vector under the greedy policy for the sinusoidal trajectory 117
3·10 (a) Residual energies over time (Problem II); (b) Residual energies over time (Problem III) 118

4·1 A 7-node network example for routing with recharging nodes. 125
4·2 Performance as a function of N (No. of subflows) 145
4·3 A 13-node network example for routing with recharging nodes. 147
4·4 Average deviation between the solution of the NLP and estimated solution of the MINLP problem for different values of N 157
4·5 Road map of Eastern Massachusetts 174
4·6 (a) An interstate highway sub-network of Eastern Massachusetts (the blue numbers indicate node indices); (b) The topology of the sub-network175
4·7 Comparison of the estimated cost functions corresponding to different time periods. 176
4·8 Performance as a function of N (No. of subflows) 179

5·1 A 49-node grid network with inhomogeneous charging nodes. 189
5·2 A 7-node road network with inhomogeneous charging nodes. 191
5·3 Subflow-level graph showing all feasible combination of nodes via which subflows may travel 192

6·1 All available road segments in the road map of Eastern Massachusetts. 205
6·2 Sample lines from speed dataset 205

6·3 (a) An interstate highway sub-network of Eastern Massachusetts (a sub-map of Fig. 6·1; the blue numbers indicate node indices); (b) The topology of the sub-network (the numbers beside arrows are link indices, and the numbers inside ellipses are node indices). 207

6·4 Relationships between speed, density, and flow based on Greenshield's traffic flow model 210

6·5 Comparison of the estimated cost functions corresponding to different time periods. 214

6·6 (a) Social-opt. (green) and user-opt. (red) flows on links 0 to 11; (b) Social-opt. (green) and user-opt. (red) flows on links 12 to 23. 215

6·7 POA for PM period (5–7 pm) in April based on avg. flow on each link. 216

List of Abbreviations

BEV	......	Battery Electric Vehicle
BPS	......	Battery-Powered System
BPV	......	Battery-Powered Vehicle
DoS	......	Denial of Service
DP	......	Dynamic Programming
EAR	......	Energy Aware Routing
EV	......	Electric Vehicle
E-VRPTW	......	Electric Vehicle Routing Problem with Time Windows
GLS	......	Generalized Least Squares
IP	......	Integer Programming
KBM	......	Kinetic Battery Model
LDV	......	Light Duty Vehicle
LP	......	Linear Programming
LSP	......	Least Squares Problem
MINLP	......	Mixed Integer Non-Linear Programming
MPO	......	Metropolitan Planning Organization
NEV	......	Non-Electric Vehicle
NLP	......	Non-Linear Programming
OCP	......	Optimal Control Problem
O-D	......	Origin-Destination
PHEV	......	Plug-in Hybrid Electric Vehicle
PMP	......	Pontryagin Maximum Principle
POA	......	Price Of Anarchy
RBS	......	Regenerative Braking System
TPBVP	......	Two Point Boundary Value Problem
TSP	......	Traveling Salesman Problem
UAV	......	Unmanned Autonomous Vehicle
WSN	......	Wireless Sensor Networks

Chapter 1

Introduction

The increasing presence of mobile and wireless battery-powered systems (BPS) has given rise to novel issues in classical network routing problems (Laporte, 1992). Due to limited accessibility to charging resources and limited battery lifetime, the power consumption is a key issue in BPS. In this thesis we study the routing problem for battery-powered energy-aware systems with applications in wireless sensor networks and Battery Power Vehicles (BPVs). In WSNs we deal with a routing problem in which nodes are energy-limited with no recharging capabilities. On the other hand, for routing of BPVs we deal with a routing problem in which entities (vehicles) are energy-limited with rechargeability. Thus, in both applications we face energy constraints in the network routing problem but in different settings.

In wireless sensor networks, nodes are mainly battery powered with sensing, processing and communicating capabilities. For most applications of interest, e.g., exploration, nodes are hardly accessible and they do not have rechargeablility. Thus, power consumption is crucial in WSNs, since it directly impacts their lifetime. In this thesis we propose algorithms determining routing schemes to optimize properly defined performance metrics which reflect the limited energy resources of WSNs.

Unlike WSNs, BPVs have recharging capability and their energy constraint has direct impact on the routing decision. Motivated by the significant role of recharging in BPVs, we study the routing problem for vehicles with limited energy through a network with at least some charging nodes.

The structure of this chapter is as follows. First, we review the lifetime maximization problem for wireless sensor networks. We define our research scope and review this problem in the literature. Then, we briefly introduce the routing problem for energy-aware battery-powered vehicles. Reviewing the literature, we define problems we will focus on. We also discuss how to use actual traffic data to investigate the transportation network performance under user-centric vs system-centric strategies. We then discuss the methodologies we are using to solve these problems and point out contributions of this thesis.

1.1 Routing in Wireless Sensor Networks

A Wireless Sensor Network (WSN) is formed by small autonomous nodes communicating over wireless links. Nodes have sensing, processing and communicating capabilities. They are mainly battery powered and tightly constrained in terms of energy, processing, and storage capacities, therefore requiring careful resource management (Al-Karaki and Kamal, 2004). Applications of such networks include exploration, surveillance, and environmental monitoring.

Since the majority of power consumption is due to the radio component (Shnayder et al., 2004), nodes rely on short-range communication and form a multi-hop network to deliver information to a base station. Because nodes are battery-powered, it is crucial to adopt a realistic battery model for nodes in order to maximize the network lifetime. Under the assumption that an electrochemical battery cell is "ideal", a constant voltage throughout the discharge process and a constant capacity for all discharge profiles are both maintained over time. However, battery discharge behavior is sensitive to several factors including the discharge rate, temperature, and the number of charge/discharge cycles (Rao et al., 2003). Therefore, battery discharge behavior deviates significantly from the behavior of an ideal battery. In this thesis, we relax the common assumption of ideal battery dynamics for nodes in the literature (e.g., (Chang and Tassiulas, 2004), (Wu and Cassandras, 2005),(Shah and Rabaey, 2002)) and revisit the lifetime maximization problem of WSNs by adopting a realistic non-ideal battery dynamics for nodes.

1.1.1 Non-Ideal Battery Dynamics

There are two important effects (Rao et al., 2003) that make battery performance nonlinear and sensitive to the discharge profile: (i) the Rate Capacity effect (Doyle and Newman, 1997), and (ii) the Recovery effect(Martin, 1999). The former leads to the loss of capacity with increasing load current, and the latter makes the battery regain portions of its capacity after a resting time.

Fig. 1·1 shows a simplified symmetric electrochemical cell to illustrate this phenomenon. We assume the distance between electrodes (anode and cathode) is 2ω. During a rest time, the electrolyte concentration is constant over the length of ω (Fig. 1·1a). When the cell is connected to a load, due to the electro-chemical reaction, the concentration of the electrolyte is reduced near the electrode and creates a gradient (Fig. 1·1b) which causes the diffusion of species towards the electrode. Then, during an idle period, this diffusion makes the electrolyte concentration gradually become uniform over the length ω showing the battery recovery effect (Fig. 1·1c). Finally, when the electrolyte concentration drops to a predetermined cutoff level, the battery is said to be depleted while it has some unused capacity. This phenomenon describes the rate capacity effect (Fig.1·1d) Therefore, the voltage as well as energy amount delivered by a battery heavily rest on the discharge profile. Thus, when dealing with energy optimization, it is necessary to take this into account along with nonlinear variations in a battery's capacity.

As a result, there are several proposed models to describe a non-ideal battery; a detailed overview is given in (Jongerden and Haverkort, 2008). Accordingly, models

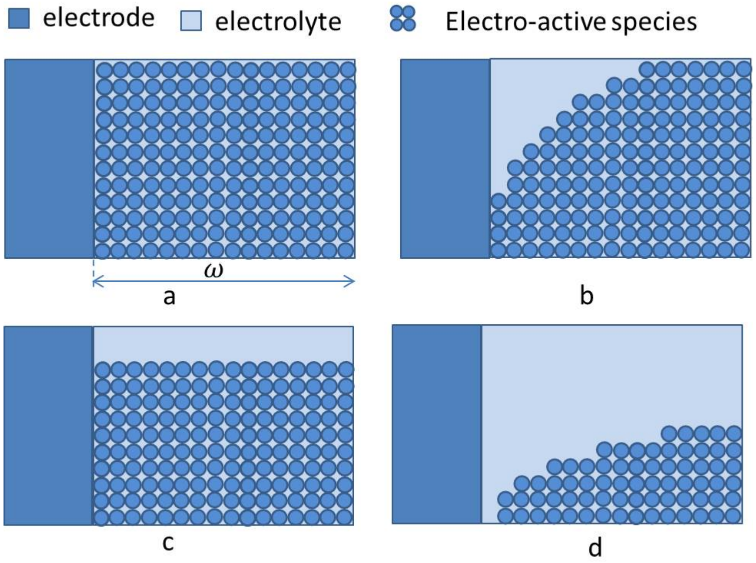


Figure 1·1: Battery operation: a) Charged battery; b) Before recovery; c) After recovery d) Discharged battery

are broadly classified as: electrochemical (Fuller et al., 1993; Doyle and Newman, 1997; Newman, 1998), circuit-based (Hageman, 1993; Chen and Rincon-Mora, 2006), stochastic (Chiasserini and Rao, 1999b; Chiasserini and Rao, 1999a; Chiasserini and Rao, 2001; Rao et al., 2005), and analytical (Rakhmatov and Vrudhula, 2001; Vrudhula and Rakhmatov, 2003; Manwell and McGowan, 1993). Electrochemical models possess the highest accuracy, but their complexity makes them impractical for most real-time applications. Electrical-circuit models are much simpler and therefore computationally less expensive but their accuracy leads to errors which may be reduced at the expense of added complexity (Chen and Rincon-Mora, 2006). Stochastic models use a discrete time Markov chain with $N + 1$ states to represent the number of charge units available in the battery. Since N is large, these models are also limited by high computational requirements. Last but not least, analytical models, including diffusion-based models (Vrudhula and Rakhmatov, 2003; Zhang and Shi, 2009; Barbarisi et al., 2006) and the Kinetic Battery Model (KBM) (Manwell and Mc-

Gowan, 1994; Rao et al., 2005), use only a few equations to capture the battery's main features, i.e., the rate capacity effect and the recovery effect. They also provide acceptable accuracy with respect to battery lifetime estimation. Thus, in this thesis, we use analytical models in order to capture the nonlinear dynamics of the battery for each node.

1.1.2 Lifetime Maximization for Static Wireless Sensor Networks

Routing schemes in WSNs aim to deliver data from the data sources (nodes with sensing capabilities) to a data sink (typically, a base station) in an energy-efficient and reliable way. A survey of state-of-the-art routing algorithms is provided in (Akkaya and Younis, 2005). Most proposed routing protocols in WSNs are based on shortest path algorithms, e.g., (Perkins and Bhagwat, 1994), (Park and Corson, 1997). Such algorithms usually require each node to maintain a global cost (or state) information table, which is a significant burden for resource-constrained WSNs. In order to deal with node failures, Ganesan et al. (Ganesan et al., 2001) proposed a multipath routing algorithm, so that a failure on the main path can be recovered without initiating a network-wide flooding process for path rediscovery. Since flooding consumes considerable energy, this routing method can extend the network's lifetime when there are failures. On the other hand, finding multiple paths and sending packets through them also consumes energy, thus adversely impacting the lifetime of the network if there are no failures. The routing policies mentioned above may indirectly reduce energy usage in WSNs, but they do not explicitly use energy consumption models to address optimality of a routing policy with respect to energy-aware metrics. Such "energy awareness" has motivated a number of minimum-energy routing algorithms which typically seek paths minimizing the energy per packet consumed (or maximizing the residual node energy) to reach a destination, e.g., (Singh and Raghavendra, 1998). However, seeking a minimum energy (or maximum residual energy) path can rapidly deplete energy from some nodes and ultimately reduce the full network's lifetime by destroying its connectivity. Thus, an alternative performance metric is the *network lifetime*. The definition of the term "lifetime" for WSNs varies. Some researchers, e.g., (Chang and Tassiulas, 2004), define the network lifetime as the time until the first node depletes its battery; however, this may just as well be defined as the time until the data source cannot reach the data sink (Bhardwaj and Chandrakasan, 2002). In this thesis, we will adopt the former definition, i.e., the time until the first node depletes its battery.

Along the lines of energy-aware routing, Shah and Rabaey (Shah and Rabaey, 2002) proposed an Energy Aware Routing (EAR) policy which does not attempt to use a single optimal path, but rather a number of suboptimal paths that are probabilistically selected with the intent of extending the network lifetime by "spreading" the traffic and forcing nodes in the network to deplete their energies at the same time. In (Paschalidis and Wu, 2012) a similar problem is studied with the inclusion of uncertainties in several WSN parameters. From a network security viewpoint, deterministic routing policies (i.e., policies where source nodes send data through one or more fixed paths) are highly vulnerable to attacks that can compromise a node and easily falsify cost information, leading to Denial of Service (DoS) attacks (Wood and Stankovic, 2002). In order to reduce the effect of such attacks, probabilistic routing is an interesting alternative, since this makes it difficult for attackers to identify an "ideal" node to take over. In this sense, the EAR policy is attractive because of its probabilistic routing structure, even though it does not attempt to provide optimal routing probabilities for network lifetime maximization. It is worth mentioning, however, that a routing policy based on probabilities can easily be implemented as a deterministic policy as well by transforming these probabilities to packet flows over links and using simple mechanisms to ensure that flows are maintained over time.

The network lifetime maximization problem studied in (Chang and Tassiulas, 2004) is based on two assumptions. First, it assumes that the energy in a battery depletes linearly with respect to the quantity of information forwarded, and does not depend on the physical dynamics of the battery itself. Second, it seeks fixed routing probabilities over time, even though the dynamic behavior of the WSN may in fact imply that a time-dependent (possibly based on state feedback closed-loop) routing policy may be optimal. More generally, routing problems in WSNs are based on ideal battery models where a battery maintains a constant voltage throughout the discharge process and a constant capacity for all discharge profiles, neither of which is generally true. In fact, the energy amount delivered by a battery heavily depends on the discharge profile and it is generally not possible to extract all the capacity stored in it (Panigrahi et al., 2001). This dynamic behavior also leads to the conjecture that an optimal routing policy should take into account the battery state over time and should, therefore, be time-dependent rather than fixed. Thus, an optimal control problem formulation for the network lifetime maximization problem seems to be a natural setting. Considering the network lifetime as the performance metric, our goal is to maximize it by finding an optimal routing strategy. As already mentioned, we consider the network lifetime as the time until the first node depletes its battery and we will see that this definition is a good characterization of the overall network's lifetime when the network topology is fixed. From a network security viewpoint, deterministic routing policies are highly vulnerable to attacks that can compromise a node and easily falsify cost information, leading to sinkhole attacks and as discussed, probabilistic routing can be a practical solution. Adopting ideal battery, in (Wu and Cassandras, 2005) routing was formulated as an optimal control problem with controllable routing probabilities over network links and it was shown that in a fixed network topology there exists an optimal policy consisting of time-invariant routing probabilities. Moreover, as shown in (Ning and Cassandras, 2009), the optimal control problem may be converted into the LP formulation used in (Chang and Tassiulas, 2004).

In this thesis, we adopt an optimal control setting with the goal of determining routing probabilities so as to maximize the lifetime of a WSN subject to a *dynamic* energy consumption model for each node. Due to their acceptable accuracy as well as manageable computational burden, we will use analytical battery models to capture the nonlinear dynamics of the battery for each node. In particular, at first we use the Kinetic Battery Model and formulate the problem as an Optimal Control Problem(OCP). We then generalize the results achieved under the KBM model by adopting a more elaborate battery model of which KBM is a special case. We also consider an alternative problem where, in addition to routing, we allocate a total initial energy over the network nodes with the same network lifetime maximization objective; the idea here is that a proper allocation of energy can further increase the network lifetime.

1.1.3 Lifetime Maximization of Wireless Sensor Networks with a Mobile Source Node

So far, we assume the network topology is fixed. Next, we investigate the lifetime maximization problem for the networks with changing topology. There are various ways to exploit WSN mobility by incorporating it into different network components. In recent years, mobility in WSNs has been increasingly introduced and studied (Rezazadeh et al., 2012), (Wang et al., 2005), and (Shah et al., 2003) with the aim of enhancing their capabilities. As discussed in (Di Francesco et al., 2011), mobility can affect different aspects of WSN design, including connectivity, cost, reliability and energy efficiency. For instance, in (Wang et al., 2005) sink mobility is exploited and a Linear Programming (LP) formulation is proposed for maximizing the network lifetime by finding the optimal sink node movement and sojourn time at different nodes in the network. In (Shah et al., 2003) mobile nodes (mules) are used to deliver data to the base station. For rechargeable WSNs, (Zhao et al., 2014) introduces a novel framework for joint energy replenishment and data gathering by employing multi-functional mobile nodes. WSNs with partial mobility are studied in (Srinivasan and Chua, 2007). As discussed in (Raja and Su, 2009), there exist two modes for sensor nodes mobility: weak mobility, forced by the death of some sensor nodes and strong mobility using an external agent (Laibowitz and Paradiso, 2005), (Dantu et al., 2005). By combining static wireless sensors and sophisticated mobile sensors, (Tseng et al., 2007) proposes a mobile, event-driven surveillance system. In a slightly closer setting to the problem investigated here, (He et al., 2004) studies the problem of tracking mobile targets using WSNs. In particular, an energy-efficient surveillance system is proposed for detecting and tracking the positions of mobile targets using cooperating static sensor nodes.

Scope of work

In this thesis, we focus on the lifetime maximization problem in WSNs when the source node is mobile. This situation frequently arises when a mobile sensor node is used to track one or more mobile targets or when there is a large area to be monitored that far exceeds the range of one or more static sensors. As already mentioned, in the case of a fully static network, the lifetime is defined as the time until the first node depletes its energy and it is a good characterization of the overall network's lifetime in practice for networks with fixed topology.

Adding mobility to nodes raises several questions. First, one can no longer expect that a routing policy would be time invariant. Second, it is no longer reasonable to define the WSN lifetime in terms of the the first node depleting its energy. Finally, if a routing policy is time-varying, then it has to be re-evaluated sufficiently fast to accommodate the real-time operation of a WSN.

Here, we consider mobility added to the source node and assume that any such node travels along a trajectory that it determines and which may or may not be known in advance. While on its trajectory, the source node continuously performs sensing tasks and generates data. Our goal is to derive an optimal routing scheme in order to maximize the network lifetime, appropriately redefined to focus on the mobile source node. We study two versions of this problem; first, we study the case with no a priori knowledge about the mobile node's trajectory; next, we investigate the case when the mobile node's trajectory is known in advance. Finally, we investigate how this information helps improving the network lifetime.

1.2 Optimal Routing for Battery-Powered Vehicles

We next study the routing problem for another group of BPS, Battery-Powered Vehicles (BPVs). Unlike WSNs, in routing problem for BPVs, entities are mobile and they have energy constraints. In general, when entities (e.g., EVs) in a network are characterized by physical attributes with a dynamic behavior, this behavior can play an important role in the routing decisions. In the case of BPVs the key physical attribute is *energy*.

From increasing energy security to reducing emissions of greenhouse gases, BPVs, such as Electric Vehicles (EVs), offer a new pathway to an energy efficient, environmentally friendly transportation system. Based on the International Energy Agency (IEA) road-map vision (IEA, 2011), at least 50% of Light Duty Vehicle (LDV) sales worldwide should include Electric and Plug-in Hybrid Electric Vehicles (EVs/PHEVs) by 2050. This significant rise of BPVs in traffic networks has introduced new challenges in classical network routing problems (Laporte, 1992). In general, BPVs face battery-related challenges which are crucial in routing problems including limited driving range, long recharge time, sparse coverage of charging stations, and the BPV energy recuperation ability (Artmeier et al., 2010) which can be exploited.

In recent years, the vehicle routing literature has been enriched by work aiming to accommodate the aforementioned BPV characteristics. For example, by incorporating the recuperation ability of EVs (which leads to negative energy consumption on some paths), extensions to general shortest-path algorithms are proposed in (Artmeier et al., 2010) that address the energy-optimal routing problem. The energy requirements in this problem are modeled as constraints and the proposed algorithms are evaluated in a prototypical navigation system. Extensions provided in (Eisner et al., 2011) employ a generalization of Johnson's potential shifting technique to make Dijkstra's algorithm applicable to the negative link cost shortest-path problem so as to improve the results and allow for route planning of EVs in large networks. This work, however, does not consider the presence of charging stations, modeled as nodes in the network. Charging times are incorporated into a multi-constrained optimal path planning problem in (Siddiqi et al., 2011), which aims to minimize the length of an EV's route and meet constraints on total traveling time, total time delay due to signals, total recharging time and total recharging cost; a particle swarm optimization algorithm is then used to find a suboptimal solution. In this formulation, however, recharging times are simply treated as parameters and not as controllable variables. In (Khuller et al., 2011), algorithms for several routing problems are proposed, including a single-vehicle routing problem with inhomogeneously priced refueling stations for which a dynamic programming based algorithm is proposed to find a least cost path from source to destination. In (Sweda and Klabjan., 2012), the same problem is revisited, assuming the recharging cost is a nonlinear function of the battery charging level and a dynamic programming algorithm is proposed to find a minimum-cost path for an EV. An EV Routing Problem with Time Windows and recharging stations (E-

VRPTW) was proposed in (Schneider et al., 2014), where an EV's energy constraint is first introduced into vehicle routing problems and recharging times depend on the battery charge of the vehicle upon arrival at the station. Controlling recharging times is circumvented by simply forcing vehicles to be always fully recharged. In (Worley et al., 2012), an integer programming optimization problem was formulated to simultaneously find optimal routes and charging station locations for commercial electric vehicles. In (Sachenbacher et al., 2011), a heuristic algorithm is proposed to find the energy-optimal routing for EVs taking into account the energy recuperation, battery capacity limitations and dynamic energy cost imposed by the vehicle properties. In recent work, (He et al., 2014) investigates the user-optimal network flow equilibrium with different scenarios for flow dependency of energy consumption of Battery Electric Vehicles (BEVs). Combinatorial optimization methods for different aspects of EV management such as energy-efficient routing and facility location problems are studied in (Touati-Moungla and Jost, 2012). In the Unmanned Autonomous Vehicle (UAV) literature, (Sunder and Rathinam, 2012) considers a UAV routing problem with refueling constraints. In this problem, given a set of targets and depots the goal is to find an optimal path such that each target is visited by the UAV at least once while the fuel constraint is never violated. A Mixed-Integer Nonlinear Programming (MINLP) formulation is proposed with a heuristic algorithm to determine feasible solutions.

1.2.1 Scope of work

Most of the aforementioned work deals with the routing problem for a single EV. This is not easily generalized to a multi-vehicle routing problem. In this thesis, we study the problem first from the driver's point of view (the user-centric "single-vehicle routing problem"), then from the system's point of view (the system-centric "multi-vehicle routing problem"). In the former, the goal is to find an optimal path along with a charging policy for a single EV acting “selfishly” to reach its destination in minimum time; under certain conditions, a Nash equilibrium may then be reached (Roughgarden, 2005). In the latter case, we define a system-wide objective and the goal is to route EVs so that a whole inflow reaches its destination in minimum time, therefore achieving a “social optimum”. We study these problems in networks with both homogeneous and inhomogeneous charging nodes where “inhomogeneity” means that charging rates at different nodes are not identical. For the user-centric case, first we formulate the problem as a MINLP which is the exact formulation. We then reduce the problem’s complexity by decomposing it into two simpler LP problems. For the network with homogeneous charging nodes, this problem decomposition doesn’t affect the optimality of the solution, however for the more complicated case, network with inhomogeneous charging nodes, the solution of the decomposed LP is sub-optimal in general. For the system-centric case, we follow the same framework. Starting with a MINLP formulation, we decompose it into simpler problems, but now due to the incorporated congestion effect, the decomposed problem remains as a MINLP with fewer decision variables. As an alternative approach, we study both user-centric and system-centric problems in a Dynamic Programming (DP) setting. We start our analysis with the assumption that every arriving vehicle is an EV. We then relax this assumption by considering both EVs with energy constraints and Non-Electric Vehicles (NEVs) in the inflow to the network. We again seek to optimize a system-centric objective by optimally routing NEVs and EVs along with an optimal policy for charging EVs along the way if needed.

1.3 The Price of Anarchy in Transportation Networks

Motivated by our game-theoretic approach in routing of BPVs, we next investigate the performance of transportation networks using actual traffic data. To do so, we borrow a well-known concept in game theory called the“ Price of Anarchy (POA)”. POA is a measure to compare system performance under a user-centric policy and a system-centric policy in a system with non-cooperative agents. A transportation (traffic) network is such a system in which each driver (agent) seeks to minimize her own cost by choosing the best route (resources) to reach her destination without taking into account the overall system performance (equivalent to the single-vehicle routing for BPVs). In these systems, the cost for each agent depends on the resources it chooses as well as the number of agents choosing the same resources (Wang et al., 2015). In such a non-cooperative setting, one often observes convergence to a Nash equilibrium, a point where no agent can benefit by altering its actions assuming that the actions of all the other agents remain fixed (Youn and Jeong, 2008). However, it is known that the Nash equilibrium is not always the best strategy from the system's point of view and results in a suboptimal behavior compared to the *socially optimal* policy (equivalent to the multiple-vehicle routing for BPVs).

1.3.1 Scope of work

In this thesis, Our goal is to quantify the social suboptimality of selfish driving in the Eastern Massachusetts traffic network by comparing the system performance, in terms of the total latency, under a user-optimal policy vs. a system-optimal policy using real traffic data.

The dataset at our disposal, provided by the Boston Region Metropolitan Planning Organization (MPO), includes the spatial average speeds and the flow capacity for each road segment of Eastern Massachusetts transportation network. We first infer equilibrium flows on each segment from the speed data which leads to obtain Origin-Destination (O-D) demand matrices. Next, we formulate a system-centric problem in which agents, here drivers, cooperate to optimize the overall system performance. This allows us to estimate the POA for a sub-network of the Eastern Massachusetts transportation network so as to determine the difference in network performance between selfish routing (non-cooperative) and system-optimal routing (cooperative). We also apply our optimal routing approach for BPVs to the same subnetwork. Using the estimated flow data and the O-D demand, we investigate the optimal solutions obtained under different charging station and energy-aware vehicle loads.

1.4 Analytical Tools

1.4.1 Optimal control approach and optimization methods

Optimal control theory focuses on problems with continuous states and exploits their rich differential structures in order to determine the inputs to the system that optimize a specified performance index while satisfying any constraints on the motion of the system (Rao, 2009), (Todorov, 2006). For some weakly nonlinear low dimensional systems, we my obtain the optimal solution analytically by investigating the necessary and sufficient conditions for optimality (Hamiltonian Analysis) (Bryson and Ho, 1975). To do so, one can employ the calculus of variation to obtain the optimality conditions, then by applying the Pontryagin Maximum Principle (PMP), the optimal solution may obtained analytically. However, due to the complexity of most applications, it is not possible to obtain a numerical solution for an Optimal Control Problem (OCP) in general. Thus, it is necessary to use numerical methods to obtain the optimal solution for a strongly nonlinear dynamical system.

Numerical methods for solving OCPs are classified into two types: indirect methods vs. direct methods (Rao, 2009). In an indirect method, the calculus of variations is employed to obtain the first order optimality conditions of the original OCP. These conditions leads to a Two Point (multi-point) Boundary Value Problem (TPBVP). Then by applying the PMP, one can use a gradient based method to find the optimal solution. Since in this method the solution is obtained by solving the TPBVP instead of the original OCP, it is called an indirect method. One disadvantage of this method is that it may have multiple solutions and the solution is highly sensitive to the initial state. Another disadvantage of this method is that boundary value problems are generally difficult to solve, specifically for problems with interior point constraints. On the other hand, a direct method is based on discretizing the state and/or control of the OCP and transcribing it to a Non-Linear Programming (NLP) problem. Then, the NLP is solved using well-known optimization techniques such as penalty function methods or methods of augmented or modified Lagrangian functions. Direct methods are widely used in commercial solver packages. As an example, GPOPS (Rao, 2009) uses a direct collocation method and simultaneously fully discretizes both controls as well as state variables which results in a large sparse NLP. Finally, this NLP can be solved using standard commercial solvers such as SNOPT (Gill et al., 2002).

Another alternative approach for designing optimal control is Dynamic Programming (DP). DP is an optimization approach that transforms a complex problem into a sequence of simpler problems and exploits the recursive nature of the problem. It relies on the Bellman's Principal of Optimality stating that "Any optimal policy has the property that, whatever the current state and decision, the remaining decisions must constitute an optimal policy with regard to the state resulting from the current decision." In other words, optimal policy in future is independent of the past action leading to the current state. Thus, starting from the final state, DP can provide a backward recursive algorithm to find the optimal policy. A key element for this procedure is the value function which is a function of the current state and provides the optimal value of the objective function completing the task starting from the current state. DP can be applied to both discrete time and continuous time dynamic optimization problems (both deterministic and stochastic problems), as well as static multi-stage optimization problems which can be transformed into a collection of simpler subproblems (e.g., shortest path problem).

In this thesis we use the optimal control approach to deal with the lifetime maximization problem in WSNs. For static networks, we adopt an optimal control setting with the goal of determining routing probabilities so as to maximize the lifetime of a WSN subject to a dynamic energy consumption model for each node. Using the Hamiltonian analysis and applying the Pontryagin Minimum Principle (PMP), we will show that there exists a time-invariant optimal control policy which minimizes the Hamiltonian and the OCP will reduce to a set of relatively simple Non-Linear Programming (NLP) problems. For networks with a mobile source, when the source node trajectory is unknown in advance, we use the same approach. In particular, we formulate an instantaneous optimal control problem that the WSN faces at each time step. Again, using the Hamiltonian analysis and applying PMP, we will show that optimal routing vectors are time-invariant and can be evaluated as solutions of a sequence of NLPs as the source node trajectory evolves. On the other hand, when the source node's trajectory is known in advance, we follow a similar approach by formulating an optimal control problem. In this case, due to the path and control constraints, the resulting OCP is more challenging and requires an explicit off-line numerical solution. Thus, we solve it numerically using GPOPS-II, a MATLAB-based general purpose optimal control software. This solver is based on a direct method and approximates a continuous-time OCP as a large sparse NLP using variable-order Gaussian quadrature collocation methods (Patterson and Rao, 2014). The resulting NLP is then solved using IPOPT, a standard NLP solver.

Another approach to the solution of an optimal control problem is to approximate the optimal policy structure (e.g., proposed by a numerical solver) by some parameters and transform the original OCP into a parametric optimization problem. Consequently rather than a "dynamic" optimization problem, we deal with a "static" optimization problem which is much easier to solve.

A general static optimization problem aims to minimize/maximize a given cost/benefit function by selecting n decision variables from a given feasible region. A feasible solution that optimize the cost function is called the optimal solution. The problem is unconstrained if the feasible region is $\mathcal{R}^n$. For the constrained problem, the feasible region is usually specified by a set of equality and inequality constraints.

The problem is called a Linear Programming problem (LP) if the objective function as well as the constraints are linear functions of decision variables. LP is a special case of convex programming and it can be solved efficiently using algorithms such as Simplex and interior-point methods (Bertsimas and Tsitsiklis, 1997).

The problem is called a Non-Linear Programming problem (NLP) if the objective function is nonlinear and/or feasible region is determined by nonlinear constraints. If the objective function is convex (minimization) or concave (maximization) and the feasible region is convex, the problem is a convex problem and the local optimum is equivalent to the global optimum. In contrast, if the feasible region and/or cost function are not convex for a minimization problem, the problem is non-convex and it may have several local optima.

If some or all decision variables are constrained to take on integer values (e.g., binary choices), the problem is an Integer Programming problem (IP). The problem is a Mixed Integer Non-Linear Programming (MINLP) when in addition it includes nonlinearity in the objective function and/or constraints. Discrete optimization problems are not convex and computationally difficult to solve. In fact, most of integer programming problems are NP-hard problems, e.g., Knapsack and Traveling Salesman problems.

In this thesis we face different optimization problems for both lifetime maximization of WSNs and the energy-constrained vehicle routing problems. For the routing problem in WSNs, we will reduce the OCP to a set of NLPs which should be solved in order to obtain optimal routing schemes. The complexity of the NLPs depends on the battery model we will adopt. For optimal routing of EVs, we investigate the problem form two point of view: user-centric vs system-centric. For the user-centric problem, we first formulate a MINLP as our exact formulation and show that one can decompose the exact formulation into two simpler LP problems: route selection and charging policy determination. For the system-centric problem, we formulate the exact problem as a MINLP and do the problem decomposition again. In this case, due to the traffic congestion effect on the cost function as well as constraints, the decomposed route selection problem remains as a MINLP with fewer variables. We finally introduce an alternative flow optimization formulation which is an NLP problem. The convexity of this problem relays on the choice of delay and energy consumption functions. We also formulate both user-centric and system-centric problems in a DP framework. For the user-centric problem, the DP formulation is effective and computationally less demanding than the exact MINLP. However, for the system-centric problem, the DP formulation is outperformed by the exact MINLP formulation when the problem size increases.

1.5 Contributions of This Work

In previous sections, we briefly point out the motivation of this dissertation, the scope of problems we will address and the methodologies we will use to solve these problems. In summary, the main contributions of this thesis are as follows:

1.5.1 Optimal routing of static wireless sensor networks

- We revisit the lifetime maximization problem for static wireless sensor networks studied in (Chang and Tassiulas, 2004) and (Wu and Cassandras, 2005) by relaxing the assumption of considering ideal battery dynamics for nodes. We adopt an optimal control setting with the goal of determining routing probabilities so as to maximize the lifetime of a WSN subject to a dynamic energy consumption model for each node. Using the Kinetic Battery Model (KBM), we show that in a fixed network topology case, there exists an optimal policy consisting of time-invariant routing probabilities determined through a set of relatively simple Non-Linear Programming (NLP) problems. We further show that under a very mild condition, this optimal routing policy is robust with respect to the battery model used, i.e., the routing probabilities are not affected by the battery model used, although the estimated WSN lifetime itself is significantly longer under a non-ideal battery model, primarily due to the recovery effect mentioned earlier. We also consider an alternative problem where, in addition to routing, we allocate a total initial energy over the network nodes with the same network lifetime maximization objective. We show that the optimal routing policy depletes all node energies simultaneously and the corresponding energy allocation and routing probabilities are obtained by solving a single NLP problem.

- We generalize the results obtained under the KBM for both the optimal routing and the joint routing and initial energy allocation problems for lifetime maximization by adopting a more elaborate non-ideal battery model and showing that the time-invariant nature of a maximal network lifetime routing policy is preserved. This leads to the conclusion that optimal policies for WSNs are indeed robust with respect to the battery model used. Strictly speaking, the robustness property suggests to find the optimal routing by adopting the ideal battery dynamics. This reduces the problem to a single LP. However, in order to precisely predict the network lifetime, we should use an accurate general non-ideal battery model while we apply the optimal routing obtained by the

LP formulation.

1.5.2 Optimal routing of wireless sensor networks with a mobile source node

- We redefine the lifetime for WSN with a mobile source node and study the lifetime maximization problem in two different settings: (*i*) when the source node's trajectory is unknown in advance vs. (*ii*) when we have prior knowledge about it. For case (*i*), we formulate the problem as instantaneous OCPs obtaining optimal routing scheme as the source node trajectory evolves. We show that the solution of these OCPs can be reduced to a sequence of NLPs which can be solved on line. For case (*ii*), when the mobile node's trajectory is known in advance, we formulate an optimal control problem which is computationally more challenging and requires an explicit off-line numerical solution. We also observe that having full knowledge about the source node trajectory may increase the network lifetime significantly compared to the case when there is no a priori trajectory knowledge.

1.5.3 Optimal routing of energy-limited vehicles

We study two versions of the problem: user-centric vs. system-centric and investigate the routing problem in networks with homogeneous as well as inhomogeneous charging stations.

Single-vehicle routing problem:

- We propose a MINLP optimization problem as the exact formulation which determines the optimal route and amount of charge at each node simultaneously. We then reduce the complexity of the exact MINLP by decomposing it into two Linear Programming (LP) problems: one to determine the optimal route and the other to find the charging policy over the optimal route. We do so by applying a locally-optimal charging policy and using some properties of the optimal solution. We show that the decomposed LP obtains the optimal solution for the network with homogeneous charging stations, though there is no guarantee for global optimality in general (for networks with inhomogeneous charging stations). We also formulate the problem in a Dynamic programming (DP) framework. This model is identical for both homogeneous and inhomogeneous charging nodes and allows us to find an optimal routing and charging policy for both cases in less computational time compared to the exact MINLP formulation.

Multiple-vehicle routing problem:

- The first challenge to address for this problem is to incorporate the traffic congestion effect on the traveling time and energy consumption over links. We do so by grouping vehicles into subflows and formulate the problem in subflow-level as a MINLP. This MINLP problem obtains optimal routes and charging policies for all subflows simultaneously.

- Adopting some properties of the optimal solution and applying a locally-optimal charging policy for each subflow, the original MINLP problem is simplified and decomposed into two problems: route selection and charging amount determination. In this case due to traffic congestion effects, the decomposed problem does not reduce to an LP. Similar to the results for the user-centric problem, when the network has inhomogeneous charging stations, the solution of the decomposed problem is sub-optimal, though it is optimal for the homogeneous case. To address the computational complexity required to solve the MINLP problem, a flow control formulation is proposed for this problem where we seek for the normalized vehicle flow on each arc for each subflow. Our numerical results show that this Non-Linear Programming formulation leads to a computationally simpler problem solution with minimal loss in accuracy. As an alternative approach to solve the exact formulation, a DP-based algorithm is also proposed to determine optimal routing and charging policies in subflow-level, by discretizing the residual energy for each subflow and defining an extended subflow-level graph. In this case, the problem size significantly increases with the number of subflows and the DP algorithm is eventually outperformed by our earlier MINLP approach as the number of subflows increases. Thus, to render the problem computationally manageabe, a proper selection of the number of subflows is essential. To address this, we propose a criterion and procedure for appropriate choice of the number of subflows reflecting a trade-off between computational complexity and proximity to an optimal objective value.

- We also study a more general problem in which the vehicle flow consists of both EVs and NEVs. We provide appropriate formulations to optimize a system-centric objective by optimally routing NEVs and EVs along with an optimal policy for charging EVs along the way if needed.

1.5.4 Price of Anarchy in transportation networks using real traffic data

We investigate the performance of the transportation network using a dataset including spatial average speed data for more than 13000 segments, composing the transportation network of the Eastern Massachusetts, for every minute of year 2012.

- We use the Greenshield's traffic flow model and infer equilibrium flows from average spatial speed data for each road segment of a highway sub-network of Eastern Massachusetts. We then feed the inferred equilibrium flow data to a Generalized least Squares (GLS) method and estimate the Origin-Destination (O-D) demand matrices for the network for different months and time-of-day periods.

- Using a set of data-driven cost functions (calculated for different months and time-of-day periods) and the estimated O-D demand matrices, we find socially optimum flows for different scenarios by solving the corresponding optimization problem. We finally quantify the POA for the inter-state highway subnetwork, by comparing the total latency obtained under user centric policy, equivalent to observed flows (estimated equilibrium flows), and that obtained under socially optimum flows.

1.6 Thesis Outline

The rest of this thesis is organized as follows. Chapters 2 and 3 address the lifetime maximization problem for wireless sensor networks. In Chapter 2, we study the problem of maximizing the lifetime of a static wireless sensor network by means of routing and initial energy allocation over its nodes. In our analysis, the energy sources (batteries) at nodes are behaving according to a dynamic energy consumption model which captures the nonlinear behavior of actual batteries. In Chapter 3, we investigate the problem of routing in wireless sensor networks when the source node is mobile. The goal is to maximize the network's lifetime which requires a new definition due to the mobility of the source node. We then consider two different settings: first we assume the mobile node's trajectory is unknown in advance; then we study the case when the mobile node's trajectory is known in advance.

Chapters 4 and 5 address the problem of routing for energy-aware Battery-Powered Vehicles in networks with charging nodes. The objective is to minimize the total elapsed time, including travel and recharging time at charging stations, so that the vehicle reaches its destination without running out of energy. We study both user-centric and system-centric routing problems. In Chapter 4, Starting with a MINLP formulation, we derive computationally simpler formulations to solve different versions of the problem. Then, we use a Dynamic Programming (DP) approach to solve the same problems in Chapter 5. We next utilize actual traffic data to study the performance of transportation networks under user-optimal vs. system-optimal strategies in Chapter 6. Finally, Chapter 7 summarizes the main results of the dissertation and discuss future research directions.

Chapter 2

Lifetime Maximization for Static Wireless Sensor Networks

2.1 Introduction

A Wireless Sensor Network (WSN) is a spatially distributed wireless network consisting of low-cost autonomous nodes which are mainly battery powered and have sensing and wireless communication capabilities (Megerian and Potkonjak, 2003). Applications range from exploration, surveillance, and target tracking, to environmental monitoring (e.g., pollution prevention, agriculture). Power management is a key issue in WSNs, since it directly impacts their performance and their lifetime in the likely absence of human intervention for most applications of interest. Since the majority of power consumption is due to the radio component (Shnayder et al., 2004), nodes usually rely on short-range communication and form a multi-hop network to deliver information to a base station. Routing schemes in WSNs aim to deliver data from the data sources (nodes with sensing capabilities) to a data sink (typically, a base station) in an energy-efficient and reliable way.

In this chapter, we focus on the problem of routing in a WSN with the objective of optimizing performance metrics that reflect the limited energy resources of the network while also preventing common security vulnerabilities for *static* (i.e., fixed topology) networks . Most proposed routing protocols in WSNs are based on shortest path algorithms, e.g., (Perkins and Bhagwat, 1994), (Park and Corson, 1997). Such algorithms usually require each node to maintain a global cost (or state) information table, which is a significant burden for resource-constrained WSNs. In order to deal with node failures, Ganesan et al. (Ganesan et al., 2001) proposed a multipath routing algorithm, so that a failure on the main path can be recovered without initiating a network-wide flooding process for path rediscovery.

Rather than indirectly reducing energy usage in WSNs, some energy-aware metrics have motivated a number of minimum-energy routing algorithms which typically seek paths minimizing the energy per packet consumed (or maximizing the residual node energy) to reach a destination, e.g., (Singh and Raghavendra, 1998). However, seeking a minimum energy (or maximum residual energy) path can rapidly deplete energy from some nodes and ultimately reduce the full network's lifetime by destroying its connectivity. Thus, an alternative performance metric is the *network lifetime*. There are different definitions for the term "lifetime" for WSNs. Some researchers define the network lifetime as the time until the first node depletes its battery (Chang and Tassiulas, 2004); however, it has been also defined as the time until the data source cannot reach the data sink in some works (Bhardwaj and Chandrakasan, 2002). In this thesis, we adopt the former definition, i.e., the time until the first node depletes its battery.

Along the lines of energy-aware routing, Shah and Rabaey (Shah and Rabaey, 2002) proposed an Energy Aware Routing (EAR) policy which does not attempt to use a single optimal path, but rather a number of suboptimal paths that are probabilistically selected with the intent of extending the network lifetime by "spreading" the traffic and forcing nodes in the network to deplete their energies at the same time. In (Paschalidis and Wu, 2012) a similar problem is studied with the inclusion of uncertainties in several WSN parameters.

The network lifetime maximization problem studied in (Chang and Tassiulas,

2004) is based on two assumptions. First, it assumes that the energy in a battery depletes linearly with respect to the quantity of information forwarded, and does not depend on the physical dynamics of the battery itself. Second, it seeks fixed routing probabilities over time, even though the dynamic behavior of the WSN may in fact imply that a time-dependent (possibly based on state feedback closed-loop) routing policy may be optimal. More generally, routing problems in WSNs are based on ideal battery models where a battery maintains a constant voltage throughout the discharge process and a constant capacity for all discharge profiles, neither of which is generally true. In fact, the energy amount delivered by a battery heavily depends on the discharge profile and it is generally not possible to extract all the capacity stored in it (Panigrahi et al., 2001). This dynamic behavior also leads to the conjecture that an optimal routing policy should take into account the battery state over time and should, therefore, be time-dependent rather than fixed. Thus, an optimal control problem formulation for the network lifetime maximization problem seems to be a natural setting. From a network security viewpoint, deterministic routing policies (i.e., policies where source nodes send data through one or more fixed paths) are highly vulnerable to attacks that can compromise a node and easily falsify cost information, leading to Denial of Service (DoS) attacks (Wood and Stankovic, 2002). In order to reduce the effect of such attacks, probabilistic routing is an interesting alternative, since this makes it difficult for attackers to identify an “ideal” node to take over.

In this chapter, we adopt an optimal control setting with the goal of determining routing probabilities so as to maximize the lifetime of a WSN subject to a *dynamic* energy consumption model for each node. In particular, we first use a *Kinetic Battery Model* (KBM) (Manwell and McGowan, 1994), (Rao et al., 2005) which has successfully been applied in other power mangement applications. We will then show that in a fixed network topology case there exists an optimal policy consisting of *time-invariant* routing probabilities. We subsequently show that the optimal control problem may be converted into a set of relatively simple Non-Linear Programming (NLP) problems. Moreover, we show that, under a very mild condition, this optimal routing policy is in fact robust with respect to the battery model used, i.e., the routing probabilities are not affected by the battery model used, although naturally the estimated WSN lifetime itself is significantly longer under a non-ideal battery model, primarily due to the recovery effect mentioned earlier. We also consider an alternative problem where, in addition to routing, we allocate a total initial energy over the network nodes with the same network lifetime maximization objective; the idea here is that a proper allocation of energy can further increase the network lifetime. We show that the solution to this problem is given by a policy that depletes all node energies *at the same time* and that the corresponding energy allocation and routing probabilities are obtained by solving again a NLP problem.

In view of these results, we next address the question of whether considering different, more elaborate, non-ideal battery models preserves the time-invariant nature of an optimal routing policy. In other words, is the relatively simple nature of the KBM responsible for this property or is this inherent in the problem regardless of how detailed a battery model one uses? We answer these questions by adopting the most general non-ideal battery model available in the literature and showing that the time-invariant nature of a maximal network lifetime routing policy is preserved. In fact, we generalize the results obtained under the KBM battery dynamics for both the optimal routing and the joint routing and initial energy allocation problems for lifetime maximization. This leads to the conclusion that optimal policies for WSNs are indeed robust with respect to the battery model used, although, naturally, the corresponding network lifetime value may be very different (therefore, accurately predicting the lifetime benefits from the increased accuracy of such general non-ideal battery models.) We note that when the battery behavior is reduced to a simple idealized model, our setting recovers that of (Wu and Cassandras, 2005) and (Ning and Cassandras, 2009) where it was shown that the set of NLP subproblems can in fact be transformed into the LP formulation in (Chang and Tassiulas, 2004). It was also shown in (Ning and Cassandras, 2009) that the initial energy allocation problem can be reformulated into a shortest path problem on a graph where the arc weights equal the link energy costs.

Finally we investigate WSN performance under common forms of security threats. We explore the network performance under one of the most severe routing attacks in WSN, namely the sink-hole attack (Krontiris et al., 2008). Although we limit ourselves to a simple empirical study, it becomes clear that the optimal policy we have derived is significantly more robust to common forms of cyber-attacks than other proposed energy-aware routing policies.

This chapter is organized as follows. In Section 2.2, we formulate the maximum lifetime optimization problem using non-ideal energy sources at nodes which have their own dynamics. We adopt a standard energy consumption model along with the aforementioned KBM. In Section 2.3, we show that for a fixed network topology there exists an optimal routing policy which is time invariant and identify a set of NLP problems which can be solved to obtain an explicit fixed optimal routing vector and the corresponding WSN lifetime. We also derive sufficient conditions under which this optimal policy is robust with respect to the battery model used. In Section 2.4, we consider a joint optimal routing and initial energy allocation problem. We show that in this case (under some conditions) it is optimal to set a routing vector and initial node energies so that all nodes have the same lifetime. An explicit solution can again be obtained by solving a NLP problem. We generalize the results obtained under the KBM battery dynamics by incorporating a more general non-ideal battery model into similar problems in Section 2.5. Finally, in section 2.6 we analyze the network performance when the network is under a “sink hole” type of routing attack in terms of its normalized throughput as a performance metric.

2.2 Optimal control problem formulation

In order to simplify our analysis, we will consider a WSN with a single source node and one base station and will assume a fixed topology. It will become clear that our methodology can be extended to multiple sources and one base station, as well as time-varying topologies, although the main fixed optimal routing result will obviously no longer hold in general.

2.2.1 Network model

Consider a network with $N+1$ nodes where 0 and N denote the source and destination (base station) nodes respectively. Except for the base station whose energy supply is not constrained, a limited amount of energy is available to all other nodes. Let $r_i(t)$ be the residual energy of node i, $i = 0, \dots, N-1$, at time t. The dynamics of $r_i(t)$ depend on the battery model used at node i; we will discuss in the next subsection the Kinetic Battery Model (KBM) we will adopt. The distance between nodes i and j at time t is denoted by $d_{i,j}(t)$; since we assume a fixed topology, we will treat $d_{i,j}(t)$ as time-invariant in the sequel. The nodes in the network may be ordered according to their distance to the destination node N so that $d_{1,N} \geq d_{2,N} \geq \cdots \geq d_{i,N} \geq \cdots \geq d_{N-1,N}$ and assume that $d_{0,N} > d_{i,N}$ for all $i = 1, \dots, N-1$.

Let O_i denote the set of nodes to which node i can send packets. We assume full coverage of the network and define $O_i = \{j : j > i,\ d_{i,j} < d_{i,N}\}$, where $j > i$ implies that $d_{i,N} > d_{j,N}$, i.e., a node only sends packets to those nodes that are closer to the destination , and $d_{i,j} < d_{i,N}$ means that a node cannot send packets to another node which is further away from it relative to the destination node N. It should be noted that this is a simplified model and it does not take into consideration the channel quality or the Shannon capacity of each wireless channel. (i.e., considering channel quality, it may be optimal to send data packets via a node located further from the base station but with a better channel condition.) We will use the notation $i \prec j$, if $j \in O_i$. Let $w_{i,j}(t)$ be the routing probability of a packet from node i to node j at time t. The vector $w(t) = [w_{0,1}(t), \ldots, w_{0,N-1}(t), \ldots, w_{N-2,N-1}(t)]'$ defines the control in our problem. We do not include $w_{0,N}(t), \ldots, w_{i,N}(t), \ldots, w_{N-1,N}(t)$ in the definition of $w(t)$, since it is clear that $w_{i,N}(t)$ is an implicit control variable given by $w_{i,N}(t) = 1 - \sum_{i \prec j,\, j<N} w_{i,j}(t), \quad i = 0, \ldots, N-2$.

For simplicity, the data sending rate of source node 0 is normalized to 1 and let $G_i(w)$ denote the data packet inflow rate to node i. Given the definitions above, we can express $G_i(w)$ through the following flow conservation recursive equation:

$$G_i(w) = \sum_{k \prec i} w_{k,i}(t) G_k(w), \ i = 1, \ldots, N \tag{2.1}$$

where $G_0(w) = 1$.

2.2.2 Non-ideal battery dynamics

Under the assumption that an electrochemical battery cell is “ideal,” a constant voltage throughout the discharge process and a constant capacity for all discharge profiles are both maintained over time. However, in real batteries the *rate capacity effect* (Doyle and Newman, 1997) leads to the loss of capacity with increasing load current and the *recovery effect* (Martin, 1999) makes the battery appear to regain portions of its capacity after some resting time. Due to these phenomena, the voltage as well as energy amount delivered by the battery heavily rest on the discharge profile. Therefore, when dealing with energy optimization, it is necessary to take this into account along with nonlinear variations in a battery’s capacity.

There are several proposed models to describe a non-ideal battery over-viewed in (Jongerden and Haverkort, 2008). Accordingly, models are broadly classified as electrochemical, circuit-based, stochastic, and analytical. Among all, analytical models like the *Kinetic Battery Model* (KBM) (Manwell and McGowan, 1994; Rao et al., 2005) or diffusion-based models (Vrudhula and Rakhmatov, 2003; Zhang and Shi, 2009; Barbarisi et al., 2006) provide a trade-off between accuracy and computational complexity. A detailed analysis of two analytical battery models, the KBM and diffusion models derived by Rakhmatov et al. (Rakhmatov and Vrudhula, 2001), is given in (Jongerden and Haverkort, 2009) where it is shown that the KBM is a first-order approximation of the popular Rakhmatov-Vrudhula-Wallach (RVW) diffusion model (Daler Rakhmatov, 2003).

While diffusion-based models are hard to combine with a performance model (Jongerden and Haverkort, 2008), a KBM combines speed with sufficient accuracy, as reported, for instance, in embedded system applications (Manwell and McGowan, 1994). Empirical evidence for the accuracy of the KBM is also provided in (Rao et al., 2005). The KBM was successfully used to study problems of optimal single and multi-battery power control in (Wang and Cassandras, 2013),(Wang and Cassandras, 2013) with results consistent with the use of a more elaborate linear state space model (Zhang and Shi, 2009) derived from the popular RVW diffusion-based model (Rakhmatov and Vrudhula, 2001). In this chapter we investigate the lifetime optimization problem for static WSNs while capturing the nonlinear battery dynamics using analytical models. We first use the simpler analytical battery model, KBM. We then generalize our results by adopting the more elaborate analytical battery model, diffusion-based model.

KBM Battery Model

The KBM models a battery as two wells of charge, as shown in Fig.2·1. The available-charge well (R-well) directly supplies electrons to the load while the bound-charge well (B-well) only supplies electrons to the R-well. The energy levels in the two wells are denoted by $r(t)$ and $b(t)$ respectively. The rate of energy flow from the B-well to the R-well is $k(b(t) - r(t))$, where k depends on the battery characteristics. The output $u(t)$ is the workload of the battery at time t.

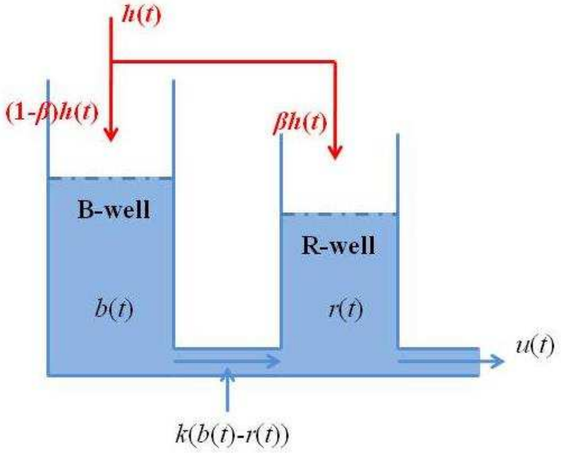


Figure 2·1: Kinetic Battery Model including recharging

The battery is said to be depleted when $r(t) = 0$. If a battery has rechargeability capabilities, we modify the KBM by adding a controllable input flow $h(t)$. For the sake of generality, we distribute the inflow $h(t)$ to both wells by adding a constant coefficient β $(0 \leq \beta \leq 1)$, as seen in Fig. 2·1. The resulting model is:

$$\dot{r}(t) = -c_1 u(t) + c_2 \beta h(t) + k(b(t) - r(t)) \tag{2.2}$$

$$\dot{b}(t) = c_2(1 - \beta)h(t) - k(b(t) - r(t)) \tag{2.3}$$

where c_1, c_2 are battery-specific influencing factors for the discharge outflow $u(t)$ and the recharge inflow $h(t)$ respectively; since, in general, a battery discharges faster than it can recharge, we assume $c_1 > c_2 \geq 0$ where the special case $c_2 = 0$ simply means the battery is not rechargeable. The state variables $r(t)$, $b(t)$ are physically constrained so that $b(0) \geq r(0)$ and $b(t) \leq \bar{B}$ where $\bar{B}$ is the battery capacity.

2.2.3 Energy consumption model

In our WSN environment, the battery workload $u(t)$ is due to three factors (e.g., see (Bhardwaj and Chandrakasan, 2002), (Bhardwaj et al., 2001)): the energy needed to sense a bit, E_{sense}, the energy needed to receive a bit, E_{rx}, and the energy needed to transmit a bit, E_{tx}. If the distance between two nodes is d, we have:

$$E_{tx} = p(d), \quad E_{rx} = C_r, \quad E_{sense} = C_e \tag{2.4}$$

where C_r, C_e are given constants dependent on the communication and sensing characteristics of nodes, and $p(d) \geq 0$ is a function monotonically increasing in d; the most common such function is $p(d) = C_f + C_s d^\beta$ where C_f, C_s are given constants and β is a constant dependent on the medium involved. We shall use this energy model but ignore the sensing energy, i.e., set $C_e = 0$. Clearly, this is a relatively simple energy model that does not take into consideration the channel quality or the Shannon capacity of each wireless channel. The ensuing optimal control analysis is not critically dependent on the exact form of the energy consumption model attributed to communication, although the ultimate optimal value of $w(t)$ obviously is. Before proceeding, it is convenient to define the following constants:

$$k_{i,j} = p(d_{i,j}) - p(d_{i,N}), \quad i < j < N \tag{2.5}$$

$$k_{0,N} = p(d_{0,N}) \tag{2.6}$$

$$k_{i,N} = C_r + p(d_{i,N}), \quad i = 1 \ldots N-1 \tag{2.7}$$

where $d_{i,j}$ is the distance between nodes i and j. Note that we may allow these constant to be time-dependent if the network topology is not fixed, i.e., $d_{i,j}(t)$ is time-varying. Let us now combine the KBM model above with (2.4). Although the ability to recharge a battery offers an interesting possibility for routing control, we shall not take it into account in this chapter, i.e., set $c_2 = 0$ in (2.2)-(2.3). Moreover, for simplicity, we set $c_1 = 1$. Then, starting with node 0, the workload $u_0(t)$ at that node is given by

$$\begin{aligned} u_0(t) &= \sum_{0 \prec j} w_{0,j}(t) p(d_{0,j}) \\ &= \sum_{0 \prec j, j<N} w_{0,j}(t) p(d_{0,j}) + w_{0,N}(t) p(d_{0,N}) \\ &= \sum_{0 \prec j, j<N} w_{0,j}(t) p(d_{0,j}) + [1 - \sum_{0 \prec j, j<N} w_{0,j}(t)] p(d_{0,N}) \\ &= \sum_{0 \prec j, j<N} w_{0,j}(t) [p(d_{0,j}) - p(d_{0,N})] + p(d_{0,N}) \\ &= \sum_{0 \prec j, j<N} w_{0,j}(t) k_{0,j} + k_{0,N} \\ &= G_0(w) \left[\sum_{0 \prec j, j<N} w_{0,j}(t) k_{0,j} + k_{0,N} \right] \end{aligned}$$

where we have used the fact that $G_0(w) = 1$. Similarly, for any node $i = 1, \ldots, N-1$, where we must include the energy for both receiving and transmitting data packets, we get

$$u_i(t) = G_i(w) \left[\sum_{i \prec j, j<N} w_{i,j}(t) k_{i,j} + k_{i,N} \right]$$

Defining $g_i(w) = \sum_{i<j,j<N} w_{i,j}(t) k_{i,j} + k_{i,N}$ the KBM equations (2.2)-(2.3) for nodes $i = 0, \ldots, N-1$ become

$$\dot{r}_i(t) = -G_i(w(t)) g_i(w(t)) + k(b_i(t) - r_i(t)) \tag{2.8}$$

$$\dot{b}_i(t) = -k(b_i(t) - r_i(t)) \tag{2.9}$$

The vectors $r(t) = [r_0(t), \ldots, r_{N-1}(t)]$ and $b(t) = [b_0(t), \ldots, b_{N-1}(t)]$ define the state variables for our problem. Observe that controlling the routing probabilities $w_{i,j}(t)$ means indirectly controlling node i's battery discharge process.

2.2.4 Optimal control problem formulation

Our objective is to maximize the WSN lifetime by controlling the routing probabilities $w_{i,j}(t)$ (equivalently, the flows through all network links). The WSN lifetime is defined as $T = \min_{0 \le i < N} T_i$ where T_i is given by $T_i = \inf\{t : r_i(t) = 0,\, t \ge 0\}$. Thus, our objective is to maximize T. Using the energy consumption model we have developed above, the optimal control problem is formulated as follows:

$$\min_{w(t)} -\int_0^T dt \tag{2.10}$$

s.t.

$$\dot{r}_i(t) = -G_i(w(t))g_i(w(t)) + k(b_i(t) - r_i(t)), r_i(0) = R_i \tag{2.11}$$

$$\dot{b}_i(t) = -k(b_i(t) - r_i(t)), \quad b_i(0) = B_i \tag{2.12}$$

$$G_i(w(t)) = \sum_{k \prec i} w_{k,i}(t) G_k(w),\ i = 1, \ldots, N-1, \quad G_0(w(t)) = 1 \tag{2.13}$$

$$g_i(w(t)) = \sum_{i \prec j, j < N} w_{i,j}(t) k_{i,j} + k_{i,N} \tag{2.14}$$

$$\sum_{i \prec j, j < N} w_{i,j}(t) \le 1, \quad 0 \le w_{i,j}(t) \le 1, i = 0, \ldots, N-1 \tag{2.15}$$

$$\min_{i=0,\ldots,N-1} r_i(T) = 0 \tag{2.16}$$

where $r_i(t), b_i(t)$ are the state variables representing node i's instantaneous battery energy level, $i = 0, \ldots, N-1$. Control constraints are specified through (2.15), where the first inequality follows from the fact that $\sum_{i \prec j < N} w_{i,j}(t) + w_{i,N}(t) = 1$. Finally, (2.16) provides boundary conditions for $r_i(t)$, $i = 0, \ldots, N-1$, at $t = T$ requiring that the terminal time is the earliest instant when $r_i(t) = 0$ for any node i. In other words, at $t = T$ we require that the minimal value over all $\{r_0(T), \ldots, r_{N-1}(T)\}$ is 0 or, equivalently, $T = \inf_{t \geq 0}\{t : r_i(t) = 0 \text{ for at least some } i = 0, \ldots, N-1\}$.

This is a classic minimum (maximum) time optimal control problem except for two complicating factors: (i) The boundary condition (2.16) which involves the nondifferentiable min function, and (ii) The control constraints (2.15). In what follows, we will use $w^*(t)$ to denote the optimal routing vector, which provides a (not necessarily unique) solution to this problem.

Remark 1: Note that there is an additional state constraint imposed by the capacity of every node battery, i.e., $b_i(t) \leq B_i$. However, it is easy to show (see (Wang and Cassandras, 2013)) that as long as $R_i < B_i$, it is always true that $r_i(t) < b_i(t) < \bar{B}_i$ for all $t > 0$ so that this constraint is never active in our problem (intuitively, since all batteries are being discharged and never recharged, it is not possible for a capacity to be reached except at $t = 0$.) Moreover, if $B_i = R_i$, then $r_i(t) < b_i(t) < \bar{B}_i$ as long as $G_i(w(t))g_i(w(t)) > 0$ for all $t > 0$. Since $k_{i,N} > 0$ in (2.7) for all $i = 0, \ldots, N-1$, this is always true unless a node i is not used in the network, i.e., $w_{k,i}(t) = 0$ for all $k \prec i$. In addition, observe that when the battery is "at rest", i.e., there is no load in (2.11), it is easy to show that $\lim_{t \to \infty}(b_i(t) - r_i(t)) = 0$. Therefore, we normally set inital conditions so that $B_i = R_i$.

2.3 Optimal control problem solution

In standard optimal control theory the Hamiltonian is defined as $H(x, \lambda, u, t) = -L(x, u, t) + \lambda^T(t) f(x, u, t)$ where $\dot{x} = f(x, u, t)$ are the state dynamics, $L(x, u, t)$ is the integrand in the objective function, and $\lambda(t)$ is the vector of costate variables interpreted as Lagrange multipliers associated with the state equations. We begin with the Hamiltonian for this optimal control problem:

$$H(w,t,\lambda) = -1 + \sum_{i<N}[\lambda_{i1}(t)\big(- G_i(w(t))g_i(w(t)) + k(b_i(t) - r_i(t))\big) - \lambda_{i2}(t)\big(k(b_i(t) - r_i(t))\big)] \tag{2.17}$$

where $\lambda_{i1}(t)$, $\lambda_{i2}(t)$ are the costates corresponding to $r_i(t)$ and $b_i(t)$ at node i, which must satisfy

$$\begin{cases} \dot{\lambda}_{i1}(t) = -\frac{\partial H}{\partial r_i} = k[\lambda_{i1}(t) - \lambda_{i2}(t)] \\ \dot{\lambda}_{i2}(t) = -\frac{\partial H}{\partial b_i} = -k[\lambda_{i1}(t) - \lambda_{i2}(t)] \end{cases} \tag{2.18}$$

To derive explicit expressions for $\lambda_{i1}(t)$, $\lambda_{i2}(t)$ it is necessary to use boundary conditions $\lambda_{i1}(T)$, $\lambda_{i2}(T)$. This is complicated by the nature of the state boundary conditions in (2.16). Thus, we proceed by considering each possible case of a node dying first which we will refer to as "scenario S_i" under which $0 = r_i(T) \leq r_j(T), \quad j \neq i$ for some fixed node i.

2.3.1 Analysis of scenario S_i

Under S_i, we have the terminal time constraints $r_i(T) = 0$ and $r_j(T) \geq 0$ for all $j \neq i$. Consequently, all $r_j(t)$, $j \neq i$, are unconstrained at $t = T$. The next theorem establishes the property that, under a fixed network topology, there exists a static optimal routing policy, i.e., there exists a vector $w^*(t)$ which is time invariant.

Theorem 1: If $0 = r_i(T) \leq r_j(T)$, $j \neq i$, for some i and the network topology is fixed, i.e., $d_{ij}(t) = d_{i,j} =$ constant for all $i, j = 0, \ldots, N-1$, then there exists a time-invariant solution of (2.10)-(2.16):

$$w^*(t) = w^*(T).$$

Proof: Since $r_i(t) \geq 0$ for all i and $t \in [0,T]$, the optimal control problem under S_i is state-unconstrained except for $r_i(T) = 0$. Thus, the terminal state constraint function $\Phi(r(T), b(T))$ is reduced to $r_i(T)$ and the costate boundary conditions (Bryson and Ho, 1975) are given by:

$$\left\{ \begin{array}{l} \lambda_{i1}(T) = \nu \frac{\partial \Phi(r(T), b(T))}{\partial r_i} = \nu \\ \lambda_{i2}(T) = 0 \end{array} \right. , \left\{ \begin{array}{l} \lambda_{j1}(T) = 0 \\ \lambda_{j2}(T) = 0 \end{array} \right. , \quad j \neq i$$

where ν is an unspecified scalar constant. This allows us to solve the costate equations in (2.18) to obtain for $t \in [0, T]$:

$$\begin{array}{l} \lambda_{i1}(t) = \frac{\nu}{2}(1 + e^{2k(t-T)}) \\ \lambda_{i2}(t) = \frac{\nu}{2}(1 - e^{2k(t-T)}) \end{array} , \begin{array}{l} \lambda_{j1}(t) = 0 \\ \lambda_{j2}(t) = 0 \end{array} , \; j \neq i \tag{2.19}$$

Using (2.19) in (2.17), we can simplify the Hamiltonian as follows:

$$\begin{aligned} H(w, t, \lambda) = -1 + \lambda_{i1}(t)[-G_i(w(t))g_i(w(t)) + \\ (b_i(t) - r_i(t))] - \lambda_{i2}(t)[k(b_i(t) - r_i(t))] \end{aligned} \tag{2.20}$$

Observe that the control variables $w_{i,j}(t)$ appear only in $G_i(w(t))$ and $g_i(w(t))$ in the problem formulation (2.10)-(2.16). Thus, we can set $U_i(t) = G_i(w(t))g_i(w(t))$, $i = 0, \ldots, N-1$ to be the effective control variables with $U_l \leq U_i(t) \leq U_u$, where $U_l \geq 0$ and U_u are, respectively, the lower bound and upper bound of $U_i(t)$ for all $t \in [0, T]$. Note that both are constant since their determination depends exclusively on (2.13), (2.14) subject to (2.15), independent of the states $r_i(t)$ and $b_i(t)$. In particular, they depend on the fixed network topology and the values of the energy parameters $k_{i,j}$, $k_{i,N}$ in (2.14). Applying the Pontryagin minimum principle to (2.20):

$$U_i^*(t) = \arg \min_{U_l \leq U_i(t) \leq U_u} H(U_i, t, \lambda^*)$$

implies that the optimal control is of bang-bang type:

$$U_i^*(t) = \left\{ \begin{array}{ll} U_u & \text{if } \lambda_{i1}(t) > 0 \\ U_l & \text{if } \lambda_{i1}(t) < 0 \end{array} \right. \tag{2.21}$$

with the possibility that there is a singular arc on the optimal trajectory if $\lambda_{i1}(t) = 0$. Moreover, the optimal solution must satisfy the transversality condition (Bryson and Ho, 1975) $\left(\lambda^* \frac{d\Phi}{dt} + L\right)_{t=T} = 0$ where $L = -1$ and we have seen that $\Phi(r(T), b(T)) = r_i(T)$. Therefore,

$$-1 + \nu \dot{r}_i(T) = -1 + \nu[-G_i(w(T))g_i(w(T)) + kb_i(T)] = 0$$

and it follows that

$$\nu = \frac{1}{-G_i(w(T))g_i(w(T)) + kb_i(T)} \tag{2.22}$$

Observing that $\nu \neq 0$ and looking at (2.19), we can immediately exclude the singular case $\lambda_{i1}(t) = 0$. Moreover, since $r_i(T) = 0$ and $r_i(t) > 0$ for all $t \in [0, T)$, it follows that $\dot{r}_i(T) < 0$ and (2.22) implies that $\nu < 0$. Therefore, from (2.19), $\lambda_{i1}(t) < 0$ throughout $[0, T)$. Consequently, $U_i^*(t) = U_l$ for $t \in [0, T]$ by (2.21). We conclude that the optimal control problem under S_i is reduced to the following optimization problem:

$$\min_{w(t)} G_i(w(t))g_i(w(t)) \tag{2.23}$$

$$\text{s.t.} \quad (2.13) - (2.15) \text{ and } 0 = r_i(T) \leq r_j(T), \quad j \neq i$$

When $t = T$, the solution of this problem is $w^*(T)$ and depends only on $r_j(T)$, $j \neq i$, and, as already argued, the fixed network topology and the values of the fixed energy parameters $k_{i,j}$, $k_{i,N}$ in (2.14). The same applies to any other $t \in [0, T)$, therefore, there exists a time-invariant optimal control policy $w^*(t) = w^*(T)$, which minimizes the Hamiltonian and proves the theorem. ■

Note that there may exist multiple optimal control policies, including some that may be time varying. Theorem 1 asserts that there is at least one which is time-invariant, i.e., $w^*(t) = w^*(T) = w^*$, and it remains to obtain the values of $w_{i,j}^*$,

$i = 0, \dots, N-2$ and $j = 1, \dots, N-1$, by explicitly solving the optimization problem (2.23). This requires knowledge of all $r_i(t)$, $t \in [0, T]$ in order to determine the values of all $r_i(T)$ and hence identify the node i such that $0 = r_i(T) \leq r_j(T)$ and use the values of $r_j(T)$, $j \neq i$. This can be accomplished by solving the differential equations (2.11)-(2.12), whose initial conditions are given, with $w(t) = w$, the unknown optimal routing vector. It is straightforward to obtain $r_i(t)$ and hence show that the "crtitical time" T_i^* such that $r_i(T_i^*) = 0$ and $r_i(t) > 0$ for all $t \in [0, T_i^*)$ is the solution of the nonlinear equation in T:

$$R_i - \frac{G_i(w)g_i(w)}{2}T - \frac{1}{2}[B_i - R_i - \frac{G_i(w)g_i(w)}{2k}](e^{-2kT} - 1) = 0 \tag{2.24}$$

which we write as $T_i^*(w)$. Thus, we may rewrite the S_i optimization problem as follows

$$\begin{aligned} \mathbf{P}_i : \ & \min_{w} G_i(w)g_i(w) \\ & \text{s.t.} \quad (2.13) - (2.15), \quad T_i^*(w) \leq T_j^*(w), \quad j \neq i \end{aligned}$$

where $T_i^*(w)$ is the solution of (2.24) for all $i = 0, \dots, N-1$. We will refer to this as problem $\mathbf{P}_i$ and note that it may not always have a feasible solution. The following Lemma establishes upper and lower bounds for $T_i^*(w)$ based on which necessary conditions for $\mathbf{P}_i$ to have a feasible solution may be derived. Before proceeding, we return to the definitions of the energy consumption constants in (2.5)-(2.7) and recall that $k_{i,N} > 0$ for all $i = 0, \dots, N-1$. Moreover, since $d_{i,j} < d_{i,N}$ if $i \prec j$ and $j < N$, we have

$$k_{i,j} = p(d_{i,j}) - p(d_{i,N}) < 0, \quad \text{if } i \prec j \text{ and } j < N \tag{2.25}$$

Let us also define

$$\gamma(i) = \arg \min_{i \prec j, j < N} k_{i,j} \tag{2.26}$$

From the definition of $k_{i,j}$, this is the nearest node in the output node set of i.

Lemma 1: For all $i \neq 0$,

$$\frac{R_i}{k_{i,N}} \leq T_i^*(w) \leq \infty \tag{2.27}$$

and for $i = 0$:

$$\frac{R_0}{k_{0,N}} \leq T_0^*(w) \leq \frac{R_0}{k_{0,\gamma(0)} + k_{0,N} - k\bar{B}_0} \tag{2.28}$$

Proof: We begin with a lower bound for $T_i^*(w)$, $i = 0, \ldots, N-1$. Recalling the state equation (2.11) and observing that $k(b_i(t) - r_i(t)) \geq 0$, it follows that a lower bound for $T_i^*(w)$, when $r_i(t)$ first reaches zero, is given by the value of w that maximizes $\left[G_i(w) \sum_{i \prec j,\, j<N} w_{i,j}k_{i,j} + k_{i,N}\right]$, i.e.,

$$T_i^*(w) \geq R_i \left[G_i(w) \sum_{i \prec j,\, j<N} w_{i,j}k_{i,j} + k_{i,N}\right]^{-1} \tag{2.29}$$

The inflow rate $G_i(w)$ is upper-bounded by the sending rate of the source $G_0(w) = 1$, therefore $G_i(w) \leq 1$. Thus,

$$G_i(w)\left(\sum_{i \prec j,\, j<N} w_{i,j}k_{ij} + k_{iN}\right) \leq \sum_{i \prec j,\, j<N} w_{i,j}k_{ij} + k_{iN} \tag{2.30}$$

Next, consider $\sum_{i \prec j,\, j<N} w_{i,j}k_{i,j} + k_{i,N}$. In view of (2.25) and $k_{i,N} > 0$, setting $w_{i,j} = 0$ for all $j < N$ and $i \prec j$ attains the maximal value of this expression, i.e.,

$$\sum_{i \prec j,\, j<N} w_{i,j}k_{i,j} + k_{i,N} \leq k_{i,N} \tag{2.31}$$

Combining (2.30) and (2.31), we have

$$G_i(w)\left(\sum_{i \prec j,\, j<N} w_{i,j}k_{i,j} + k_{i,N}\right) \leq k_{i,N}$$

and it follows from (2.29) that

$$T_i^*(w) \geq \frac{R_i}{k_{i,N}} \tag{2.32}$$

Regarding an upper bound for $T_i^*(w)$, if $i \neq 0$ it is possible to have $G_i(w) = 0$, while the upper bound for the term $k(b_i(t) - r_i(t))$ in (2.11) is $k\bar{B}_i$ where $\bar{B}_i$ is the battery capacity. Hence, we can only write $T_i^*(w) \leq \infty$ and this establishes (2.27).

If $i = 0$, we have $G_0(w) = 1$ and it follows from (2.11) that

$$\dot{r}_0(t) \leq -\left(\sum_{0 \prec j,\, j<N} w_{0,j}k_{0,j} + k_{0,N}\right) + k\bar{B}_0$$

Therefore, an upper bound for $T_0^*(w)$ is obtained by minimizing $[\sum_{0 \prec j,\, j<N} w_{0,j}k_{0,j} + k_{0,N} - k\bar{B}_0]$. This entails solving a Linear Programming (LP) problem as follows:

$$\begin{aligned} \min_{w} \quad & \sum_{0 \prec j,\, j<N} w_{0,j}k_{0,j} + k_{0,N} - k\bar{B}_0 \\ \text{s.t.} \quad & \sum_{0 \prec j,\, j<N} w_{0,j} \leq 1, \quad 0 \leq w_{0,j} \leq 1 \end{aligned}$$

For this problem, one of the extreme points of the feasible set will be an optimal solution. There are N extreme points $[w_{0,1}, \ldots, w_{0,N-1}]$ such that:

$$w_{0,j} = \begin{cases} 1 & \text{if } j = m \\ 0 & \text{otherwise} \end{cases}, \quad m = 1, \ldots, N-1 \tag{2.33}$$

and the point $[0, \ldots, 0]$. The latter cannot minimize the objective function, since, from (2.25), we know that $k_{0,j} < 0$. Thus, the optimal extreme point must be one of the $N - 1$ extreme points in (2.33). In this case, the objective function becomes

$$\begin{aligned} \sum_{0 \prec j,\, j<N} w_{0,j}k_{0,j} + k_{0,N} - k\bar{B}_0 = & \; w_{0,m}k_{0,m} + k_{0,N} - k\bar{B}_0 \\ = & \; k_{0,m} + k_{0,N} - k\bar{B}_0 \end{aligned}$$

for some $m = 1, \ldots, N - 1$. Thus, in order to minimize the objective function, we need find the smallest $k_{0,m}$. It follows from (2.26) that the optimal extreme point is such that

$$w_{0,j} = \begin{cases} 1 & \text{if } j = \gamma(0) \\ 0 & \text{otherwise} \end{cases}$$

and the optimal value is $k_{0,\gamma(0)} + k_{0,N} - k\bar{B}_0$. It follows that

$$T_0^*(w) \leq \frac{R_0}{k_{0,\gamma(0)} + k_{0,N} - k\bar{B}_0} \tag{2.34}$$

which along with (2.32) proves (2.28) and completes the proof. ■

Note that it is possible for $k_{0,\gamma(0)} + k_{0,N} - k\bar{B}_0$ to be negative. In practice, however, values of the battery parameter k are small and likely to make the contribution of $k\bar{B}_0$ much smaller than $k_{0,\gamma(0)} + k_{0,N}$. Lemma 1 allows us to determine necessary conditions for $\mathbf{P}_i$ to have a feasible solution. In particular, if $i \neq 0$ and

$$\frac{R_i}{k_{iN}} > \frac{R_0}{k_{0\gamma(0)} + k_{0N} - k\bar{B}_0} \tag{2.35}$$

then $T_i^*(w) > T_0^*(w)$ and $\mathbf{P}_i$ has no feasible solution. Thus, the necessary condition that $\mathbf{P}_i$ $(i > 0)$ to have a feasible solution is

$$\frac{R_i}{k_{iN}} \leq \frac{R_0}{k_{0\gamma(0)} + k_{0N} - k\bar{B}_0} \tag{2.36}$$

2.3.2 Algorithm for solving the optimal control problem

Based on our analysis thus far, if we focus on a fixed scenario S_i, the solution to the optimal control problem is simply the solution of the NLP problem $\mathbf{P}_i$. However, since we do not know which node will die first, determining the value of i such that $T_i^*(w) \leq T_j^*(w)$ for all $j \neq i$ requires solving *all* $\mathbf{P}_i$ problems and find the best policy among them. Since not all $\mathbf{P}_i$ problems have feasible solutions we can use (2.36) to check for feasibility before solving the associated NLP problem. The complete algorithm, referred to as **A1**, to solve this optimal control problem is as follows.

Algorithm A1:

1. Solve problem $\mathbf{P}_0$ to obtain $T_0^*(w)$.

2. For $0 < i < N$, if $\frac{R_i}{k_{iN}} > \frac{R_0}{k_{0\gamma(0)}+k_{0N}-k\bar{B}_0}$, set $T_i^*(w) = -1$ (no feasible solution exists); otherwise solve problem $\mathbf{P}_i$ and obtain $T_i^*(w)$ if it exists.

3. The optimal lifetime is given by $\max_i \{T_i^*(w)\}$ and the corresponding optimal policy w^* is the one obtained for the associated problem $\mathbf{P}_i$.

If the network topology is such that every node i can communicate with every downstream node j, then the algorithm can be substantially simplified due to the following result.

Lemma 2: For a single-source fixed topology network such that $O_i = \{j : j = i+1, \ldots, N\}$ for all $i = 0, \ldots, N-1$, then the source node lifetime is no longer than any other node lifetime under the optimal routing policy w^*, i.e.,

$$T_0^*(w^*) \leq T_i^*(w^*), \quad \text{for all } i = 1, \ldots, N-1$$

Proof: There are two cases to consider:

Case 1: If $w_{0,N}^* = 1$, then it is obvious that $T_0^*(w^*) \leq T_i^*(w^*)$, for all $i = 1, \ldots, N-1$ since none of the non-source nodes is used.

Case 2: If $w_{0,N}^* < 1$, we use a contradiction argument. Let us assume that under the optimal routing vector w^* there exists a node, say $p > 0$, which dies first in the network, i.e., $T_p^*(w^*) = T^* < T_0^*(w^*)$. Next, let us introduce the following perturbation to the optimal routing vector:

$$w'_{m,n} = \begin{cases} w_{m,n}^* + K\epsilon & \text{if } m = 0, n = N \\ w_{mn}^* - \epsilon & \text{if } m = 0,\ 1 \leq n \leq N-1, w_{m,n}^* > 0 \\ w_{m,n}^* & \text{otherwise} \end{cases}$$

where $\epsilon > 0$ is sufficiently small so that the new routing policy w' is still feasible, and $K = \sum_{j=1}^{N} \mathbf{1}[w_{0,j} > 0]$ where $\mathbf{1}[w_{0,j} > 0]$ is the usual indicator function (this is always possible for $w^*_{0,n} > 0$). In other words, we only perturb routing probabilities from the source node 0 to other nodes. Consequently, we increase the flow rate from the source to the sink node, and decrease flow rates into other nodes so as to maintain the same total flow out of node 0. It follows that the source node's life must decrease since it sends more traffic through the longest link. At the same time, the lifetimes of all other nodes receiving positive flows from node 0 must increase since the inflow rates into all of them decrease . Therefore, letting T'_i denote the node i lifetime under the perturbed routing vector w', we have

$$T'_0 = T^*_0(w^*) - f_0(\epsilon), T^{'}_j = T^*_j(w^*) + f_j(\epsilon), \quad j = 1, \ldots, N-1$$

where $f_k(x)$,$k = 0, \ldots, N-1$, is a continuous function such that $f_k(x) \geq 0$ and $f_k(0) = 0$. Since $f_k(x)$ is continuous, we can find a small enough $\epsilon > 0$ such that $T^{'}_p < T^{'}_0$ so that the source node 0 cannot die first under w'. Therefore, the lifetime under routing policy w' is $T' = \min_{j \neq 0} T^{'}_j$.

Since the lifetimes of all non-source nodes increase under w', it follows that $T' = \min_{j \neq 0} T^{'}_j > T^*$. In other words, w' provides a longer network lifetime that w^* contradicting the assumption that w^* is optimal. ■

This lemma allows us to reduce the original optimal control problem to a single problem $\mathbf{P}_0$ as follows:

$$\mathbf{P}_0 : \min_{w} g_0(w) \tag{2.37}$$

$$\text{s.t.} \quad (2.13) - (2.15) \text{ and } T^*_0(w) \leq T^*_i(w), \quad i > 0$$

where we have used the fact that $G_0(w) = 1$. Clearly, this provides a much simpler approach to the solution.

Remark 2: Our analysis can recover the ideal battery case by setting $k = 0$ in (2.11)-(2.12). We can then obtain $T_i^*(w)$ for a fixed routing vector w from $\dot{r}_i(t) = -G_i(w)g_i(w)$, $r_i(0) = R_i$ as $T_i^*(w) = R_i \left[G_i(w)g_i(w)\right]^{-1}$ which greatly simplifies the process of obtaining a solution through **Algorithm A1**. In this case, as shown in (Ning and Cassandras, 2009), the set of NLP problems $\mathbf{P}_i$ can be transformed into the LP formulation in (Chang and Tassiulas, 2004).

2.3.3 A robustness property of the optimal routing policy

In this section, we show that the optimal routing vector w^* obtained through **Algorithm A1** under the ideal battery assumption, i.e., $k = 0$ in (2.11)-(2.12), is often unchanged when the non-ideal battery model ($k > 0$) is used. The intuition behind such a robustness property lies in the nature of the NLPs $\mathbf{P}_i$ in the previous section: observe that the solution depends on the values of $G_i(w)g_i(w)$ and the associated constraints (2.13)-(2.15), while the only effect of the parameter k enters through the inequalities $T_i^*(w) \leq T_j^*(w)$, $j \neq i$. Therefore, if a solution is obtained under $k = 0$ (a much easier problem which, as we have seen, can be reduced to a LP) and these inequalities are still satisfied when $k > 0$, then there is no need to re-solve the $\mathbf{P}_i$ NLPs. Naturally, when this property holds, the value of the resulting optimal network lifetime is generally different, but the actual routing policy remains unchanged.

Let $w^i(k)$ denote the solution of problem $\mathbf{P}_i$ when the KBM is invoked with parameter k, including the ideal batterty case $k = 0$. The corresponding node lifetimes are denoted by $T_i^*(w^i, k)$. The robustness property we identify rests on the following lemma, which provides simple sufficient conditions under which $w^i(0) = w^i(k)$ for any $k > 0$.

Lemma 3: Consider the NLP $\mathbf{P}_i$ with solution $w^i(k)$ under battery parameter $k \geq 0$. If the initial conditions for the node energies satisfy $B_j = R_j$ for all $j =$

$0, \ldots, N-1$, then

$$w^i(0) = w^i(k) \qquad \text{for any } k > 0 \tag{2.38}$$

Proof: Let $r_i^k(t)$, $b_i^k(t)$ denote the node i battery state variables under $k \geq 0$. When $k = 0$, (2.11) becomes $\dot{r}_i^0(t) = -G_i(w^i(0))g_i(w^i(0))$. Therefore, for any $j \neq i$ we have

$$\frac{\dot{r}_i^0(t)}{\dot{r}_j^0(t)} = \frac{G_i(w^i(0))g_i(w^i(0))}{G_j(w^i(0))g_j(w^i(0))} \tag{2.39}$$

When $k > 0$, let $z_i^k(t) = b_i^k(t) - r_i^k(t)$ and note that by subtracting (2.11) from (2.12) we have

$$\dot{z}_i^k(t) = G_i(w(t))g_i(w(t)) - 2kz_i^k(t)$$

Fixing the routing vector $w(t)$ to $w^i(0)$ and solving the differential equation above with initial condition $z_i^k(0) = B_i - R_i = 0$ by assumption, we get $z_i^k(t) = \frac{G_i(w^i(0))g_i(w^i(0))}{2k}(1 - e^{-2kt})$

Using this in (2.11), we have

$$\begin{aligned} &\dot{r}_i^k(t) = \\ &\quad -G_i(w^i(0))g_i(w^i(0)) + \frac{G_i(w^i(0))g_i(w^i(0))}{2}(1 - e^{-2kt}) \\ &\quad = -G_i(w^i(0))g_i(w^i(0))\frac{1 + e^{-2kt}}{2} \end{aligned} \tag{2.40}$$

Therefore,

$$\frac{\dot{r}_i^k(t)}{\dot{r}_j^k(t)} = \frac{G_i(w^i(0))g_i(w^i(0))}{G_j(w^i(0))g_j(w^i(0))}, \quad k > 0$$

which is identical to (2.39). Thus, under $k > 0$, the inequalities $T_i^*(w^i, k) \leq T_j^*(w^i, k)$ remain just as valid as $T_i^*(w^i, 0) \leq T_j^*(w^i, 0)$ under $k = 0$ and it follows that the solution $w^i(k)$ is unaffected relative to $w^i(0)$, completing the proof. ■

The next theorem is a direct consequence of Lemma 3:

Theorem 2: If the initial conditions for all node energies satisfy $R_i = B_i$,

$i = 0, \ldots, N-1$, then the optimal routing policy under an ideal battery model, $k = 0$, is unaffected when $k > 0$:

$$w^*(0) = w^*(k), \quad k > 0$$

Proof: By assumption, Lemma 3 applies to all nodes $i = 0, \ldots, N-1$, i.e., $w^i(0) = w^i(k)$. **Algorithm 1** gives $w^*(k)$ as the solution of the NLP $\mathbf{P}_i$ such that $\max_i \{T_i^*(w)\} = T_i^*(w^i(k))$ for some i for any $k \geq 0$. It then follows from Lemma 3 that $w^*(0) = w^i(0) = w^i(k) = w^*(k)$. ■

It should be noted that $B_j = R_j$ for all $j = 0, \ldots, N-1$ is a condition that is almost always automatically satisfied by Remark 1: when a battery is initialized at node j, it is normally "at rest", therefore $B_j = R_j$. From a practical standpoint, Theorem 2 implies that we can obtain $w^*(0)$ under the ideal battery model assumption using a simple LP (see (Ning and Cassandras, 2009)) and still rely on this solution even if the batteries are in fact non-ideal. Naturally, the resulting lifetimes are different, but the computational effort involved to derive an optimal routing policy is substantially reduced. Moreover, it makes the optimal routing policy *independent of the parameter* k, which is often difficult to estimate.

2.3.4 Simulation examples

In order to illustrate the results of our analysis, let us consider a 7-node network as shown in Fig. 2·2 where node coordinates are given next to each node. Nodes 1 and 7 are the source and base nodes respectively, while the rest are relay nodes. We set $C_s = 0.0001, C_f = C_r = 0.05$, and $\beta = 2$ in the energy model. The total initial energy is $R = 100$ and we assume all nodes have the same initial energy, so that $R_i = 16.67$, $i = 1, ..., 6$. We also set initial conditions for the KBM at all nodes so that $R_i = B_i$, $i = 1, ..., 6$. Table 2.1 shows the optimal routing probabilities for this network obtained through **Algorithm 1** when ideal batteries are used (this can be recovered in our analysis by setting $k = 0$ in applying **Algorithm 1**.) The optimal network lifetime in this case is 54.55 and Table 2.2 shows all node lifetimes under the optimal routing policy (we do not provide specific units in our examples, but, based on standard known data, distance units in feet and time units in months or weeks are reasonable.) Note that nodes 1-5 die virtually simultaneously, while the lifetime of node 6 is considerably longer. This is because energy consumption at each node depends on both the inflow rate to that node and the transmitting distances to other nodes. In this example, node 6 is located close to the base, hence using little energy in packet transmissions. In fact, by relocating node 6 to (120,120) and roughly doubling its distance from the base, it was observed that all 6 nodes die at the same time under the optimal policy. Another important observation in this example is that node 2 receives only 34% of the network inflow and this happens because there is no benefit to sending data packets to a relatively close relay node. The network topology in Fig.2·2 and all energy model parameter values are taken from an example in (Wu and Cassandras, 2005) where the routing problem was solved for the ideal battery case. Our results under $k = 0$ recover almost the same routing probabilities and the exact same lifetimes as in this example. Moreover, (Wu and Cassandras, 2005) contains a comparison of the WSN lifetime obtained here with the one obtained using a locally greedy policy, random routing, and the EAR policy in (Shah and Rabaey, 2002); it was shown that the former provides significant lifetime improvements over all three alternatives.

Table 2.1: Optimal routing probs., 7-node network, ideal batteries

w_{ij}	1	2	3	4	5	6
1	N/A	0.343073	0.656927	0	0	0
2	N/A	N/A	0.837081	0.000002	0	0.162917
3	N/A	N/A	N/A	0.971801	0	0.028199
4	N/A	N/A	N/A	N/A	0.929019	0.070981
5	N/A	N/A	N/A	N/A	N/A	1

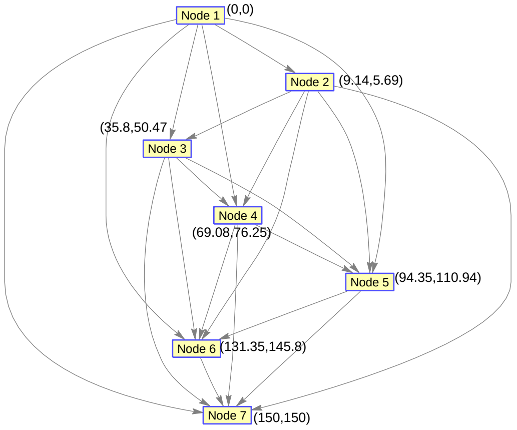


Figure 2·2: Network topology-1

Node i	1	2	3	4	5	6
Lifetime	54.553	54.554	54.557	54.554	54.555	122.055

Table 2.2: Lifetimes under routing policy given in Table 2.1

Next, we revisit the same problem with the KBM battery dynamics (2.11)-(2.12). Assuming $k = 0.001$ and using **Algorithm A1**, the optimal routing probabilities and node lifetimes are given in Tables 2.3 and 2.4 respectively. It is interesting to observe that even such a small value of k results in a lifetime improvement of approximately 3%, which is due to the recovery effect in the battery dynamics captured in (2.11)-(2.12). Tables 2.5 and 2.6 provide the resulting optimal routing probabilities and node lifetimes for two additional larger values of k, showing considerable network lifetime improvements.

Comparing Tables 2.1 and 2.3, note that the optimal routing probabilities for the ideal and non-ideal battery cases are virtually identical, thus confirming our result in Theorem 2 (whose conditions are satisfied in this example.) As a result, one can adopt in practice a simple ideal battery model, leading to a simple optimal routing solution through a LP as in (Chang and Tassiulas, 2004) and (Ning and Cassandras, 2009). Similar results are obtained for a symmetric network topology with the same positions for source and base nodes. As one would expect, all nodes die simultaneously due to this symmetry.

w_{ij}	1	2	3	4	5	6
1	N/A	0.343110	0.656890	0	0	0
2	N/A	N/A	0.837114	0	0	0.162886
3	N/A	N/A	N/A	0.971793	0	0.028207
4	N/A	N/A	N/A	N/A	0.929022	0.070978
5	N/A	N/A	N/A	N/A	N/A	1

Table 2.3: Optimal routing probs., 7-node network, *non-ideal* batteries ($k = 0.001$)

Node i	1	2	3	4	5	6
Lifetime	56.0697	56.0696	56.0695	56.0696	56.0697	129.795

Table 2.4: Lifetimes under routing policy given in Table 2.3 and $k = 0.001$

Next, we consider an example in which initial node energies are no longer identical, specifically: $R_1 = 20$, $R_2 = 17$, $R_3 = 14$, $R_4 = 11$, $R_5 = 8$, $R_6 = 5$, while still maintaining the condition $R_i = B_i$. We use the same network shown in Fig.2·2 and only shift the source node to the point (-15,-15). Using **Algorithm A1** the optimal routing probabilities and network lifetime for different values of k are shown in Tables 2.7 and 2.8 respectively. As expected, the robustness property identified in Theorem 2 still applies.

2.4 A joint optimal routing and initial energy allocation problem

In this section, we go a step beyond routing as a mechanism through which we can control the WSN resources by also controlling the allocation of initial energy over its nodes so as to maximize the lifetime. An application where this problem arises is in a network with rechargeable nodes. In this case, solving the joint optimal routing and

Routing Probability	$k = 0$	$k = 0.002$	$k = 0.01$
w_{12}	0.343073	0.343110	0.343110
w_{13}	0.656927	0.656890	0.656890
w_{14}	0	0	0
w_{15}	0	0	0
w_{16}	0	0	0
w_{23}	0.837081	0.837114	0.837113
w_{24}	0.000002	0	0.000001
w_{25}	0	0	0
w_{26}	0.162917	0.162886	0.162886
w_{34}	0.971801	0.971793	0.971793
w_{35}	0	0	0
w_{36}	0.028199	0.028207	0.028207
w_{45}	0.929019	0.929022	0.929022
w_{46}	0.070981	0.070978	0.070978
w_{56}	1	1	1
Lifetime	54.554539	57.635541	71.157489
Improvement(%)	N/A	5.65	30.43

Table 2.5: Optimal routing probabilities and network lifetime for a 7-node network (Fig.2·2)

Node i	1	2	3	4	5	6
$k = 0.002$	57.636	57.635	57.635	57.636	57.636	138.038
$k = 0.01$	71.157	71.157	71.157	71.157	71.158	195.12

Table 2.6: Lifetimes under routing policy given in Table 2.5

Routing Probability	$k = 0$	$k = 0.001$	$k = 0.01$
w_{12}	0.910030	0.910034	0.910034
w_{13}	0	0	0
w_{14}	0	0	0
w_{15}	0	0	0
w_{16}	0.089970	0.089966	0.089966
w_{23}	0.950300	0.950301	0.950301
w_{24}	0	0	0
w_{25}	0	0	0
w_{26}	0.049700	0.049699	0.049699
w_{34}	0.889337	0.889332	0.889332
w_{35}	0	0	0
w_{36}	0.110663	0.110668	0.110668
w_{45}	0.823208	0.823210	0.823210
w_{46}	0.176792	0.176790	0.176790
w_{56}	1	1	1
Lifetime	35.25	35.88	42.06
Improvement(%)	N/A	1.79	19.32

Table 2.7: Optimal routing probabilities and network lifetime for a 7-node network with different initial energies

i	1	2	3	4	5	6
$k = 0$	35.252	35.253	35.254	35.253	35.254	37.295
$k = 0.001$	35.882	35.882	35.882	35.882	35.882	36.616
$k = 0.01$	42.064	42.065	42.065	42.065	42.065	43.98

Table 2.8: Lifetimes under routing policy given by Table 2.7

initial energy allocation problem provides optimal recharging amounts maximizing the network lifetime which may not correspond to full charges for all nodes.

Let us define the total initial energy available as $\bar{R}$ and let $R = [R_0, \ldots, R_{N-1}]$. From Theorem 1, we know that the optimal routing policy is fixed unless the topology of the network changes. Then, we can formulate the following problem:

$$\max_{\substack{R_i, i=0,\ldots,N-1 \\ w_{ij}, j=1,\ldots,N-1}} T \tag{2.41}$$

$$\text{s.t. } T \leq T_i^*(w, R_i), \quad i = 0, \ldots, N-1$$

$$\sum_{i \prec j,\ j<N} w_{i,j} \leq 1,\ 0 \leq w_{i,j} \leq 1,\ \ i,\ j = 0, \ldots, N,\ i \prec j$$

$$0 < R_i < \min(B_i, \bar{R}), \qquad \sum_{i=0}^{N-1} R_i = \bar{R}$$

This is a NLP problem where the control variables are both the routing probabilities $w_{i,j}$ and the initial energies R_i. In this case, $T_i^*(w, R_i)$ is the solution of (2.24) for all $i = 0, \ldots, N-1$, which is now dependent on both w and R_i. Differentiating (2.24) with respect to R_i we get

$$\frac{1}{2} - \frac{G_i(w)g_i(w)}{2}\frac{\partial T}{\partial R_i} + k\left[B_i - \frac{G_i(w)g_i(w)}{2k}\right]e^{-2kT}\frac{\partial T}{\partial R_i} + \frac{1}{2}e^{-2kT} - kR_i e^{-2kT}\frac{\partial T}{\partial R_i} = 0$$

which yields:

$$\frac{\partial T}{\partial R_i} =$$

$$\frac{1}{2}(1+e^{-2kT})[\frac{G_i(w)g_i(w)}{2}(1+e^{-2kT})-k(B_i-R_i)e^{-2kT}]^{-1}$$

Observe that $\frac{\partial T}{\partial R_i} > 0$ if and and only if

$$B_i - R_i < \frac{G_i(w)g_i(w)}{2k}(1+e^{2kT}) \tag{2.42}$$

Recalling Remark 1, we may assume that $B_i = R_i$ since all batteries are normally initialized at an equilibrium state. In this case, (2.42) holds. Otherwise, (2.42) becomes a condition we need to impose so as to ensure that $\frac{\partial T}{\partial R_i} > 0$ which will be used in the result which follows.

If the solution of problem (2.41) is (w^*, R^*), then $T_i^*(w^*, R_i^*)$ is the solution of (2.24) under this routing vector and initial energy at node i. The following theorem establishes the fact that this optimal solution is such that all nodes deplete their energy at the same time.

Theorem 3: If condition (2.42) holds, the solution of problem (2.41) satisfies

$$T^* = T_0^*(w^*, R_0^*) = T_1^*(w^*, R_1^*) = \cdots = T_{N-1}^*(w^*, R_{N-1}^*) \tag{2.43}$$

Proof: We use a contradiction argument. Let us assume that under the optimal policy (w^*, R^*) not all nodes die together. We then define the following two index sets:

$$S_1 = \{i : T_i^*(w^*, R_i^*) = T^*\}, \quad S_2 = \{i : T_i^*(w^*, R_i^*) > T^*\}$$

According to our assumption, S_2 is not empty and let $j = \arg\min_{i\in S_2}\{T_i^*(w^*, R_i^*)\}$ i.e., node j is the first one to die after time T^* and for all $i \in S_1$ we have $T^* = T_i^*(w^*, R_i^*) < T_j^*(w^*, R_j^*)$ (if there are two or more nodes with the same value $T_j^*(w^*, R_j^*)$, then we select any one of them.) Keeping the routing vector to its optimal value w^*, we then perturb the energy allocation vector R^* to a new vector R' as follows:

$$
\begin{aligned}
&R'_i = R^*_i + \epsilon, \ \text{ for all } i \in S_1 \\
&R^{'}_j = R^*_j - |S_1|\epsilon \\
&R^{'}_k = R_k, \text{ for all } k \in S_2, \ k \neq j
\end{aligned}
$$

where $\epsilon > 0$ is sufficiently small to ensure $R'_j > 0$. Since $\sum_{i=0}^{N-1} R^{'}_i = \bar{R}$, it follows that (w^*, R') is a feasible policy. Under this policy, the node lifetimes are given by $T^{'}_i = T^*_i(w^*, R'_i)$, the solution of (2.24) under (w^*, R'). Since we have shown that $\frac{\partial T}{\partial R_i} > 0$ under (2.42), we have

$$
T^{'}_i = \begin{cases} T^*_i(w^*, R^*_i) + f_i(\epsilon) & \text{if } i \in S_1 \\ T^*_i(w^*, R^*_i) - f_i(\epsilon)|S_1| & \text{if } i = j \\ T^*_i(w^*, R^*_i) & \text{otherwise} \end{cases}
$$

where $f_k(x)$ is a continuous function such that $f_k(x) \geq 0$ and $f_k(0) = 0$. Since $f_k(x)$ is continuous, we can find a small enough $\epsilon > 0$ and hence $f_i(\epsilon)$ to guarantee that

$$
T^* = T^*_i(w^*, R^*_i) < T^{'}_i < T^{'}_j, \quad \text{for all } i \in S_1
$$

and the lifetime under (w^*, R') is $T^{'} = \min_{i \in S_1}\{T^{'}_i\} > T^*$.

Thus, by choosing a small enough $\epsilon > 0$, the network lifetime under (w^*, R') is larger than under (w^*, R^*) which contradicts the optimality of (w^*, R^*). Therefore, we conclude that S_2 must be empty, which implies (2.43). ■

Remark 3: In order to guarantee (2.43), it is necessary that $T^*_i(w^*, R^*_i) < \infty$. Looking at (2.24) and recalling that $g_i(w) > 0$, this is equivalent to assuming that $G_i(w) > 0$, i.e., that no node is left unutilized.

Based on Theorem 3, we can simplify the NLP problem (2.41). In particular, we solve it in two steps. In Step 1, assuming a fixed routing policy w, we determine the corresponding optimal initial energy distribution policy by solving the set of equations:

$$\begin{aligned} & T_0^*(w,R_0) = T_1^*(w,R_1) = \cdots = T_{N-1}^*(w,R_{N-1}) \\ \text{s.t.} \quad & \sum_{i=0}^{N-1} R_i = \bar{R} \end{aligned} \tag{2.44}$$

Its solution is defined to be $R^*(w)$ with an associated lifetime $T^*(w)$. Then, in Step 2 we search over the feasible set of w given by (2.15) to determine the optimal $T^*(w)$ by using a standard nonlinear optimization solution procedure. We should point out, however, that solving problem (2.44) to obtain parametric solutions for $T^*(w)$ and $R^*(w)$ is not a simple task and common solvers fail to accomplish it. Instead, we can proceed by selecting one of the parametric equations for $T_i^*(w,R_i)$ as an objective function and add (2.44) as constraints to a new NLP problem below, whose solution we can obtain with standard optimization solvers:

$$\max_{R_i, w_{ij}, j=1,\ldots,N-1} \; T_i^*(w,R_i) \tag{2.45}$$

$$\text{s.t. } T_i^*(w,R_i) - T_j^*(w,R_j) = 0 \;\; i,j = 0,\ldots,N-1, i \neq j$$

$$\sum_{i \prec j,\, j<N} w_{i,j} \leq 1, \; 0 \leq w_{i,j} \leq 1, \; i, \; j = 0,\ldots,N, \; i \prec j$$

$$0 < R_i < \min(B_i, \bar{R}), \qquad \sum_{i=0}^{N-1} R_i = \bar{R}$$

Remark 4: As in Section 2.3, our analysis can recover the ideal battery case by setting $k = 0$ in (2.11)-(2.12), which implies that $T_i^*(w) = R_i \left[G_i(w) g_i(w)\right]^{-1}$. This simplifies the solution of (2.44) as follows. Setting $K_i(w) = \left[G_i(w) g_i(w)\right]^{-1}$, (2.44) implies that

$$R_i = \frac{K_0(w)}{K_i(w)} R_0, \quad i = 1,\ldots,N-1$$

$$R_0 = \bar{R}\left[1+\sum_{i=1}^{N-1}\frac{K_0(w)}{K_i(w)}\right]^{-1} = \frac{\bar{R}}{K_0(w)}\left[\sum_{i=0}^{N-1}\frac{1}{K_i(w)}\right]^{-1}$$

and it follows that

$$R_i^*(w) = \frac{\bar{R}}{K_i(w)}\left[\sum_{j=0}^{N-1}\frac{1}{K_j(w)}\right]^{-1}, \quad i = 1,\ldots,N-1$$

Then, the lifetime $T^*(w)$ is given by

$$T^*(w) = K_0(w)R_0 = \bar{R}\left[\sum_{j=0}^{N-1}\frac{1}{K_j(w)}\right]^{-1} =$$
$$\bar{R}\left[\sum_{i=0}^{N-1}G_i(w)\left(\sum_{i\prec j,\ j<N} w_{i,j}k_{ij} + k_{i,N}\right)\right]^{-1}$$

Consequently, the solution of problem (2.41) is the same as that of the NLP problem:

$$\min_{w} \sum_{i=0}^{N-1}G_i(w)\left(\sum_{i\prec j,\ j<N} w_{i,j}k_{ij} + k_{i,N}\right)$$
$$\text{s.t.} \quad 0 \le w_{i,j} \le 1, \quad 0 \le i,j \le N \text{ and } i \prec j$$
$$\sum_{i\prec j,\ j<N} w_{i,j} \le 1,\ G_i(w) > 0$$

2.4.1 Simulation examples

In this section, we consider a numerical example for the joint optimal routing and initial energy allocation problem. As in section 2.3.4, first the problem is solved for a network with ideal node batteries and then using the KBM dynamics (2.11)-(2.12). Let us consider the same network as in Fig. 2·2 except we relocate the source node to (-15,-15). Table 2.9 shows the optimal routing probabilities and initial energies of all nodes under different vaklues of k, including the ideal battery case where $k = 0$

in (2.11)-(2.12). Note that the WSN lifetime with $k = 0$ is 63.33 which considerably exceeds the value 54.55 seen in section 2.3.4, even though the distance between the source and base nodes is larger in this case. Moreover, once again we observe that both optimal initial energies and routing probabilities are the same over different values of k. Finally, note the fact that the network lifetime coincides with all individual node lifetimes, which are the same by Theorem 3, provides a strong justification for the definition of network lifetime being that of the first node to deplete its energy.

Routing Probability	$k = 0.001$	$k = 0.002$	$k = 0.01$
w_{12}	1	1	1
w_{23}	1	1	1
w_{34}	1	1	1
w_{45}	1	1	1
w_{56}	1	1	1
R_1	9.57	9.57	9.57
R_2	23.53	23.53	23.53
R_3	17.55	17.55	17.55
R_4	18	18	18
R_5	22.7	22.7	22.7
R_6	8.65	8.65	8.65
Lifetime	65.3752	67.501	85.6695
Improvement(%)	3.23	6.59	35.27

Table 2.9: Optimal routing probabilities, initial battery energy and network lifetime for a 7-node network

2.5 Lifetime maximization problem under a more general nonlinear battery model

The results obtained under the KBM battery dynamics, pave the way for an investigation of the same problem using a more accurate model. Next we adopt a detailed dynamic battery model of which the KBM is a special case. Our goal is to investigate whether the results obtained under the KBM are still preserved or the relatively simple nature of the KBM is responsible for them. In what follows, we briefly review a linear state space model (Zhang and Shi, 2009) derived from the diffusion-based model (Rakhmatov and Vrudhula, 2001). A one-dimensional diffusion equation describing the concentration behavior inside a battery (Rakhmatov and Vrudhula, 2001) is given by:

$$\begin{aligned} J(x,t) &= -D\frac{\partial C(x,t)}{\partial x} \\ \frac{\partial C(x,t)}{\partial t} &= D\frac{\partial^2 C(x,t)}{\partial x^2} \end{aligned} \tag{2.46}$$

where $C(x,t)$ represents the electrolyte concentration at time t at a distance $x \in [0,\omega]$ from the electrode (the distance between anode and cathode is 2ω). $J(x,t)$ stands for the electrolyte flux at time t at distance x and D denotes a constant diffusion coefficient. Let the initial concentration be a constant C^*. As described in (Rakhmatov and Vrudhula, 2001), applying the following two boundary conditions:

$$D\frac{\partial C(x,t)}{\partial x}\bigg|_{x=0} = \frac{i(t)}{\nu AF} \quad , \quad D\frac{\partial C(x,t)}{\partial x}\bigg|_{x=\omega} = 0$$

where $i(t)$ is the battery load, A is the area of the electrode, F is Faraday's constant, and ν is a scaling factor, the final solution for the concentration of electrolyte at the electrode ($x = 0$) is (using * to denote convolution):

$$C(0,t) = C^* - \frac{i(t)}{\nu\omega FA} * \left(1 + 2\sum_{m=1}^{\infty} e^{-\frac{\pi^2 m^2}{\omega^2}Dt}\right) \tag{2.47}$$

Defining $\rho(t) = 1 - \frac{C(0,t)}{C^*}$, at $t = 0$ we have $C(0,0) = C^*$ and $\rho(0) = 0$. Note that during discharge, $C(0,t)$ decreases, hence $\rho(t)$ increases. When the battery is depleted (electrolyte concentration reaches C_{cutoff}), $\rho(t)$ reaches the corresponding threshold $\rho_{cutoff} = (1 - C_{cutoff}/C^*)$. In order to derive a state space realization as in (Zhang and Shi, 2009), we define $y(t) = \rho(t)/\rho_{cutoff}$ which results in $y(0) = 0$ and $y(T) = 1$ at the failure time $t = T$. Replacing the infinite sum in (2.47) by a finite one with M terms, we obtain:

$$\begin{aligned} y(t) &= \frac{i(t)}{\alpha} * 1 + \frac{i(t)}{\alpha} * 2 \sum_{m=1}^{M} e^{-\delta_m t} \\ &= [1\ 1 ... 1] \begin{bmatrix} \dfrac{i(t)}{\alpha} * 1 \\ \dfrac{2i(t)}{\alpha} * e^{-\delta_1 t} \\ ... \\ \dfrac{2i(t)}{\alpha} * e^{-\delta_M t} \end{bmatrix} \end{aligned} \tag{2.48}$$

where $\delta_m = \pi^2 m^2 D/\omega^2$ and $\alpha = C^* \nu \omega F A \rho_{cutoff}$. Next, we define the state vector $\mathbf{x}(t) = [x_0(t), ..., x_M(t)]^T$ such that:

$$\begin{aligned} &\dot{x}_0(t) = \frac{1}{\alpha} i(t) \\ &\dot{x}_m(t) = \frac{2}{\alpha} i(t) - \delta_m x_m(t) \quad && m \in \{1, 2, ..., M\} \\ &x_m(0) = 0 && m \in \{0, 1, ...M\} \end{aligned} \tag{2.49}$$

which can be written as

$$\begin{aligned} &x_0(t) = \frac{i(t)}{\alpha} * 1 \\ &x_m(t) = \frac{2i(t)}{\alpha} * e^{-\delta_m t} \qquad && m \in \{1, 2, ..., M\} \end{aligned} \tag{2.50}$$

Substituting (2.50) into (2.48), we have:

$$y(t) = [1\ 1 ... 1] \begin{bmatrix} x_0(t) \\ x_1(t) \\ ... \\ x_M(t) \end{bmatrix} = [1\ 1 ... 1] \mathbf{x}(t) \tag{2.51}$$

For each node $i = 0, \dots, N-1$, $y_i(t)$ is the battery status indicator at time t. Setting $y_i(0) = 0$, it follows that $y_i(T) = 1$ which indicates that the battery is out of charge at the failure time $t = T$.

We consider the energy consumption model as in Section 2.2.3. Similarly, the workload of node 0, $u_0(t)$, is given by

$$u_0(t) = G_0(w)\left[\sum_{0\prec j, j<N} w_{0,j}(t)k_{0,j} + k_{0,N}\right] \tag{2.52}$$

where $G_0(w) = 1$. Also, for any node $i = 1, \ldots, N-1$, where we must include the energy for both receiving and transmitting data packets, we can show that:

$$u_i(t) = G_i(w)\left[\sum_{i\prec j, j<N} w_{i,j}(t)k_{i,j} + k_{i,N}\right] \tag{2.53}$$

Defining $g_i(w) = \sum_{i<j,j<N} w_{i,j}(t)k_{i,j} + k_{i,N}$ the dynamic model (2.49) and (2.51) for nodes $i = 0, \ldots, N-1$ becomes:

$$\begin{aligned}
&\dot{\mathbf{x}}_i(t) = A_i\mathbf{x}_i(t) + b_iG_i(w(t))g_i(w(t)) \\
&y_i(t) = c\mathbf{x}_i(t) && (2.54) \\
&A_i = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ 0 & -\delta_1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & -\delta_M \\ & & & \end{bmatrix} && (2.55) \\
&\quad = diag[0, -\delta_1, ..., -\delta_M]_{(M+1)\times(M+1)} \\
&b_i = [\frac{1}{\alpha}, \frac{2}{\alpha}, ..., \frac{2}{\alpha}]^T \quad c = [1\ 1...1]_{1\times(M+1)} && (2.56)
\end{aligned}$$

This is a more general, high-dimensional model compared to the KBM with only two state equations. Note that in the KBM, k is a crucial parameter modeling the “recovery effect” in the battery dynamics, similar to the role that the D parameter plays in (2.46).

Note that we consider identical battery characteristics for all nodes in the network, i.e. $A_i = A_j, b_i = b_j$ for all $i, j = 0, ..., N-1$ (we will discuss the reason for this assumption later in Remark 5). The vectors $\mathbf{x}_i(t) = [x_{i0}, ...x_{iM}]^T$ for $i = 0, ..., N-1$

define the state variables for our problem. Finally, observe that by controlling the routing probabilities $w_{i,j}(t)$ in (2.52) and (2.53) we directly control node i's battery discharge process.

2.5.1 Optimal control problem formulation

Our objective is to maximize the WSN lifetime by controlling the routing probabilities $w_{i,j}(t)$. The maximum lifetime optimal control problem is formulated as follows:

$$\min_{w(t)} -\int_0^T dt \tag{2.57}$$

$$\text{s.t.} \quad \text{for} \quad i = 0, ..., N-1$$

$$\dot{\mathbf{x}}_i(t) = A\mathbf{x}_i(t) + bG_i(w(t))g_i(w(t)) \tag{2.58}$$

$$y_i(t) = c\mathbf{x}_i(t)$$

$$A = diag[0, -\delta_1, ..., -\delta_M]_{(M+1)\times(M+1)}$$

$$b = [\frac{1}{\alpha}, \frac{2}{\alpha}, ..., \frac{2}{\alpha}]^T \quad c = [1\ 1...1]_{1\times(M+1)}$$

$$G_i(w(t)) = \sum_{k\prec i} w_{k,i}(t)G_k(w(t)) \tag{2.59}$$

$$G_0(w(t)) = 1$$

$$g_i(w(t)) = \sum_{i\prec j, j<N} w_{i,j}(t)k_{ij} + k_{i,N} \tag{2.60}$$

$$\sum_{i\prec j, j<N} w_{i,j}(t) \leq 1, \quad 0 \leq w_{i,j}(t) \leq 1 \tag{2.61}$$

$$\min_{i=0,...,N-1} y_i(T) = 1 \tag{2.62}$$

where $\mathbf{x}_i(t) = [x_{i0}, ..., x_{iM}]^T$ are the state variables representing node i's battery dynamics for $i = 0, \ldots, N-1$ and $y_i(t) = \sum_{j=0}^{M} x_{ij}(t)$ is the battery status indicator at node i. Control constraints are specified through (2.61), where the first inequality follows from the fact that $\sum_{i\prec j<N} w_{i,j}(t) + w_{i,N}(t) = 1$. Finally, (2.62) provides

boundary conditions for $\mathbf{x}_i(t)$, $i = 0, \ldots, N-1$, at $t = T$ requiring that the terminal time is the earliest instant when $y_i(t) = \sum_{j=0}^{M} x_{ij}(t) = 1$ for any node i (recall that $y_i(T) = 1$ indicates battery depletion). In other words, at $t = T$ we require that the maximal value over all $\{y_0(T), \ldots, y_{N-1}(T)\}$ is 1 or, equivalently, $T = \inf_{t \geq 0}\{t : y_i(t) = 1$ for at least some $i = 0, \ldots, N-1\}$.

This is a classic minimum (maximum) time optimal control problem except for two complicating factors: (*i*) The boundary condition (2.62) which involves the non-differentiable min function, and (*ii*) The control constraints (2.61). In what follows, we will use $w^*(t)$ to denote the optimal routing vector, which provides a (not necessarily unique) solution to this problem.

2.5.2 Optimal control problem solution

Our analysis is similar to that in Section 2.3, but it is complicated by the high-dimensional dynamics in (2.58). We begin with the Hamiltonian for this optimal control problem:

$$\begin{aligned} H(w, t, \lambda) &= -1 + \sum_{i<N} [\lambda_{i0}\dot{x}_{i0} + \lambda_{i1}\dot{x}_{i1} + \ldots + \lambda_{iM}\dot{x}_{iM}] \\ &= -1 + \sum_{i<N} [\lambda_{i0}\frac{1}{\alpha}G_i(w(t))g_i(w(t)) + \ldots + \lambda_{iM}\big(\frac{2}{\alpha}G_i(w(t))g_i(w(t)) - \delta_M x_{iM}\big)] \end{aligned} \tag{2.63}$$

where $\lambda_{i0}(t), \ldots, \lambda_{iM}(t)$ are the costates corresponding to $x_{i0}(t)$,..., $x_{iM}(t)$ at node i, which must satisfy

$$\begin{cases} \dot{\lambda}_{i0}(t) = -\frac{\partial H}{\partial x_{i0}} = 0 \\ \dot{\lambda}_{im}(t) = -\frac{\partial H}{\partial x_{im}} = -\delta_m \lambda_{im}(t) \quad m = 1, \ldots, M \end{cases} \tag{2.64}$$

Due to the nature of the state boundary conditions in (2.62), it is hard to derive explicit expressions for the costates $\lambda_{ij}(t)$. Thus, similar to our procedure in Section 2.3, we proceed by considering each possible case of a node dying first, which we will refer to as "scenario S_i" under which $1 = y_i(T) \geq y_j(T)$, $j \neq i$ for some fixed node i.

2.5.3 Analysis of scenario S_i

Under S_i, we have the terminal time constraints $y_i(T) = 1$ and $y_j(T) \leq 1$ for all $j \neq i$. Consequently, all $y_j(t)$, hence $\mathbf{x}_j(t)$, $j \neq i$, are unconstrained at $t = T$. The next theorem establishes the property that, under a fixed network topology, there exists a static optimal routing policy, i.e., there exists a vector $w^*(t)$ which is time invariant.

Theorem 4: If $1 = y_i(T) \geq y_j(T)$, $j \neq i$, for some i and the network topology is fixed, i.e., $d_{ij}(t) = d_{i,j} =$ constant for all $i, j = 0, \ldots, N-1$, then there exists a time-invariant solution of (2.57)-(2.62): $w^*(t) = w^*(T)$.

Proof: To derive explicit expressions for $\lambda_{i0}(t)$,..., $\lambda_{iM}(t)$ it is necessary to use boundary conditions $\lambda_{i0}(T)$,..., $\lambda_{iM}(T)$. Since $0 \leq y_i(t) \leq 1$ for all i and $t \in [0, T]$, the optimal control problem under S_i is state-unconstrained except for $y_i(T) = \sum_{j=0}^{M} x_{ij}(T) = 1$. Thus, the terminal state constraint function $\Phi(\mathbf{x}_i(T), ..., \mathbf{x}_{N-1}(T))$ is reduced to $\sum_{j=0}^{M} x_{ij}(T)$ and the costate boundary conditions are given by:

$$\begin{cases} \lambda_{im}(T) = \nu \frac{\partial \Phi(\mathbf{x}_i(T),...,\mathbf{x}_{N-1}(T))}{\partial x_{im}} = \nu & m = 0, ..., M \\ \lambda_{jm}(T) = 0 \qquad j \neq i \quad m = 0, ..., M \end{cases}$$

where ν is an unspecified scalar constant. This allows us to solve the costate equations in (2.64) to obtain for $t \in [0, T]$:

$$\begin{cases} \lambda_{i0}(t) = \nu \\ \lambda_{im}(t) = \nu e^{-\delta_m (t-T)}, & m = 1, ..., M \\ \lambda_{jm}(t) = 0 \qquad j \neq i & m = 0, ..., M \end{cases} \tag{2.65}$$

Using (2.65) in (2.63), we can simplify the Hamiltonian as follows:

$$H(w,t,\lambda) = -1 + \lambda_{i0}\frac{1}{\alpha}G_i(w(t))g_i(w(t)) + \sum_{j=1}^{M}\lambda_{ij}(\frac{2}{\alpha}G_i(w(t))g_i(w(t)) - \delta_j x_{ij}) \tag{2.66}$$

Observe that the control variables $w_{i,j}(t)$ appear only in $G_i(w(t))$ and $g_i(w(t))$ in the problem formulation (2.57)-(2.62). Thus, we can set $U_i(t) = G_i(w(t))g_i(w(t))$, $i = 0,\ldots,N-1$ to be the effective control variables with $U_l \leq U_i(t) \leq U_u$, where $U_l \geq 0$ and U_u are, respectively, the lower bound and upper bound of $U_i(t)$ for all $t \in [0,T]$. Note that both are constant since their determination depends exclusively on (2.59), (2.60) subject to (2.61), independent of the states. In particular, they depend on the fixed network topology and the values of the energy parameters $k_{i,j}$, $k_{i,N}$ in (2.60). Applying the Pontryagin minimum principle to (2.66):

$$U_i^*(t) = \arg\min_{U_l \leq U_i(t) \leq U_u} H(U_i,t,\lambda^*)$$

implies that the optimal control is of bang-bang type:

$$U_i^*(t) = \begin{cases} U_u & \text{if } \nu < 0 \\ U_l & \text{if } \nu > 0 \end{cases} \tag{2.67}$$

Moreover, the optimal solution must satisfy the transversality condition $\left(\lambda^* \frac{d\Phi}{dt} + L\right)_{t=T} = 0$ where $L = -1$ and we have seen that $\Phi(\mathbf{x}_i(T),...,\mathbf{x}_{N-1}(T)) = \sum_{j=0}^{M} x_{ij}(T)$. Therefore,

$$-1 + \nu\sum_{j=0}^{M}\dot{x}_{ij}(T) = 0$$

and it follows that $\nu = 1/\dot{y}_i(T)$. Since $y_i(T) = 1$, $y_i(0) = 0$ and $0 < y_i(t) < 1$ for all $t \in [0,T)$, we have $\dot{y}_i(T) > 0$, therefore, $\nu > 0$. By (2.67), $U_i^*(t) = U_l$ for all $t \in [0,T]$. We conclude that the optimal control problem under S_i is reduced to the following optimization problem:

$$\min_{w(t)} G_i(w(t))g_i(w(t)) \tag{2.68}$$
$$\text{s.t.} \quad (2.59)-(2.61) \text{ and } 1 = y_i(T) \geq y_j(T), \quad j \neq i$$

When $t = T$, the solution of this problem is $w^*(T)$ and depends only on $y_j(T)$, $j \neq i$, and, as already argued, the fixed network topology and the values of the fixed energy parameters $k_{i,j}$, $k_{i,N}$ in (2.60). The same applies to any other $t \in [0, T)$, therefore, there exists a time-invariant optimal control policy $w^*(t) = w^*(T)$, which minimizes the Hamiltonian and proves the theorem. ■

Note that there may exist multiple optimal control policies, including some that may be time varying. Theorem 4 asserts that there is at least one which is time-invariant, i.e., $w^*(t) = w^*(T) = w^*$, and it remains to obtain the values of $w^*_{i,j}$, $i = 0, \ldots, N-2$ and $j = 1, \ldots, N-1$, by explicitly solving the optimization problem (2.68). This requires knowledge of all $y_i(t)$, $t \in [0, T]$ in order to determine the values of all $y_i(T)$ and hence identify the node i such that $1 = y_i(T) \geq y_j(T)$ and use the values of $y_j(T)$, $j \neq i$. This can be accomplished by solving the differential equations (2.54)-(2.56), whose initial conditions are given as $x_{im}(0) = 0$, $i = 0, ..., N-1$ and $m = 0, ..., M$, with $w(t) = w$, the unknown optimal routing vector. It is straightforward to obtain $x_{ij}(t)$ as follows:

$$\begin{aligned}
x_{i0}(t) &= \frac{1}{\alpha} G_i(w)g_i(w)t \\
x_{ij}(t) &= \frac{2}{\alpha\delta_j} G_i(w)g_i(w)(1 - e^{-\delta_j t}),\ j = 1, ..., M
\end{aligned}$$

Recall that $y_i(t) = \sum_{j=0}^{M} x_{ij}(t)$, the "critical time" T_i^* such that $y_i(T_i^*) = 1$ and

$0 < y_i(t) < 1$ for all $t \in [0, T_i^*)$ is the solution of the nonlinear equation in T:

$$\frac{1}{\alpha} G_i(w) g_i(w) T + \sum_{j=1}^{M} \frac{2}{\alpha \delta_j} G_i(w) g_i(w)(1 - e^{-\delta_j T}) = 1 \tag{2.69}$$

which we write as $T_i^*(w)$. Thus, we may rewrite the S_i optimization problem as follows

$$\begin{aligned} \mathbf{P}_i : \ & \min_w G_i(w) g_i(w) \\ & \text{s.t.} \quad (2.59) - (2.61), \quad T_i^*(w) \leq T_j^*(w), \quad j \neq i \end{aligned}$$

where $T_i^*(w)$ is the solution of (2.69) for all $i = 0, \ldots, N-1$. Note that $\mathbf{P}_i$ may not always have a feasible solution.

Based on our analysis thus far, if we focus on a fixed scenario S_i, the solution to the optimal control problem is simply the solution of the NLP problem $\mathbf{P}_i$. However, since we do not know which node will die first, determining the value of i such that $T_i^*(w) \leq T_j^*(w)$ for all $j \neq i$ requires solving *all* $\mathbf{P}_i$ problems and find the best policy among them. This is accomplished through the following algorithm, referred to as **A2**.

Algorithm A2

1. Solve problem $\mathbf{P}_i$ for $i = 0, ..., N-1$ to obtain $T_i^*(w)$.
2. Set $T_i^*(w) = -1$ if a problem is infeasible.
3. The optimal lifetime is given by $\max_i \{T_i^*(w)\}$ and the corresponding optimal policy w^* is the one obtained for the associated problem $\mathbf{P}_i$.

2.5.4 A Robustness Property of the Optimal Routing Policy

In this section, we show that the optimal routing vector w^* obtained through Algorithm **A2** is robust with respect to the diffusion coefficient constant, D. This is similar to the robustness property established in Lemma 3 and Theorem 2 in Section 2.3.3 where it is shown that the solution of problem $\mathbf{P}_i$ is robust with respect to the parameter k of the KBM in (2.8) and (2.9). Here, the intuition behind this property lies in the nature of the NLPs $\mathbf{P}_i$: observe that the solution depends on the values of $G_i(w)g_i(w)$ and the associated constraints (2.59)-(2.61), while the only effect of the parameter D enters through the inequalities $T_i^*(w) \leq T_j^*(w)$, $j \neq i$. Therefore, if a solution is obtained under $D = 0$ and these inequalities are still satisfied when $D > 0$, then the actual routing policy remains unchanged, while the value of the resulting optimal network lifetime is generally different. Let $w^i(D)$ denote the solution of problem $\mathbf{P}_i$ when the RVW model is invoked with parameter D, including the case $D = 0$. The corresponding node lifetimes are denoted by $T_i^*(w^i, D)$. The robustness property we identify rests on the following Theorem:

Theorem 5: The optimal routing policy under $D = 0$, is unaffected when $D > 0$, i.e.,

$$w^*(0) = w^*(D) \qquad \text{for any } D > 0 \tag{2.70}$$

Proof: Let $y_i^D(t)$ denote the battery status indicator of node i under $D \geq 0$. Recall that $\delta_m = \pi^2 m^2 D/\omega^2$, therefore, $\delta_m = 0$ when $D = 0$ and the state equations in (2.49) for node i become:

$$\begin{aligned} \dot{x}_{i0}(t) &= \frac{1}{\alpha} i(t) \\ \dot{x}_{im}(t) &= \frac{2}{\alpha} i(t) \quad m \in \{1, 2, ..., M\} \end{aligned}$$

Hence,

$\dot{y}_i^0(t) = \sum_{j=0}^{M} \dot{x}_{ij}(t) = G_i(w^i(0))g_i(w^i(0))\dfrac{(1+2M)}{\alpha}$. Therefore, for any $j \neq i$ we have

$$\frac{\dot{y}_i^0(t)}{\dot{y}_j^0(t)} = \frac{G_i(w^i(0))g_i(w^i(0))}{G_j(w^i(0))g_j(w^i(0))} \tag{2.71}$$

When $D > 0$, by fixing the routing vector $w(t)$ to $w^i(0)$ and solving the differential equation (2.54)-(2.56) with initial condition $\mathbf{x}_i(0) = 0$, we get

$$y_i^D(t) = \frac{1}{\alpha} G_i(w^i(0)) g_i(w^i(0)) t + \sum_{j=1}^{M} \frac{2}{\alpha \delta_j} G_i(w^i(0)) g_i(w^i(0)) (1 - e^{-\delta_j t}) \tag{2.72}$$

Recall that $\delta_m = \dfrac{\pi^2 m^2 D}{\omega^2}$, D is a constant multiplier in δ_j. Consequently we have

$$\dot{y}_i^D(t) = \frac{1}{\alpha} G_i(w^i(0)) g_i(w^i(0)) \left(1 + 2 \sum_{j=1}^{M} e^{-\delta_j t} \right)$$

Therefore,

$$\frac{\dot{y}_i^D(t)}{\dot{y}_j^D(t)} = \frac{G_i(w^i(0)) g_i(w^i(0))}{G_j(w^i(0)) g_j(w^i(0))}, \quad D > 0$$

which is identical to (2.39). Thus, under $D > 0$, the inequalities $T_i^*(w^i, D) \leq T_j^*(w^i, D)$ remain just as valid as $T_i^*(w^i, 0) \leq T_j^*(w^i, 0)$ under $D = 0$ and it follows that the solution $w^i(D)$ is unaffected relative to $w^i(0)$. Note that **Algorithm A2** gives $w^*(D)$ as the solution of the NLP $\mathbf{P}_i$ such that $\max_i \{T_i^*(w)\} = T_i^*(w^i(D))$ for some i for any $D \geq 0$. Hence, $w^*(0) = w^i(0) = w^i(D) = w^*(D)$. ■

Remark 5: It should be noted that the robustness property of the optimal solution may not be valid if nodes have different battery parameters, i.e., A_i, b_i in (2.55) and (2.56) are not all the same. However, the time-invariant nature of the optimal routing vector in Theorem 4 remains unaffected.

2.5.5 Optimal routing by solving a single NLP

Based on Theorem 4 (or Theorem 1 in Section 2.3), when the topology of the network is fixed, there is at least one optimal routing policy which is time-invariant. Now, by defining a new variable T as the network lifetime (the first node whose battery is depleted), we merge **Algorithm A2** (or Algorithm A1) into a single NLP problem which determines an optimal routing vector and the network lifetime at the same time as follows:

$$\max_{w} T \tag{2.73}$$
$$\text{s.t.} \quad (2.59)-(2.61), \quad T \leq T_i^*(w)$$

Note that $T_i^*(w)$ is the parametric solution of the node i lifetime based on the energy dynamics considered for the battery. We consider the following three cases:

1. For nodes with ideal battery dynamics, the energy consumption is directly proportional to the battery load, i.e., $\dfrac{\partial r_i(t)}{\partial t} = -i(t)$, $T_i^*(w) = \dfrac{R_i}{G_i(w)g_i(w)}$ where R_i is the initial energy of node i.

2. For the KBM, as we have already observed the battery dynamics are:

   $$\dot{r}_i(t) = -i_i(t) + k(b_i(t) - r_i(t))$$
   $$\dot{b}_i(t) = -k(b_i(t) - r_i(t))$$

   and the battery lifetime, $T_i^*(w)$, is the solution of the following equation:

   $$R_i - \frac{G_i(w)g_i(w)}{2}T - \frac{1}{2}[B_i - R_i - \frac{G_i(w)g_i(w)}{2k}](e^{-2kT} - 1) = 0$$

3. If we consider the diffusion model (2.54)-(2.56) to describe the battery dynamics, $T_i^*(w)$ is the solution of (2.69).

2.5.6 Simulation examples

In order to illustrate the results of our analysis, we consider the 7-node network shown in Fig. 2·2 where node coordinates are given next to each node. Nodes 1 and 7 are the source and base respectively, while the rest are relay nodes. We solve the problem for a 2-state model ($M = 1$) and set $C_s = 0.0001, C_f = C_r = 0.05$, and $\beta = 2$ in the energy model. We assume $\alpha_i = 40375$, $i = 1, ..., 6$ (Rakhmatov et al., 2002). Table

2.10 shows the optimal routing probabilities and network lifetime for different values for δ_m ($D = 0$ and $D > 0$) obtained through **Algorithm A2**. Note that D is a constant multiplier in δ_m. To validate the robustness property discussed in Theorem 2, we apply the routing vector obtained when $\delta_m = 0$ (column 2 of Table. 2.10), $w^*(0)$, to the $T_i(w)$ equations with $\delta_m = (0.273)^2 m^2$ to see if it results in the same network lifetime of 80723.17 (we do not provide specific units, but, based on standard known data, distance units in feet and time units in minutes are reasonable for RVW model). Table 2.11 shows node lifetimes under $w^*(0)$ when $\delta_m = (0.273)^2 m^2$. It is observed that adopting $w^*(0)$, node 1 dies first and the network lifetime is equal to that obtained by solving the NLP problem (2.73) when $\delta_m = (0.273)^2 m^2$. This illustrates the robustness property as expected.

Remark 6: We should point out that solving (2.69) to obtain a parametric solution for node i lifetime, $T_i(w)$, is a hard task when we consider the battery model with more than 2 state variables ($M > 1$). However, the robustness property of the optimal solution with respect to the diffusion coefficient, D, obtained in Theorem 4, allows us to find the optimal routing vector for the simpler case when $D = 0$ and the same routing vector is optimal for other cases with $D > 0$. Assuming $D = 0$ (consequently $\delta_m = 0$) in (2.49) and (2.51), we obtain a closed-form expression for the lifetime of node i as $T_i(w) = \dfrac{\alpha}{(1+2M)G_i(w)g_i(w)}$. We can then find the optimal routing vector for any value of M by solving a single NLP problem (2.73).

Reduction in computational complexity. In order to investigate the reduction in the computational effort needed to find the optimal routing probabilities using the proposed "single NLP" formulation, we solve (2.73) for all three battery dynamic models discussed in Section 2.5.5 and compare the CPU times with those needed when implementing **Algorithms A1** or **A2**. For the network in Fig. 2·3, we adopt the diffusion-based model ($\delta_m = (0.273)^2 m^2$), the KBM (with $k = 0.02$), and the ideal

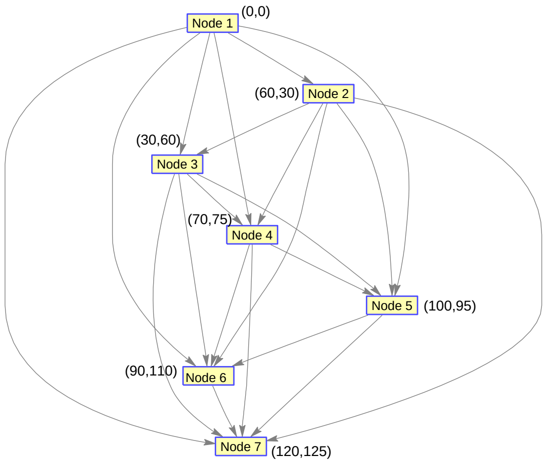


Figure 2·3: Network topology

Table 2.10: Optimal routing probabilities and network lifetime for a 7-node network with different diffusion coefficients

Routing Probability	(I) δ_m=0	(II) $\delta_m = (0.273)^2 m^2$
$w_{1,2}$	0.495974	0.506774
$w_{1,3}$	0.504026	0.493226
$w_{1,4}$	0	0
$w_{1,5}$	0	0
$w_{1,6}$	0	0
$w_{2,3}$	0.091704	0.132088
$w_{2,4}$	0.279738	0.15061
$w_{2,5}$	0.218011	0.334065
$w_{2,6}$	0.157214	0.173207
$w_{3,4}$	0.376457	0.183185
$w_{3,5}$	0.197358	0.407184
$w_{3,6}$	0.132955	0.218149
$w_{4,5}$	0.326819	0.34064
$w_{4,6}$	0.28053	0.300537
$w_{5,6}$	0.295149	0.418332
Lifetime	**26916.66**	**80723.17**

Table 2.11: Node Lifetimes under $w^*(0)$ when $\delta_m = (0.273)^2 m^2$

i	1	2	3
$T_i(w^*(0))$	80723.17	110347.25	100335.6
i	4	5	6
$T_i(w^*(0))$	303883.87	608865.92	550302.16

battery model. The corresponding CPU times are as shown in Table 2.12 where one can see that the new formulation offers a reduction in computation time of an order of magnitude or more, with the understanding that this reduction depends on the size of the network and its topology. Note that in order to obtain an optimal routing vector using **Algorithms A1** or **A2**, one should solve $(N-1)$ NLP problems.

Table 2.12: CPU time under different battery dynamics using Single NLP formulation compared to **Algorithm A1**

Battery Dynamics	Single NLP	**Algorithm A1**
Diffusion model	430.62 sec	28098 sec
KBM	61.27 sec	450.12 sec
Ideal Battery	59.56 sec	500.16 sec

Remark 7: The extension to a network with multiple source nodes is straightforward. Let us assume a network with k source nodes, each with a data generation rate of u_k. Let us also assume that the source nodes do not act as relay nodes and that each node routes data to nodes which are closer to the base station. The rest of the analysis is the same as the problem with a single-source network. The optimal control problem remains as in (2.57)-(2.62) (or (2.10)-(2.16)) except that the inflow rate to each node becomes:

$$\begin{aligned} G_i(w(t)) &= u_i && \forall i \in N_s, \\ G_i(w(t)) &= \sum_{j \prec i} w_{j,i}(t) G_j(w(t)) && \forall i \neq N_s \end{aligned}$$

where N_s is the set of all source nodes. Beginning with the Hamiltonian and defining Scenario S_i as we did in Section 2.3, one can show that there exists a time-invariant optimal routing policy for networks with multi-source nodes and fixed topology.

2.5.7 Joint optimal routing and initial energy allocation

In this section, we incorporate the diffusion-based battery dynamics into the joint optimal routing and initial energy allocation problem. Unlike our analysis for this problem in Section 2.4, here the battery model works based on changes in electrolyte concentration, therefore, finding an optimal initial energy allocation for the nodes is equivalent to finding an optimal initial electrolyte concentration for each one. Consequently, we need to relate the battery residual energy to the equivalent electrolyte concentration. We assume a linear relationship as follows: since we consider identical batteries for all nodes, we define R_{nom} to be the rated energy of the battery. Then, for each node we have $R_i = SoC_i \cdot R_{nom}$ where SoC_i denotes the "state of charge" of node i. One of the methods used to find the SoC of a battery is by measuring the specific gravity (SG) of its electrolyte. For example, for a lead-acid battery, as the SoC decreases through discharge, sulfuric acid is consumed and its concentration in water decreases. Consequently, the SG of the solution is reduced in direct proportion to the SoC (mpo, 2005). We assume a linear relationship between SoC and SG such that $SoC_i = a \cdot SG_i + b$ where a and b can be calculated based on available SoC vs SG lookup tables. Note that the electrolyte concentration is proportional to the SG of the solution, i.e., $SG_i = \sigma \cdot C_i$ where C_i stands for the electrolyte concentration at node i and σ is a constant coefficient which can be calculated based on the molecular wight of the electrolyte and mass percent of the solution. Finally, initial energy is a linear function of the initial electrolyte concentration:

$$R_i = m \cdot C_i + n \tag{2.74}$$

where $m = R_{nom} a \sigma$ and $n = R_{nom} b$. Let us define the total initial energy available as $\bar{R}$ and let $R = [R_0, \ldots, R_{N-1}]$. Using (2.74), we define corresponding terms for electrolyte concentrations as $\bar{C}$ and $C = [C_0, \ldots, C_{N-1}]$. From Theorem 1, we know that the optimal routing policy is fixed unless the topology of the network changes.

Then, we can formulate the following problem:

$$\max_{\substack{C_i, i=0,\dots,N-1\\ w_{i,j}, j=1,\dots,N-1}} \quad T \tag{2.75}$$

$$\text{s.t. } T \leq T_i^*(w, C_i), \quad i = 0, \dots, N-1$$

$$\sum_{i \prec j,\ j<N} w_{i,j} \leq 1,\ 0 \leq w_{i,j} \leq 1,\ \ i,\ j = 0, \dots, N,\ i \prec j$$

$$-\frac{n}{m} < C_i < \frac{R_{nom} - n}{m}, \quad \sum_{i=0}^{N-1} C_i = \bar{C}$$

$$\bar{C} = \frac{\bar{R} - Nn}{m}$$

This is a NLP problem where the control variables are both the routing probabilities $w_{i,j}$ and the initial concentrations C_i for nodes $i = 0, ..., N-1$. Looking at (2.74), the constraints on C_i above are to ensure that the equivalent R_i stays between 0 and R_{nom} and that $\sum_{i=0}^{N-1} R_i = \bar{R}$. In this case, $T_i^*(w, C_i)$ is the solution of (2.69) for all $i = 0, \dots, N-1$, which is now dependent on both w and C_i. Recalling that $\alpha_i = \nu\omega F A \rho_{cutoff} C_i$, we observe that unlike the problem discussed in the previous section, α is not identical for all nodes in the network. Differentiating (2.69) with respect to α_i we get

$$G_i(w) g_i(w) \frac{\partial T}{\partial \alpha_i} + \sum_{j=1}^{M} \left(2 G_i(w) g_i(w) \frac{\partial T}{\partial \alpha_i} e^{-\delta_j T} \right) = 1$$

which yields:

$$\frac{\partial T}{\partial \alpha_i} = \frac{1}{G_i(w) g_i(w) + 2 \sum_{j=1}^{M} G_i(w) g_i(w) e^{-\delta_j T}} > 0$$

Observe that $\frac{\partial T}{\partial C_i} = \frac{\partial T}{\partial \alpha_i} \cdot \frac{\partial \alpha_i}{\partial C_i} = \nu\omega F A \rho_{cutoff} \frac{\partial T}{\partial \alpha_i}$ which results in $\frac{\partial T}{\partial C_i} > 0$.

If the solution of problem (2.75) is (w^*, C^*), then $T_i^*(w^*, C_i^*)$ is the solution of (2.69) under this routing vector and initial electrolyte concentration at node i. The following theorem establishes the fact that this optimal solution is such that all nodes deplete their energy at the same time.

Theorem 6: The solution of problem (2.75) satisfies

$$T^* = T_0^*(w^*, C_0^*) = T_1^*(w^*, C_1^*) = \cdots = T_{N-1}^*(w^*, C_{N-1}^*) \tag{2.76}$$

Proof: The proof is similar to the same problem using the KBM battery model (see proof of Theorem 3 in Section 2.4). The critical fact needed in the proof is $\frac{\partial T}{\partial C_i} > 0$ (replacing $\frac{\partial T}{\partial R_i} > 0$ in the proof of Theorem 3). ■

Remark 8: In order to guarantee (2.76), it is necessary that $T_i^*(w^*, C_i^*) < \infty$. Looking at (2.69) and recalling that $g_i(w) > 0$, this is equivalent to assuming that $G_i(w) > 0$, i.e., that no node is left unutilized.

Similar to the way we proceeded in Section 2.4 and in view of Theorem 6, we can simplify the NLP problem (2.75) by selecting one of the parametric equations for $T_i^*(w, C_i)$ as an objective function and add (2.76) as constraints to a new NLP problem below, whose solution we can obtain with standard optimization solvers:

$$\max_{C_i, w_{i,j}, j=1,\ldots,N-1} \quad T_i^*(w, C_i) \tag{2.77}$$

$$\text{s.t. } T_i^*(w, C_i) - T_j^*(w, C_j) = 0 \ \ i, j = 0, \ldots, N-1, i \neq j$$

$$\sum_{i \prec j,\ j<N} w_{i,j} \leq 1,\ 0 \leq w_{i,j} \leq 1,\ i,\ j = 0, \ldots, N,\ i \prec j$$

$$-\frac{n}{m} < C_i < \frac{R_{nom} - n}{m}, \quad \sum_{i=0}^{N-1} C_i = \bar{C}$$

2.5.8 Simulation examples

We provide a numerical example for the joint optimal routing and initial energy allocation problem using the network in Fig. 2·4 with node coordinates shown next to each node. We set $m = 43.75$, $n = -200$ in (2.74), $R_{nom} = 25$, $\bar{R} = 100$ ($\bar{C} = 29.71$) , $\alpha = 40375$, $\delta_m = 0.273^2 m^2$ and other numerical values as before. Table 2.13 shows the optimal routing probabilities and initial energies of all nodes. Note that the WSN lifetime for this case is 98353 which is equal to the network lifetime when we consider all batteries initially fully charged ($R_i = R_{nom}, i = 1, ..., N-1$) and we just control the routing vector as discussed in Section 2.5.6. However, here we observe that only the source node needs a fully charged battery. Finally, the fact that the network lifetime coincides with all individual node lifetimes (as expected by Theorem 6), provides a strong justification for the definition of network lifetime as the time when the first node depletes its energy.

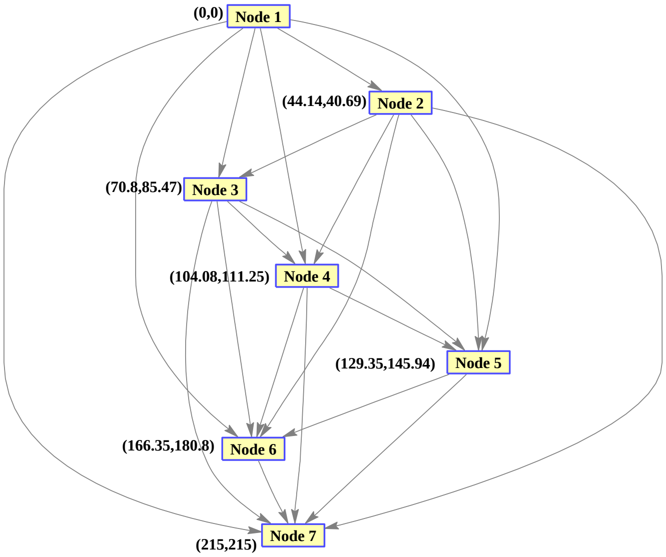


Figure 2·4: Network topology

2.6 Network Performance Under Security Threats

In this section we compare the WSN's performance under our optimal routing policy and the probabilistic routing policy introduced in (Shah and Rabaey, 2002) when a cyber-attack takes place. We limit ourselves to an example aimed at simply illustrating the advantages of the optimal routing policy we have derived for a specific

Table 2.13: Optimal routing probs., 7-node network, *non-ideal* batteries

$w_{i,j}$	1	2	3	4	5	6
1	N/A	1	0	0	0	0
2	N/A	N/A	0.9784	0.0154	0.0041	0.0015
3	N/A	N/A	N/A	0.7978	0.1914	0.0097
4	N/A	N/A	N/A	N/A	0.7082	0.2879
5	N/A	N/A	N/A	N/A	N/A	0.8325
R_i	25	16.6	14	14	14.4	16

form of attack. In (Shah and Rabaey, 2002) an Energy Aware Routing (EAR) policy is proposed in which a number of suboptimal paths are probabilistically selected with the intent of extending the network lifetime by spreading the traffic and forcing nodes in the network to deplete their energy at the same time. In EAR, each node builds a cost information table and propagates local cost information to other nodes. Costs are determined by the residual energy of each node and by the distances between them. Each node also maintains a routing probability table determined by local cost information. In this method, the routing probabilities are set periodically. At the beginning of each period, the routing probabilities are computed recursively as follows:

$$w_{i,j} = \frac{C_{ij}^{-1}}{\sum_{k \in O\{i\}} C_{ik}^{-1}} \tag{2.78}$$

$$C_{ij} = d_{ij}^{k_1} r_j^{k_2} + \mathcal{C}_j \quad \text{for all } j \in O\{i\} \tag{2.79}$$

$$\mathcal{C}_i = \sum_{k \in O\{i\}} w_{i,j} C_{ij} \tag{2.80}$$

where $w_{i,j}$ is the routing probability on the edge (i, j), C_{ij} is the cost of sending a data packet from node i to the destination via node j and $\mathcal{C}_i$ is the average cost of sending a packet from node i to the base station (Note that $\mathcal{C}_N = 0$ where N is the base station). Moreover, r_j is the residual energy of node j and k_1 and k_2 are weighting factors which can be chosen to find the minimum energy path or the path with the most energy or a combination of the above (Shah and Rabaey, 2002). Since the EAR method works based on the residual battery energy assuming ideal battery dynamics, we likewise use the same settings and determine the optimal routing vector and the network lifetime assuming ideal battery dynamics, i.e., Case 1) of problem (2.73).

Consider the network topology shown in Fig. 2·3. Table 2.14 shows the optimal routing probabilities obtained by solving (2.73) under normal (no threat) conditions. Under this routing policy, the network lifetime is 33.33. Figure 2·5 shows the routing probability updates obtained using the EAR policy by computing routing probabilities, $w_{i,j}$s, through (2.78)-(2.80) periodically when $k_1 = 5$ and $k_2 = 1$. Under the EAR routing policy, the network lifetime is 25.94. As expected, our optimal routing policy results in the longer lifetime compared to the EAR solution. Next, we investigate the network performance under a "sink-hole attack," one of the most severe routing attacks in sensor networks (Krontiris et al., 2008), for the two routing policies. Under a sink-hole attack, a compromised node broadcasts a fake low cost to the neighboring nodes, thus enticing all such nodes to route packets to it. We will assume an attacker uses the following strategy:

Table 2.14: Optimal routing probs., 7-node network, ideal batteries

$w_{i,j}$	1	2	3	4	5	6
1	N/A	0.3705	0.6295	0	0	0
2	N/A	N/A	0.1792	0.2045	0.1843	0.1960
3	N/A	N/A	N/A	0.2665	0.2046	0.3338
4	N/A	N/A	N/A	N/A	0.2133	0.3539
5	N/A	N/A	N/A	N/A	N/A	0.3627

1. The attacker compromises one node in the network randomly
2. At each time kT, where T is the updating period for the routing probabilities, the compromised node will (i) broadcast a fake near-zero cost ($\mathcal{C}_i$) to all

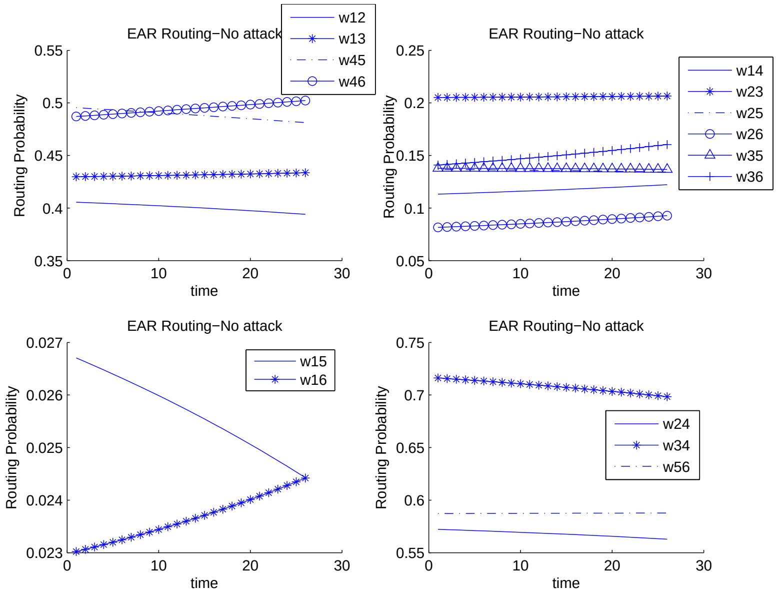


Figure 2·5: Routing probability updates under EAR policy

nodes with probability p to attract more flow; (ii) act as a normal node with probability $(1-p)$.

3. The compromised node corrupts all the packets it has received and forwards them to other nodes to deplete their energy.

In particular, we compare the network performance under the attack in terms of the normalized throughput (the ratio of the number of uncorrupted packets to the total number of packets) for the EAR and our optimal policy. Recall that in the EAR policy, each node i needs to know its neighbors' residual energies, r_j, and average costs, $\mathcal{C}_j$, $\forall j \in O_i$, to update its routing table. Thus, it is vulnerable to faked-cost-based attacks. We will further illustrate this through the same network in Fig. 2·3. Assume that node 2 is under sink-hole attack and that in each updating period it broadcasts faked-cost information to its neighbors with probability $p = 0.5$. Figure 2·6 shows how routing probability updates are affected in this scenario. Based

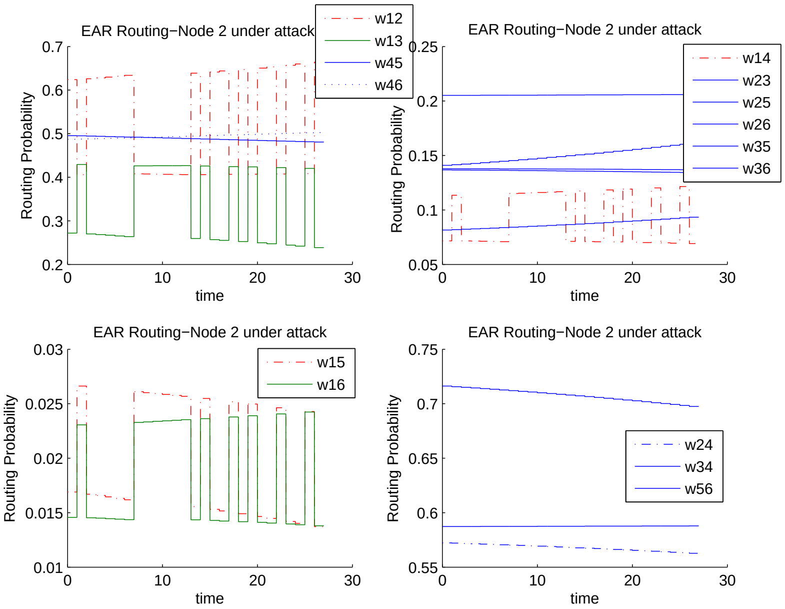


Figure 2·6: Routing probability updates under EAR policy when node 2 is under attack

on the network topology, node 1 is the only node that sends data to node 2. One can observe how routing probabilities from source node, node 1, to the other nodes, $[w_{12}\ w_{13}\ w_{14}\ w_{15}\ w_{16}]$, are affected at the periods in which node 2 broadcasts faked-cost data. On the other hand, our optimal policy uses the network topology to calculate routing probabilities and is robust with respect to this kind of attacks. However, the normalized throughput will be affected in both routing policies. Figure 2·7 shows the normalized throughput as a function of the probability of broadcasting faked-cost, p, when node 2 is under sink-hole attack. It can be observed that for this specific example, under our optimal policy the normalized throughput drops to 63%, but it is not sensitive to p. However, under the EAR policy it drops significantly as p increases. This happens because our routing policy is calculated based on the network topology and consequently robust with respect to p. Hence, the inflow rate to the compromised node as well as the normalized throughput, are not affected by the propagated faked-cost. On the other hand, in the EAR routing strategy, the data

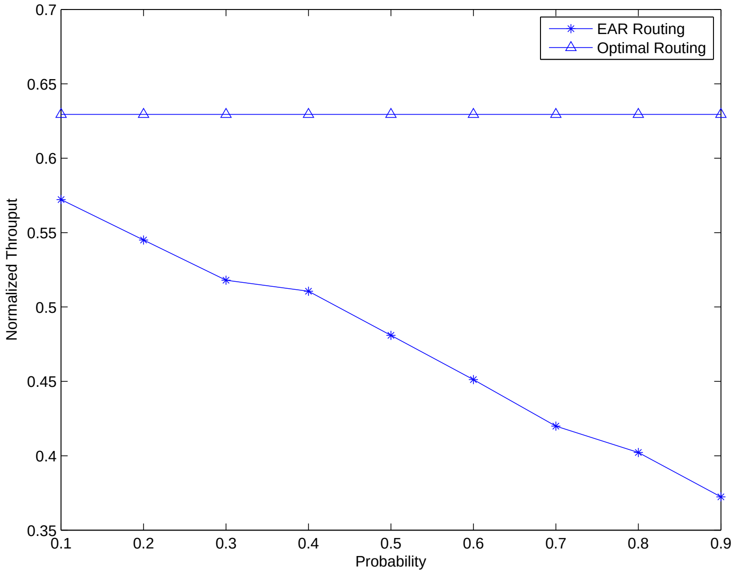


Figure 2·7: Normalized throughput vs prob. of broadcasting faked-cost

inflow rate to the compromised node increases with p which drops the normalized throughput correspondingly.

Remark 9: Depending on the network topology, it is possible that the optimal routing policy dictates all data packets to be routed through a specific node, i, which gives $G_i = 1$, (e.g., assume $w_{1,2} = 1$ in the previous examples). Under a sinkhole attack, if this node is the compromised one, the normalized throughput drops to zero. Clearly, this node should be a top priority in terms of protection against routing attacks. One way to address this problem is to purposely deviate from the optimal solution by routing a fraction q of data packets via node i and the remaining $1 - q$ through other nodes. This randomization-by-design degrades the network lifetime from its optimal value under normal operation (no attack), but protects the network against becoming completely useless when under attack by increasing its normalized throughput to $1 - q$.

2.7 Summary

We have shown that an optimal routing policy for minimizing a fixed topology sensor network's lifetime is time invariant even when the batteries used as energy sources for the nodes are modeled so as to take into account "non-ideal" phenomena such as the rate capacity effect and the recovery effect. The associated fixed routing probabilities may be obtained by solving a single Non-Linear Programming (NLP) problems In addition, under very mild conditions, this optimal policy is independent of the battery parameter. Therefore, one can resort to the case of ideal batteries where the optimal routing problem is much simpler to solve and can be reduced to a Linear Programming (LP) problem. We have also considered a joint routing and initial energy allocation problem over the network nodes with the same network lifetime maximization objective. In this case, the solution to this problem is given by a policy that depletes all node energies at the same time and the corresponding energy allocation and routing probabilities are obtained by solving an NLP problem.

Chapter 3

Lifetime Maximization for Wireless Sensor Networks with a Mobile Source Node

3.1 Introduction

In Chapter 2, we studied the lifetime maximization problem for a fully static wireless sensor network. This problem was previously studied in (Wu and Cassandras, 2005) and (Chang and Tassiulas, 2004) by defining the WSN lifetime as the time until the first node depletes its energy. We used the same definition in Chapter 2 and observed that it is often the case that an optimal policy controlling a static WSN's resources leads to individual node lifetimes being the same or almost the same as those of others, thus this definition is a good characterization of the overall network's lifetime in practice. We then relaxed the simplifying assumption of idealized batteries used as energy sources for nodes and more elaborate models were used to capture nonlinear dynamic phenomena that are known to occur in non-ideal batteries. A somewhat surprising result was that again an optimal policy exists which consists of time-invariant routing probabilities and that in fact this property is independent of the parameters of the battery model. However, this attractive property for routing is limited to a fixed network topology.

There are various ways to exploit WSN mobility and incorporating it into different network components. For instance, in (Wang et al., 2005) sink mobility is exploited for maximizing the network lifetime and a Linear Programming (LP) formulation is proposed in order to find the optimal sink node movement and sojourn time at different nodes in the network. In (Shah et al., 2003) mobile nodes (mules) are used to deliver data to the base station. In this chapter, we focus on the lifetime maximization problem in WSNs when source nodes are mobile. This situation frequently arises when a mobile sensor node is used to track one or more mobile targets or when there is a large area to be monitored that far exceeds the range of one or more static sensors.

Adding mobility to nodes raises several questions. First, one can no longer expect that a routing policy would be time invariant. Second, it is no longer reasonable to define the WSN lifetime in terms of the the first node depleting its energy. For instance, if a source node travels far from some relay nodes it was originally using, it is likely that it should no longer rely on them for delivering data to the base station. In this scenario, the network remains "alive" even when any or all of these relay nodes die. Thus, in view of node mobility, we need to revisit the definition of network lifetime. Finally, if a routing policy is time-varying, then it has to be re-evaluated sufficiently fast to accommodate the real-time operation of a WSN.

In the sequel, we consider mobility added to the source node and assume that any such node travels along a trajectory that it determines and which may or may not be known in advance. We limit ourselves to a single source node (the case of multiple mobile source nodes depends on the exact setting and is not addressed in this thesis). While on its trajectory, the source node continuously performs sensing tasks and generates data. Our goal is to derive an optimal routing scheme in order to maximize the network lifetime, appropriately redefined to focus on the mobile source node. Assuming first that the source node trajectory is not known in advance, we formulate three optimal control problems (OCPs) with differences in their terminal costs and terminal constraints and investigate how they compare in terms of the optimal routing policy obtained, total energy consumption, and the actual network lifetime. We will also limit ourselves to ideal battery dynamics for all nodes. However, adopting non-ideal battery models as in Chapter 2 does not change our analysis and only complicates the solution computation. We then consider the more challenging (from a computational perspective) problem where the source node's trajectory is known in advance, in which case this information can be incorporated into an optimal lifetime maximization policy.

In Section 3.2, we define the network model and the energy consumption model is presented in Section 3.3. In Section 3.4 we formulate the maximum lifetime optimization problem for a WSN with a mobile source node whose trajectory is not known in advance. Starting with a new definition for the network lifetime, we show that the solution is a sequence of Non-Linear Programming (NLP) problems along the source node trajectory. Numerical examples are included to illustrate our analytical results. In Section 3.5, we consider the case when the source node trajectory is known in advance and solve the corresponding optimal control problem using a standard numerical solver. We also compare lifetimes between this case and that of no a priori trajectory knowledge.

3.2 Network model

Consider a network with $N+1$ nodes where 0 and N denote the source and destination (base station) nodes respectively. Nodes $1, ..., N-1$ act as relay nodes to deliver data packets from the source node to the base station. We assume the source node is mobile and travels along a trajectory with a constant velocity while generating data packets which need to be transferred to the fixed base through static relay nodes. First, we assume the trajectory is not known in advance. Then, we discuss the case when the trajectory is known in Section 3.5. Except for the base station whose energy supply is not constrained, a limited amount of energy is available to all other nodes.

Let $r_i(t)$ be the residual energy of node i, $i = 0, \dots, N-1$, at time t. The dynamics of $r_i(t)$ depend on the battery model used at node i. Here, we assume ideal battery dynamics in which energy is depleted linearly with respect to the node's load, $U_i(t)$, i.e.,

$$\dot{r}_i(t) = -U(t) \tag{3.1}$$

The distance between nodes i and j at time t is denoted by $d_{i,j}(t)$. Since the source node is mobile, $d_{0,j}(t)$ is time-varying for all $j = 1, ..., N$. However, $d_{i,j}(t) = d_{i,j},\ i = 1, ..., N-1,\ j = 2, ..., N$ are treated as time-invariant with the assumption that the source node cannot be used as a relay, i.e., any node $i > 0$ must transfer data to other relay nodes $j > 0$, $j \neq i$ or directly to the base station node N. The source node can send data packets to any of the relay nodes as well as to the base station, while relay nodes can transmit/receive data packets to/from nodes in their transmission range. Let $O(i)$ and $I(i)$ denote the set of nodes to/from which node i can send/receive data packets respectively. Then, $O(i) = \{j : d_{i,j} \leq \tau_i\}$ and $I(i) = \{j : d_{j,i} \leq \tau_j\}$ where $\tau_i,\ i = 1, ..., N-1$ denotes the transmission range of node i. We define $w_{ij}(t)$ to be the routing probability of a packet from node i to node j at time t (equivalently, a data flow from i to j) and the vector $w(t) = [w_{ij}(t)]'$ defines the control in our problem. Let us also define $\mathbf{r}(t) = [r_0(t), ..., r_{N-1}(t)]$ as the vector of residual energies at time t. For simplicity, the data sending rate of source node 0 is normalized to 1 and let $G_i(w)$ denote the data packet inflow rate to node i. Given these definitions, we can express $G_i(w)$ through the following flow conservation equations:

$$G_i(w) = \sum_{k \in I(i)} w_{ki}(t) G_k(w),\ i = 1, \dots, N,\ G_0(w) = 1 \tag{3.2}$$

3.3 Energy Consumption Model

We use the energy consumption model similar to the model introduced for static networks in Chapter 2. Therefore, for any node $i = 1, \ldots, N-1$, the workload $U_i(t)$ at that node is given by

$$U_i(w(t)) = G_i(w(t))[\sum_{j \in O(i)} w_{ij}(t)(C_s d_{i,j}^{\beta}(t) + C_f) + C_r] \tag{3.3}$$

and the workload $U_0(t)$ at the source node 0 (recalling that $G_0(w(t)) = 1$) is given by

$$U_0(w(t)) = \sum_{j \in O(0)} w_{0j}(t)(C_s d_{0,j}^{\beta}(t) + C_f) \tag{3.4}$$

Assuming an ideal battery behavior for all nodes as in (3.1), the state variables for our problem are $r_i(t),\ i = 0, ..., N-1$. Note that $d_{0,j}(t) = \|(x_0(t), y_0(t)) - (x_j, y_j)\|$, the Euclidean distance of the source node from any other node is known at any time instant t (but not in advance) as determined by the source node's trajectory. Finally, observe that by controlling the routing probabilities $w_{ij}(t)$ in (3.3) and (3.4) we directly control node i's battery discharge process.

3.4 Optimal control problem formulation

Our objective is to maximize the WSN lifetime by controlling the routing probabilities $w_{ij}(t)$. For a static network, where all nodes including the source node are fixed, as we have already seen in Chapter 2, the network lifetime is usually defined as the time until the first node depletes its battery, i.e., $\min_{i=0,\ldots,N-1} r_i(T) = 0$ requiring that the terminal time is the earliest instant when $r_i(t) = 0$ for any node i (Chang and Tassiulas, 2004). However, when the source node is mobile, this definition of network lifetime is no longer appropriate as explained in Section 3.2. In the sequel, we formulate three optimal control problems for maximizing lifetime in a WSN with a mobile source node and investigate their relative effect in terms of an optimal routing policy, total energy consumption, and the network lifetime.

3.4.1 Optimal Control Problem - I

We define the network lifetime as the time when the source node runs out of energy. Consider a fixed time t_0 when the source node is at position $(x_0(t_0), y_0(t_0)) \in \mathbb{R}^2$. In the absence of any future information regarding the position of this node (e.g., the node may actually stop for some time interval before moving again), the routing problem we face is one of a fixed topology WSN similar to the one in (Wu and Cassandras, 2005) (equivalent to the problem discussed in Chapter 2 with non-ideal battery dynamics) but with different terminal state constraints due to the new network lifetime definition. Thus, this *instantaneous* maximum lifetime optimal control problem that the WSN faces at time t_0 is formulated as follows, using the variables defined in (3.2), (3.3) and (3.4):

$$\min_{w(t)} - \int_{t_0}^{T} dt \tag{3.5}$$

$$\text{s.t. } \dot{r}_i(t) = -U_i(w(t)),\ r_i(t_0) = R_i^{t_0},\ i = 0, .., N-1 \tag{3.6}$$

$$U_i(w(t)) = G_i(w(t))[\sum_{j \in O(i)} w_{ij}(t)(C_s d_{i,j}^2 + C_f) + C_r], \quad i = 1, ..., N-1 \tag{3.7}$$

$$U_0(w(t)) = \sum_{j \in O(0)} w_{0j}(t)(C_s d_{0,j}^2(t) + C_f) \tag{3.8}$$

$$d_{0,j}(t) = \|(x_0(t_0), y_0(t_0)) - (x_j, y_j)\|,\ x_0(t_0), y_0(t_0) \text{ given}$$

$$G_i(w(t)) = \sum_{k \in I(i)} w_{ki}(t) G_k(w(t)),\ i = 1, .., N-1 \tag{3.9}$$

$$\sum_{j \in O_i} w_{ij}(t) = 1, \quad 0 \leq w_{ij}(t) \leq 1,\ i = 0, \ldots, N-1 \tag{3.10}$$

$$r_0(T) = 0 \tag{3.11}$$

$$r_0(t) > 0,\ t \in [t_0, T);\ r_i(t) \geq 0,\ i = 1, .., N-1,\ t \in [t_0, T] \tag{3.12}$$

where $r_i(t)$, $i = 0, ..., N-1$, are the state variables representing the node i battery dynamics with the initial value of $R_i^{t_0}$ and $(x_0(t_0), y_0(t_0))$ are the given instantaneous coordinates of the source node at time t_0. Control constraints are specified through (3.10). Finally, (3.11) provides the boundary conditions for $r_0(t)$ at $t = T$ requiring that the terminal time is the time when the source node depletes its energy.

Since at time t_0 we do not have any knowledge about the future of the source node trajectory and, consequently, the network topology at $t > t_0$, we solve OCP-I at $t = t_0$ as if the topology were fixed to determine an *instantaneous* optimal routing vector. Then, we re-solve the problem for the new topology at $t = t_0 + \delta$. Thus, as the trajectory of the source node evolves, we discretize it using a constant time step δ and solve OCP-I at time instants $t_0 + k\delta$, $k = 0, 1, \ldots$. In what follows, we will use $w^*(t)$ to denote the optimal routing vector at any fixed time t.

Optimal control problem I solution

We begin with the Hamiltonian analysis for this optimal control problem (Bryson and Ho, 1975).

$$H(\mathbf{r}, x_0, y_0, w, \lambda, t) = -1 + \lambda_0(t)(-U_0(t)) + \sum_{i=1}^{N-1} \lambda_i(t)(-U_i(t)) \tag{3.13}$$

where $\lambda_i(t)$ is the costate corresponding to $r_i(t)$, $i = 0, ..., N-1$ and must satisfy:

$$\dot{\lambda}_i(t) = -\frac{\partial H}{\partial r_i} = 0 \quad i = 0, ..., N-1 \tag{3.14}$$

Therefore, λ_i, $i = 0, ..., N-1$, are constants. To determine their values we make use of the boundary conditions which follow from (3.11), i.e., the terminal state constraint function is $\Phi(\mathbf{r}(T)) = \nu r_0(T)$ and the costate boundary conditions are given by:

$$\lambda_i(T) = \frac{\partial\Phi(r_0(T), ..., r_{N-1}(T))}{\partial r_i(T)}, \quad i = 0, ..., N-1$$

which implies that

$$\lambda_i = 0 \quad i = 1, ..., N-1, \quad \lambda_0 = \nu \tag{3.15}$$

where ν is some scalar constant. Finally, the optimal solution must satisfy the transversality condition $H(T) + \partial\Phi/\partial t|_{t=T} = 0$, i.e.,

$$-1 + \nu\dot{r}_0(T) + \nu\dot{r}_0(T) = 0$$

which yields: $\nu = 1/2\dot{r}_0(T) < 0$, where the inequality follows from (3.11) and (3.12) which imply that $\dot{r}_0(T) < 0$ and consequently $\nu < 0$.

Theorem 1: There exists a time-invariant solution of (3.5)-(3.12): $w^*(t) = w^*(T)$, $t \in [t_0, T]$.

Proof: Observe that the control variables $w_{ij}(t)$ appear in the problem formulation (3.5)-(3.12) only through $U_i(w(t))$. Applying the Pontryagin minimum principle to (3.13):

$$[U_0^*(t), ..., U_{N-1}^*(t)] = \arg \min_{U_i \geq 0;\ i=0,...,N-1} H(U_i, t, \lambda^*)$$

and making use of the fact that we found $\lambda_i = 0$, $i = 1, ..., N-1$, we have: $U_0^*(t) = \arg\min_{U_0(t)>0}(-1 - \nu U_0(t))$. Recalling that $\nu < 0$, in order to minimize the Hamiltonian, we need to minimize $U_0(t)$. Therefore, the optimal control problem (3.5)-(3.12) is reduced to the following optimization problem which we refer to as **P1**(t):

$$\min_{w(t)} U_0(t) \tag{3.16}$$

$$\text{s.t. } U_i(w(t)) = G_i(w(t))[\sum_{j\in O(i)} w_{ij}(t)(C_s d_{i,j}^2 + C_f) + C_r], \quad i = 1, ..., N-1 \tag{3.17}$$

$$U_0(w(t)) = \sum_{j\in O(0)} w_{0j}(t)(C_s d_{0,j}^2(t)^2 + C_f) \tag{3.18}$$

$$d_{0,j}(t) = \|(x_0(t_0), y_0(t_0)) - (x_j, y_j)\|, \ x_0(t_0), y_0(t_0) \text{ given}$$

$$G_i(w(t)) = \sum_{h\in I(i)} w_{hi}(t) G_h(w), \ i = 1, \ldots, N-1 \tag{3.19}$$

$$\sum_{j\in O(i)} w_{ij}(t) = 1, \quad 0 \le w_{ij}(t) \le 1, i = 0, .., N-1 \tag{3.20}$$

$$\int_{t_0}^{T} U_0(t)dt = R_0^{t_0} \tag{3.21}$$

When $t = T$, the solution of this problem is $w^*(T)$ and depends only on the fixed network topology and the values of the fixed energy parameters in (3.18) and the control variable constraints (3.20). The same applies to any other $t \in [t_0, T)$, therefore, there exists a time-invariant optimal control policy $w^*(t) = w^*(T)$, which minimizes the Hamiltonian and proves the theorem. ■

We emphasize that the solution $w^*(t)$ evaluated at $t = t_0$, is time-invariant in the sense that it does not depend on the energy dynamics in (3.6). However, this does not mean that the optimal routing vector is time-invariant as the source node moves, i.e., that $w^*(t_0) = w^*(t_0 + k\delta)$ for all $k = 0, 1, \ldots$. As already mentioned, we need to solve OCP-I at $t = t_0$ so as to determine $w^*(t_0)$. The value of Theorem 1 is that it allows us to obtain an optimal routing vector through the following NLP, whereas otherwise we would have to solve for an entire vector $w^*(t), \ t \in [t_0, T]$ simply to recover the initial value $w^*(t_0)$:

$$\min_{w(t_0)} \sum_{j\in O(0)} w_{0j}(t_0)(C_s(d_{0,j}(t_0))^2 + C_f) \tag{3.22}$$

$$\text{s.t. } \sum_{j\in O(i)} w_{ij}(t_0) = 1, \ 0 \le w_{ij}(t_0) \le 1, i = 0, .., N-1 \tag{3.23}$$

Since the solution $w^*(t_0)$ obtained through this NLP applies only at $t = t_0$, $w^*(t)$ for $t > t_0$ needs to be updated (unless the source node were to stop moving). Thus, updating the value of t_0 through $t_0 = k\delta$, $k = 1, 2, \ldots$, we solve a sequence of problems $\mathbf{P1}(t_0)$, based on the associated source node positions $(x_0(t_0), y_0(t_0))$ as they become available. Theorem 1 asserts that at each time step, there exists a fixed optimal routing vector $w^*(k\delta) \equiv w_k^*$ associated with the source node's position. Thus, an optimal routing vector at each time step is obtained by solving the corresponding NLP:

$$\min_{w^k} \sum_{j \in O^k(0)} w_{0j}^k (C_s (d_{0,j}^k)^2 + C_f) \tag{3.24}$$

$$\text{s.t.} \sum_{j \in O^k(i)} w_{ij}^k = 1,\ 0 \le w_{ij}^k \le 1, i = 0, .., N-1 \tag{3.25}$$

where w^k is a routing vector at step k, $O^k(i)$ is the set of output nodes of i (which may have changed since some relay nodes may have died), and $d_{0,j}^k = \|(x_0^k, y_0^k) - (x_j, y_j)\|$ is the distance between the source node and node j at the kth step. Observe that in (3.24) the objective value is minimized over w_{0j}^k, $j \in O^k(0)$ leaving the remaining routing probabilities w_{ij}^k, $i = 1, ..., N-1$, $j \in O^k(i)$, subject only to the feasibility constraints (3.25). Therefore, at each iteration, the source node sends data packets to its nearest neighbors in $O^k(0)$ in order to minimize its load. The remaining routing probabilities need to be feasible according to (3.25). The simplest such feasible solution is obtained by sending the inflow of data packets to the neighbors of a relay node uniformly, i.e., $w_{ij}^k = \dfrac{1}{|O^k(i)|}$, $i = 1, ..., N-1$. Finally, at the end of each iteration we update the residual energy of all nodes (initial energies for the next iteration) as follows:

$$r_i^{k+1} = r_i^k - U_i(w^k) \cdot \delta \tag{3.26}$$

If $r_0^{k+1} \le 0$ we declare the network to be dead. However, if $r_i^{k+1} \le 0$, $i = 1, .., N-1$, then we omit dead nodes and update the network topology to calculate w^*_{k+1} in the next iteration with fewer nodes. Note that it is possible for all relay nodes to be dead while $r_0^{k+1} > 0$, implying that the source node still has the opportunity to transmit data directly to the base if $N \in O^{k+1}(0)$.

The fact that the solution of $\mathbf{P1}(t)$ does not allow any direct control over the relay nodes is a potential drawback of this formulation and motivates the next definition of WSN lifetime.

3.4.2 Optimal Control Problem - II

As already mentioned, the optimization problem (3.24)-(3.25) does not directly control the way relay nodes consume their energy. To impose such control on their energy consumption, we add $\sum_{i=1}^{N-1} r_i(T)$ as a terminal cost to the objective function of the optimal control problem (3.5)-(3.12) and formulate a new problem as follows:

$$\min_{w(t)} \left(-\int_{t_0}^{T} dt + \epsilon \sum_{i=1}^{N-1} r_i(T) \right) \quad \text{s.t.} \quad (3.6) - (3.12) \tag{3.27}$$

where $\epsilon > 0$ is a weight reflecting the importance of the total residual energy relative to the lifetime as measured at time t. Thus, in order to minimize the terminal cost, relay nodes are compelled to drive their residual energy to be as close to zero as possible at $t = T$. This plays a role as we solve the sequence of problems resulting for the source node movement: the inclusion of this terminal cost tends to preserve some relay node energy which may become important in subsequent time steps. The solution of (3.27) obviously results in a different network lifetime T^* relative to that of problem (3.5)-(3.12), which is recovered when $\epsilon = 0$. Thus, (3.27) may simply be viewed as a generalization of (3.5)-(3.12) or, conversely, (3.5)-(3.12) is a special case of (3.27).

Optimal control problem II solution

The Hamiltonian based on the new objective function (3.27), as well as the costate equations, are the same as (3.13) and (3.14) respectively. However, the terminal state constraint is now

$$\Phi(\mathbf{r}(T)) = \epsilon \sum_{i=1}^{N-1} r_i(T) + \nu r_0(T)$$

and the costate boundary conditions are given by:

$$\lambda_i(T) = \frac{\partial \Phi(r_0(T), ..., r_{N-1}(T))}{\partial r_i(T)}, \quad i = 0, ..., N-1$$

so that $\lambda_i = \epsilon, \quad i = 1, ..., N-1 \quad \lambda_0 = \nu$. Finally, the transversality condition $H(T) + \partial \Phi / \partial t|_{t=T} = 0$ for this problem is

$$-1 + \nu \dot{r}_0(T) + \epsilon \sum_{i=1}^{N-1} \dot{r}_i(T) + \nu \dot{r}_0(T) + \epsilon \sum_{i=1}^{N-1} \dot{r}_i(T) = 0$$

resulting in

$$\nu = \frac{1 - 2\epsilon \sum_{i=1}^{N-1} \dot{r}_i(T)}{2\dot{r}_0(T)} \leq 0 \tag{3.28}$$

Looking at (3.11) and (3.12) and as already discussed in the previous section, we have $\dot{r}_0(T) < 0$. For the any relay node $i = 1, ..., N-1$, there are two possible cases: (*i*) Node i is not transmitting any data at $t = T$, i.e., the node is already out of energy or the inflow rate to that node is zero, $G_i(w(T)) = 0$. In this case, $U_i(T) = 0$, consequently $\dot{r}_i(T) = 0$. (*ii*) Node i is transmitting, i.e., $U_i(T) > 0$, therefore, $\dot{r}_i(T) < 0$. It follows that $\sum_{i=0}^{N-1} \dot{r}_i(T) \leq 0$ and we conclude that $\nu \leq 0$.

Theorem 2: There exists a time-invariant solution of (3.27): $w^*(t) = w^*(T)$, $t \in [t_0, T]$.

Proof: The proof is similar to that of Theorem 1. First, observe that the control variables $w_{ij}(t)$ appear in the problem formulation (3.27) only through $U_i(w(t))$. Next, applying the Pontryagin minimum principle to (3.13) and based on our analysis we get:

$$[U_0^*(t), ..., U_{N-1}^*(t)] = \arg \min_{U_i(t) \geq 0} [-1 - \nu U_0(t) - \epsilon \sum_{i=1}^{N-1} U_i(t)] \tag{3.29}$$

Recalling that $\nu \leq 0$ in (3.28), in order to minimize (3.29) the routing vector should minimize $U_0(t)$ while maximizing $\epsilon \sum_{i=1}^{N-1} U_i(t)$. Therefore, the optimal control problem (3.27) can be written as the following problem $\mathbf{P2}(t)$:

$$\min_{w(t),\nu} (U_0(t) + \frac{\epsilon}{\nu} \sum_{i=1}^{N-1} U_i(t)) \quad \text{s.t.} \quad (3.17) - (3.21) \tag{3.30}$$

where $\nu < 0$ is an unknown constant which must also be determined (if $\nu = 0$, the problem in (3.29) reduces to maximizing $\epsilon \sum_{i=1}^{N-1} U_i(t)$ and can be separately solved). Using the same argument as in Theorem 1, at $t = T$, the solution $w^*(T)$ depends only on the fixed network topology and the values of the fixed energy parameters in (3.18) and the control variable constraints (3.17)-(3.20). The same applies to any other $t \in [t_0, T)$, therefore, there exists a time-invariant optimal control policy $w^*(t) = w^*(T)$, which minimizes the Hamiltonian and proves the theorem. ■

The intuition behind $\mathbf{P2}(t)$ in (3.30) is that one may prolong the network lifetime by minimizing the load of the source node while maximizing the workload of relay nodes. As in the case of Theorem 1, the value of Theorem 2 is that once again it allows us to reduce the evaluation of the *instantaneous* routing vector $w^*(t_0)$ to a NLP, rather than solving for a full vector $w^*(t)$ just to get $w^*(t_0)$. Once again, this does not mean that the full $w^*(t)$ is time-invariant as the source node moves. As in the case of $\mathbf{P1}(t)$, we proceed by discretizing the source node trajectory and determining at step k an optimal routing vector w_k^* and associated ν_k^* by solving the following NLP:

$$\min_{w^k,\nu^k} \left(U_0(w^k) + \frac{\epsilon}{\nu^k} \sum_{i=1}^{N-1} U_i(w^k) \right) \quad \text{s.t. (3.28) and} \tag{3.31}$$

$$U_i(w^k) = G_i(w^k)[\sum_{j\in O^k(i)} w_{ij}^k(C_s d_{i,j}^2 + C_f) + C_r] \tag{3.32}$$

$$U_0(w^k) = \sum_{j\in O^k(0)} w_{0j}^k(C_s(d_{0,j}^k)^2 + C_f) \tag{3.33}$$

$$G_i(w^k) = \sum_{h\in I^k(i)} w_{hi}^k G_h(w^k),\ i = 1,\ldots,N-1 \tag{3.34}$$

$$\sum_{j\in O^k(i)} w_{ij}^k = 1, \quad 0 \leq w_{ij}^k \leq 1,\ i = 0,\ldots,N-1 \tag{3.35}$$

We then evaluate and update the energy level of all nodes using (3.26) and check the terminal constraint (3.11) at the end of each iteration. If the source node is “alive”, we update the network topology to eliminate any relay nodes that may have depleted their energy in the current time step. Note that in order to solve (3.31)-(3.35) we also need to determine ν^k so that it satisfies (3.28) with $\dot{r}_i^*(T) = -U_i(w^*(T)) = -U_i(w_k^*)$. To do so, we start with an initial value and iteratively update it until (3.28) is satisfied. This extra step adds to the problem's computational complexity and motivates yet another definition of WSN lifetime.

3.4.3 Optimal Control Problem - III

In this section, we revise the terminal constraint used in Problem I in order to improve the total energy consumption in the network and possibly reduce the computational effort required in $\mathbf{P2}(t)$ due to the presence of ν in (3.31) and (3.30). Thus, let us replace the terminal constraint (3.11), i.e., $r_0(T) = 0$, by $\sum_{i=0}^{N-1} r_i(T) = 0$, therefore redefining the WSN lifetime as the time when *all* nodes deplete their energy. Compared to Problem II where we included $\sum_{i=1}^{N-1} r_i(T)$ as a *soft* constraint on the total residual relay node energy, here we impose it as a *hard* constraint. The following result asserts that the source node 0 must still die at $t = T$, just as in Problem I.

Lemma 1: Consider (3.5)-(3.12) with (3.11)-(3.12) replaced by $\sum_{i=0}^{N-1} r_i(T) = 0$. Then, $\dot{r}_0(T) < 0$.

Proof: Proceeding by contradiction, suppose $\dot{r}_0(T) = 0$, consequently $r_0(t_1) = 0$ for some $t_1 < T$ and there must exist some node $i > 0$ such that $r_i(t_1) > 0$ otherwise the network would be dead at $t_1 < T$. Then, $w_{0j}(t_1) = 0$. This implies that $G_j(w(t_1)) = 0$ for all $j \in O(0)$, i.e., there is no inflow to process at any node $j \in O(0)$, therefore, $G_i(w(t_1)) = 0$ at all nodes $i > 0$ contradicting the fact that $r_i(t_1) > 0$ for some $i > 0$. ■

Optimal control problem III solution

We apply the new terminal constraint to problem (3.5)-(3.12), i.e., replace (3.11)-(3.12) by

$$\sum_{i=0}^{N-1} r_i(T) = 0 \tag{3.36}$$

The Hamiltonian is still the same as (3.13) and the costate equations remain as in (3.14). However, the terminal state constraint, as well as the costate boundary conditions, are modified as follows:

$$\Phi(\mathbf{r}(T)) = \nu \sum_{i=0}^{N-1} r_i(T) \tag{3.37}$$

$$\lambda_i(T) = \nu \frac{\partial \Phi(r_0(T), ..., r_{N-1}(T))}{\partial r_i(T)} = \nu,\ i = 0, \ldots, N-1 \tag{3.38}$$

Thus, the costates over all $t \in [t_0, T]$ are identical constants, $\lambda_0(t) = ... = \lambda_{N-1}(t) = \nu$. Similar to our previous analysis, we use the transversality condition $H(T) + \partial \Phi / \partial t|_{t=T} = 0$ to investigate the sign of ν: $-1 + \sum_{i=0}^{N-1} \nu \dot{r}_i(T) + \nu \sum_{i=0}^{N-1} \dot{r}_i(T) = 0$ and we get

$$\nu = \frac{2}{\sum_{i=0}^{N-1} \dot{r}_i(T)} \leq 0$$

by examining all possible cases for the state of relay nodes at $t = T$ as we did for (3.28). Finally, applying the Pontryagin minimum principle leads to the following optimization problem $\mathbf{P3}(t)$:

$$\min_{w(t)} \sum_{i=0}^{N-1} U_i(t) \quad \text{s.t.} \quad (3.17) - (3.20) \text{ and} \tag{3.39}$$

$$\sum_{i=0}^{N-1} \int_{t_0}^{T} U_i(t)dt = \sum_{i=0}^{N-1} R_i^{t_0} \tag{3.40}$$

This new formulation indicates that the optimal routing vector corresponds to a policy minimizing the overall network workload during its lifetime, T. We can once again establish the fact that there exists a time-invariant solution of (3.39)-(3.40) $w^*(t) = w^*(T)$, $t \in [t_0, T]$ with similar arguments as in Theorems 1 and 2, so we omit this proof. We then proceed as before by discretizing the source node trajectory and determining at step k an optimal routing vector w_k^* by solving the NLP:

$$\min_{w^k} \sum_{i=0}^{N-1} U_i(w^k) \quad \text{s.t.} \quad (3.32) - (3.35) \tag{3.41}$$

Note that problem (3.39)-(3.40) is not always feasible. In fact, its feasibility depends on the initial energies of the nodes at each iteration, i.e., $r_i(t_0) = R_i^{t_0}$, $i = 0, .., N-1$, in (3.6). As we discussed it in Chapter 2 for a fixed network topology, if we can optimally allocate initial energies to all nodes, this results in all nodes dying simultaneously, which is exactly what (3.36) requires. More specifically, recalling Remark-4 in Chapter 2 for the problem with ideal battery dynamics, the optimal routing vector can be obtained by solving the following NLP (Wu and Cassandras, 2005):

$$\min_{w} \sum_{i=0}^{N-1} G_i(w) \left(\sum_{i \prec j,\ j<N} w_{i,j} k_{ij} + k_{i,N} \right)$$

$$\text{s.t.} \quad 0 \leq w_{i,j} \leq 1, \quad 0 \leq i, j \leq N \text{ and } i \prec j$$

$$\sum_{i \prec j,\ j<N} w_{i,j} \leq 1,\ G_i(w) > 0$$

which is equivalent to problem 3.41. We then determine the optimal initial energy for each node $R_i^*,\ i = 0, ..., N-1$, through

$$R_i^*(w) = \frac{\bar{R}}{K_i(w)} \left[\sum_{j=0}^{N-1} \frac{1}{K_j(w)} \right]^{-1}, \quad i = 1, \ldots, N-1$$

This solution results in a scenario in which all nodes die simultaneously. where $K_i(w) = [G_i(w) g_i(w)]^{-1}$ and $\bar{R}$ is the total initial energy.

However, such degree of freedom does not exist in (3.39)-(3.40), therefore, one or more instances of (3.41) for $k = 0, 1, \ldots$ is likely to lead to an infeasible solution for the original problem since we cannot control R_i^k. Clearly, this makes the definition of WSN lifetime through (3.36) undesirable. Nonetheless, we follow up on it for the following reason: We will show next that (3.41), if feasible, is equivalent to a shortest path problem and this makes it extremely efficient for on-line solution at each time step along the source node trajectory. Thus, if we adopt a shortest path routing policy at every step k, even though it is no longer guaranteed that this solves (3.41) since (3.36) may not be satisfied for the values of R_i^k at this step, we can still update all node residual energies through (3.26) and check whether $r_0^{k+1} \leq 0$. The network is declared dead as soon as this condition is satisfied, even if $\sum_{i=0}^{N-1} r_i^{k+1} \geq 0$. Although (3.36) is not satisfied at the kth step, this approach provides a computationally efficient heuristic for maximizing the WSN lifetime over the source node trajectory in the sense that when $r_0^{k+1} \leq 0$ at time $k\delta$, the lifetime is $T = k\delta$ and this may compare favorably to the solution obtained through the Problem II formulation where both lifetimes satisfy $r_0(T) = 0$ with $\dot{r}_0(T) < 0$ (by Lemma 1). This idea is tested in Section 3.4.4.

Transformation of Problem III to a shortest path problem

The WSN can be modeled as a directed graph from the source (node 0) to a destination (node N). Each arc (i, j) is a transmission link from node i to node j. The weight of arc (i, j) is defined as $Q_{ij} = C_r + C_s \cdot d_{i,j}^2 + C_f$ (for the source node $C_r = 0$) which is the energy consumption to receive one bit of information and transmit it from node i to node j. A path from the source to the destination node is denoted by p with an associated cost defined as $C_p = \sum_{(i,j)\in p} Q_{ij}$. Clearly, for each bit of information, the total energy cost to deliver it from the source node to the base station through path p is C_p.

Theorem 3: If problem (3.39)-(3.40) is feasible, then its solution obtained using (3.41), is equivalent to the shortest path on the graph weighted by the transmission energy costs Q_{ij} for each arc (i, j).

Proof: We first prove that if the solution of (3.41) includes multiple paths from node 0 to N where nodes in the path have positive residual energy, then the paths have the same cost. We proceed using a contradiction argument. Suppose that in the optimal solution there exist two distinct paths P_1^* and P_2^* such that $C_{P_1^*} < C_{P_2^*}$. Let $q_{P_1^*}$ and $q_{P_2^*}$ be the amounts of information transmitted through P_1 and P_2 respectively in a time step of length δ, i.e., $q_{P_1^*} + q_{P_2^*} = G_0 \cdot \delta$.

In addition, let $\bar{r}_k^*$ be the total amount of energy consumed under an optimal routing vector w_k^* over the time step of length δ, i.e., $\bar{r}_k^* = \sum_i U_i(w_k^*) \cdot \delta$. It follows that $q_{P_1^*} C_{P_1^*} + q_{P_2^*} C_{P_2^*} = \bar{r}_k^*$. Suppose we perturb the optimal solution so that an additional amount of data $\xi > 0$ is transmitted through P_1^*. Then:

$$(q_{P_1^*} + \xi)C_{P_1^*} + (q_{P_2^*} - \xi)C_{P_2^*} = \bar{r}_k^* + \xi(C_{P_1^*} - C_{P_2^*}) < \bar{r}_k^*$$

This implies that $\sum_i U_i(w_k^*)$ is not the minimum cost and the original solution is not optimal, leading to a contradiction.

We have thus established that if the solution of (3.41) includes multiple paths from node 0 to N where nodes in the path have positive residual energy, then the paths have the same cost. Recall that arc weights correspond to energy consumed, therefore the shortest path on the graph weighted by the transmission energy costs guarantees the lowest cost to deliver every bit of data from the source node to the base station, i.e., $\min \sum_{i=0}^{N-1} U_i(w^k)$. ■

3.4.4 Numerical examples

In this section, we use a WSN example to compare the performance of different formulations based on the three different network lifetime definitions we have considered. We consider a 6-node network as shown in Fig. 3·1. Nodes 1 and 6 are the source and base respectively, while the rest are relay nodes. Let us set $C_s = 0.0001, C_f = C_r = 0.05$, and $\beta = 2$ in the energy model. We also set initial energies for the nodes $R_i = 80$, $i = 1, ..., 5$. Starting with the source node at $(x_0(0), y_0(0)) = (0, 0)$, we solve the two optimization problems (3.31)-(3.35) with $\epsilon = 1$ and the equivalent shortest path problem of (3.39) for OCPs II and III respectively as the trajectory of the source node evolves. Since this trajectory is not known in advance, in this example we assume the source node moves based on a random walk as shown in Fig. 3·1. We first find the optimal routing vector by solving (3.31)-(3.35) at each time step along the source node trajectory treating the network topology as fixed for that step. Fig. 3·2 shows the routing vectors as well as the evolution of residual energies of all nodes during the network lifetime, i.e., the time when the source node depletes its battery.

We can see that at $T = 187.6$ the residual energy of the source node drops to zero, hence that is the optimal lifetime obtained using the definition where the soft constraint $\sum_{i=1}^{N-1} r_i(T)$ is included in (3.27) with $\epsilon = 1$. Next, we use the WSN definition where $\sum_{i=1}^{N-1} r_i(T) = 0$ is used as a hard constraint. As already discussed,

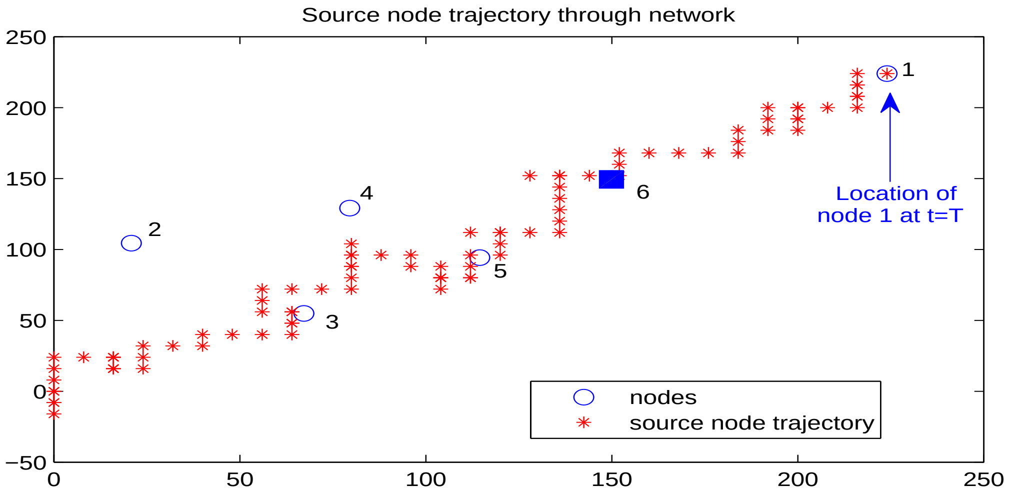
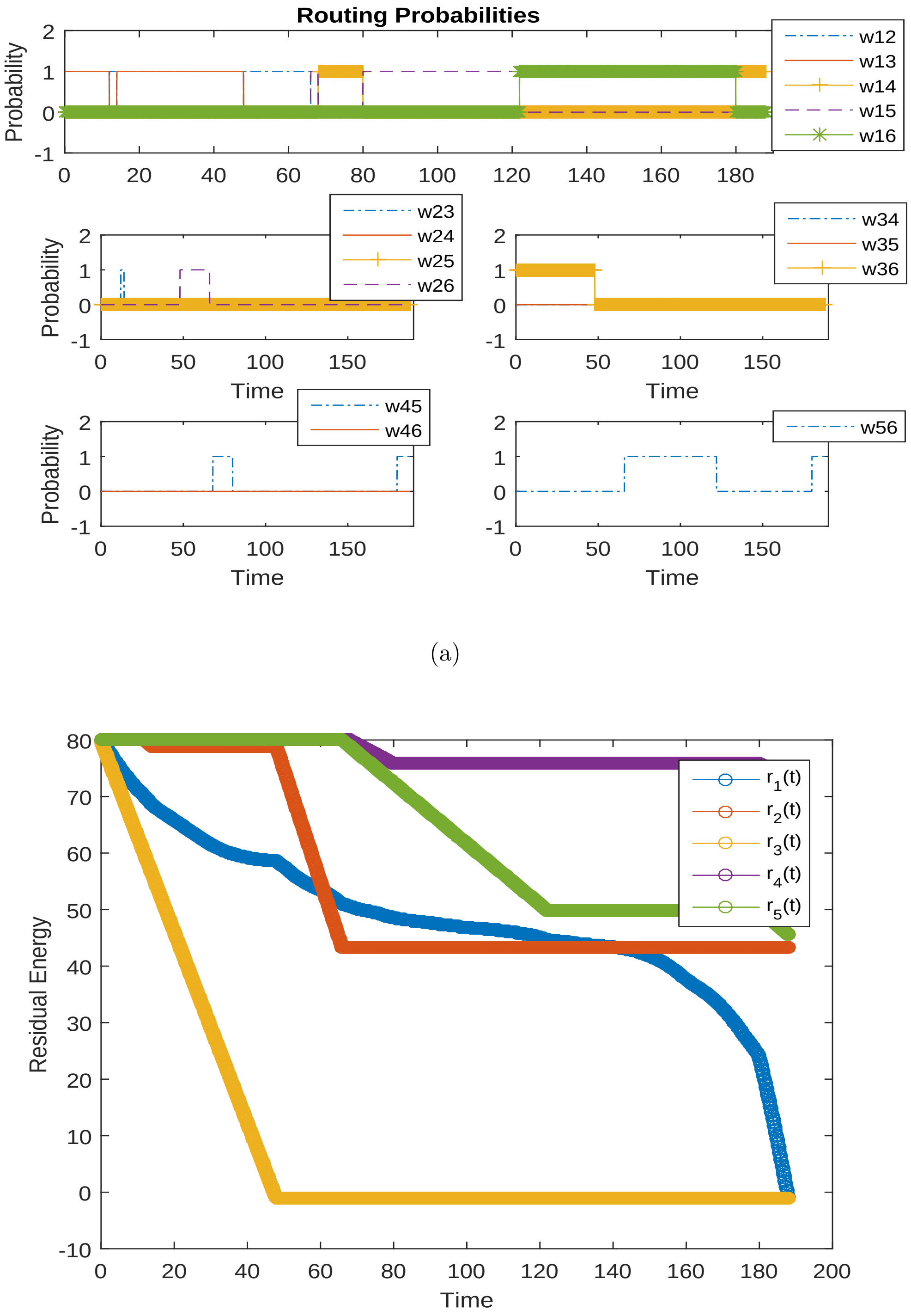


Figure 3·1: 6-node network with mobile source (node 1)

the corresponding problems (3.41) over the source node trajectory are generally results in an infeasible solution for the original problem. Instead, we adopt the shortest path routing policy at each step to exploit Theorem 3 with the understanding that the result (for this particular WSN definition) is suboptimal. We consider the same source node trajectory as in Fig. 3·1. The optimal routing vector updates as well as the residual energy of the nodes during the network lifetime are shown in Fig. 3·3. In this case $T = 194.1$, which is slightly longer than the one obtained in Fig. 3·2.(b) with considerably less computational effort. Also, note that since the source node always sends data packets through the shortest path, it never uses nodes 2 and 4 for this particular trajectory. As expected, (3.39)-(3.40) is not feasible, however finding the shortest path at each step in fact improves the network lifetime in the sense of the first time when the source node depletes its energy. We point out, however, that this is not always the case and several additional numerical examples show that this depends on the actual trajectory relative to the relay node locations.

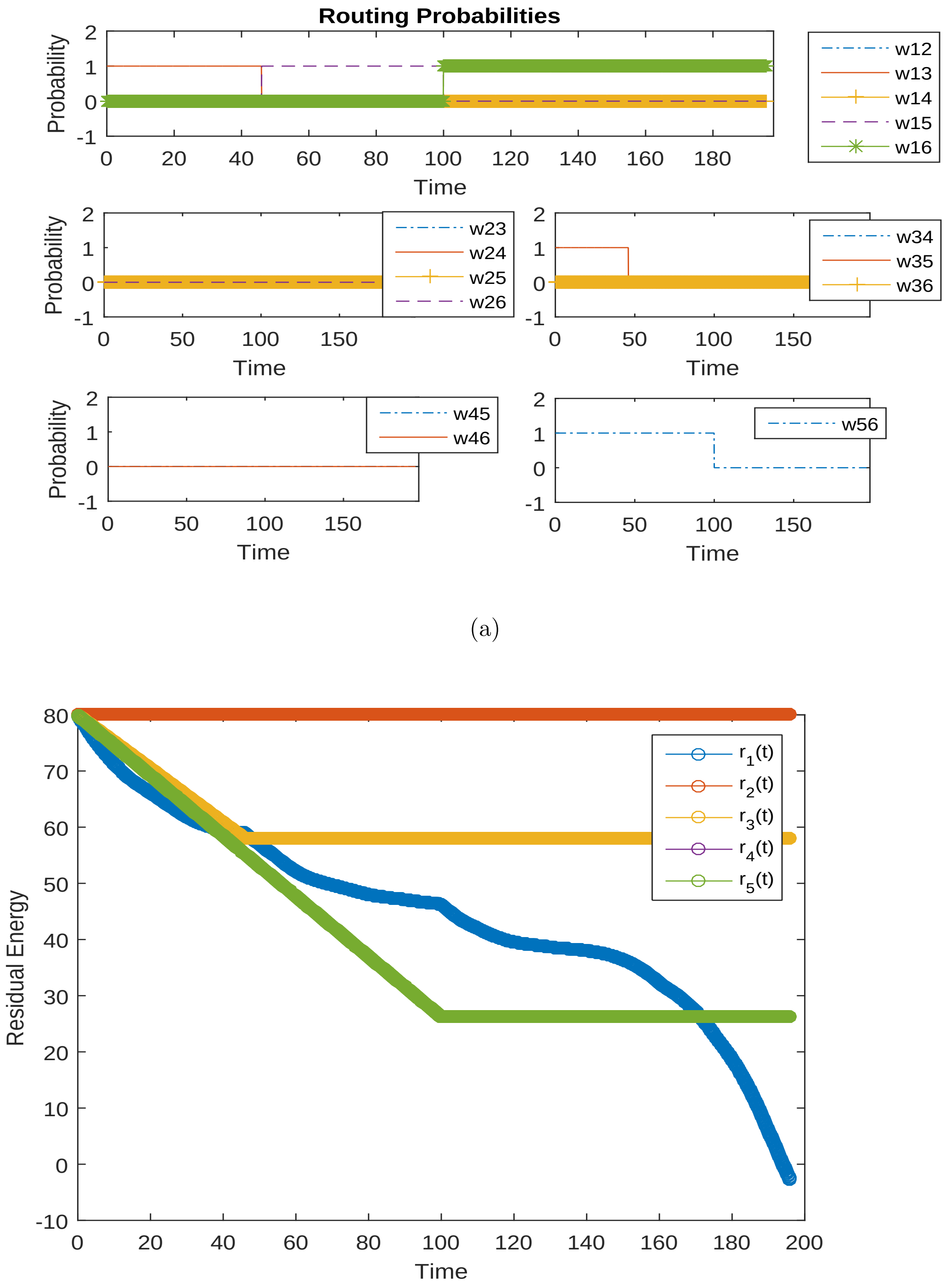
Recall that ϵ is the weight of the soft constraint in problem $\mathbf{P2}(t)$. Applying small or large ϵ makes the problem closer to $\mathbf{P1}(t)$ or $\mathbf{P3}(t)$ respectively. Tab. 3.1 shows



(a)

(b)

Figure 3·2: (a) Routing vector ; (b) Residual energies over time during the network lifetime (Problem II)



(a)

(b)

Figure 3·3: (a) Routing vector ; (b) Residual energies over time during the network lifetime (Problem III)

Table 3.1: Network lifetime using OCP-II for different values of ϵ

ϵ	0	0.1	0.5	1	8
T	203.1	199	198.8	187.6	160.5

the network lifetime for different values of ϵ. It is observed that in this scenario, it is not optimal to encourage the nodes to die simultaneously which is often viewed as a desirable heuristic. On the other hand, applying OCP-I ($\epsilon = 0$) with uniform routing probabilities for relay nodes, i.e., $w_{ij}^k = 1/|O^k(i)|$, results in the longest lifetime $T = 203.1$. Based on the numerical results, it is obvious that the definition of a static WSN lifetime is not appropriate here. Finally, we observe that the routing vectors are such that at each time step a subset of nodes is fully used ($w_{ij} = 1$) while the rest are not used at all. This suggests the possibility of conditions under which a "bang-bang" type of optimal routing policy, an issue which deserves further investigation.

3.5 Optimal Control Formulation when source node trajectory is known in advance

In this section, we consider the case when we have full advance knowledge of the source node trajectory and include this information in the optimal control problem. Defining the WSN lifetime to be the time when the source node depletes its energy, i.e., using the definition in OCP I, Section 3.4.1, the problem is formulated as follows:

$$\min_{w(t)} - \int_0^T dt \tag{3.42}$$

$$\text{s.t. } \dot{r}_i(t) = -U_i(w(t)),\ r_i(0) = R_i, \quad i = 0, .., N-1 \tag{3.43}$$

$$\begin{bmatrix} \dot{x}_0(t) \\ \dot{y}_0(t) \end{bmatrix} = \begin{bmatrix} f_x(x_0(t), y_0(t)) \\ f_y(x_0(t), y_0(t)) \end{bmatrix}, \quad (x_0(0), y_0(0)) \text{ given} \tag{3.44}$$

$$U_i(w(t)) = G_i(w(t))[\sum_{j \in O(i)} w_{ij}(t)(C_s d_{i,j}^2 + C_f) + C_r], \quad i = 1, ..., N-1 \tag{3.45}$$

$$U_0(w(t)) = \sum_{j \in O(0)} w_{0j}(t)(C_s d_{0,j}^2(t) + C_f) \tag{3.46}$$

$$G_i(w(t)) = \sum_{k \in I(i)} w_{ki}(t) G_k(w(t)),\ i = 1,..,N-1 \tag{3.47}$$

$$\sum_{j \in O_i} w_{ij}(t) = 1, \quad 0 \le w_{ij}(t) \le 1, i = 0, \ldots, N-1 \tag{3.48}$$

$$r_0(T) = 0 \tag{3.49}$$

$$r_0(t) > 0,\ t \in [0,T);\ r_i(t) \ge 0,\ i = 1,..,N-1,\ t \in [0,T] \tag{3.50}$$

where (3.44) specifies the trajectory of the source node. In this problem, the state variables are the residual node energies, $r_i(t)$, as well as the source node location at time t, $(x_0(t), y_0(t))$. One should note that we no longer need to use t_0 as the initial time, since we solve the problem for the entire network lifetime, i.e., $t \in [0,T]$.

Similar to Section 3.4.1, we obtain the Hamiltonian (Bryson and Ho, 1975):

$$\begin{aligned} &H(w,t,\lambda) = -1 + \lambda_0(t)(-U_0(t)) + \sum_{i=1}^{N-1} \lambda_i(t)(-U_i(t)) + \\ &\lambda_x(t) f_x(x_0(t), y_0(t)) + \lambda_y(t) f_y(x_0(t), y_0(t)) \end{aligned} \tag{3.51}$$

As before, $\lambda_i(t)$ is the costate corresponding to $r_i(t),\ i = 0, ..., N-1$ and we add $\lambda_x(t),\ \lambda_y(t)$ to be the costates of $x_0(t)$ and $y_0(t)$. Since in this case we know the equation of motion for the source node in advance, this imposes terminal constraints for the location of the source node at $t = T$. Thus, based on the dynamics in (3.44) we can specify $x_0(T)$ and $y_0(T)$ as $x_0(T) = F_{x_0}(T)$ and $y_0(T) = F_{y_0}(T)$. Therefore, the terminal state constraint is:

$$\begin{aligned} &\Phi(\mathbf{r}(T), x_0(T), y_0(T)) = \\ &\nu r_0(T) + \mu_x(x_0(T) - F_{x_0}(T)) + \mu_y(y_0(T) - F_{y_0}(T)) \end{aligned} \tag{3.52}$$

where $\nu,\ \mu_x,$ and μ_y are unknown constants. It is straightforward to show that $\lambda_i(t)$, $i = 1, ..., N-1$ are as in (3.15). On the other hand, λ_x and λ_y must satisfy:

$$\dot{\lambda}_x(t) = -\frac{\partial H}{\partial x_0} = 2C_s\lambda_0(t) \sum_{j\in O(0)} [w_{0j}(t)(x_0(t) - x_j)] - \lambda_x(t)\frac{\partial f_x}{\partial x_0} - \lambda_y(t)\frac{\partial f_y}{\partial x_0} \tag{3.53}$$

$$\dot{\lambda}_y(t) = -\frac{\partial H}{\partial y_0} = 2C_s\lambda_0(t) \sum_{j\in O(0)} [w_{0j}(t)(y_0(t) - y_j)] - \lambda_x(t)\frac{\partial f_x}{\partial y_0} - \lambda_y(t)\frac{\partial f_y}{\partial y_0} \tag{3.54}$$

with boundary conditions:

$$\lambda_x(T) = \frac{\partial\Phi(\mathbf{r}(T), x_0(T), y_0(T))}{\partial x_0(T)} = \mu_x \tag{3.55}$$

$$\lambda_y(T) = \frac{\partial\Phi(\mathbf{r}(T), x_0(T), y_0(T))}{\partial y_0(T)} = \mu_y \tag{3.56}$$

The transversality condition $H(T) + \left.\frac{\partial\Phi}{\partial t}\right|_{t=T} = 0$ gives:

$$\begin{aligned} -1 + \nu\dot{r}_0(T) + \lambda_x(T)\dot{x}_0(T) + \lambda_y(T)\dot{y}_0(T) + \nu\dot{r}_0(T) + \\ \mu_x\dot{x}_0(T) - \mu_x\frac{dF_{x_0}(T)}{dT} + \mu_y\dot{y}_0(T) - \mu_y\frac{dF_{y_0}(T)}{dT} = 0 \end{aligned} \tag{3.57}$$

Owing to the complexity of (3.53) and (3.54), we cannot analytically obtain $\lambda_x(t)$ and $\lambda_y(t)$. We shall also adjoin equality and inequality path constraints (3.48) and (3.50) to the Hamiltonian and investigate optimality conditions at potential corner points (Bryson and Ho, 1975).

The solution of this problem is computationally challenging. Thus, we solve this optimal control problem (OCP) numerically using GPOPS-II (Patterson and Rao, 2014), a MATLAB-based general purpose optimal control software that approximates a continuous-time OCP as a large sparse nonlinear programming problem (NLP) using variable-order Gaussian quadrature collocation methods (Patterson and Rao, 2014). The resulting NLP is then solved using IPOPT, an NLP solver. Fortunately, this procedure can be done off line in advance of the source node initiating its known trajectory.

3.5.1 Numerical Examples

Consider a 5-node network as shown in Fig. 3·4 in which nodes 1 and 5 are the source and base respectively while the rest are relay nodes. First we assume the source node travels along a straight line with a constant velocity, then, $\dot{x}_0(t) = v_x$, $\dot{y}_0(t) = v_y$ in (3.44) with $v_x = 1$ and $v_y = 2/3$. We consider the energy model parameters similar to those in section 3.4.4 and set the initial energies for the nodes as $R_1 = 140$ and $R_{2,3,4} = 100$. Assuming $(x_0(0), y_0(0)) = (0, 0)$, we solve the corresponding OCP (3.42)-(3.50) using GPOPS-II. Fig. 3·5 shows the routing vector during the network lifetime as well as evolution of the residual energies of all nodes while the source node travels.

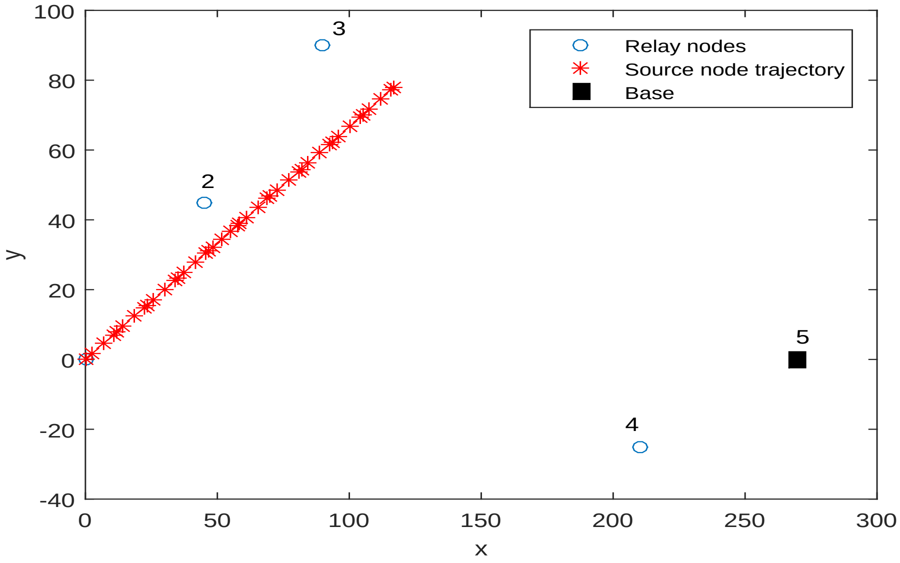
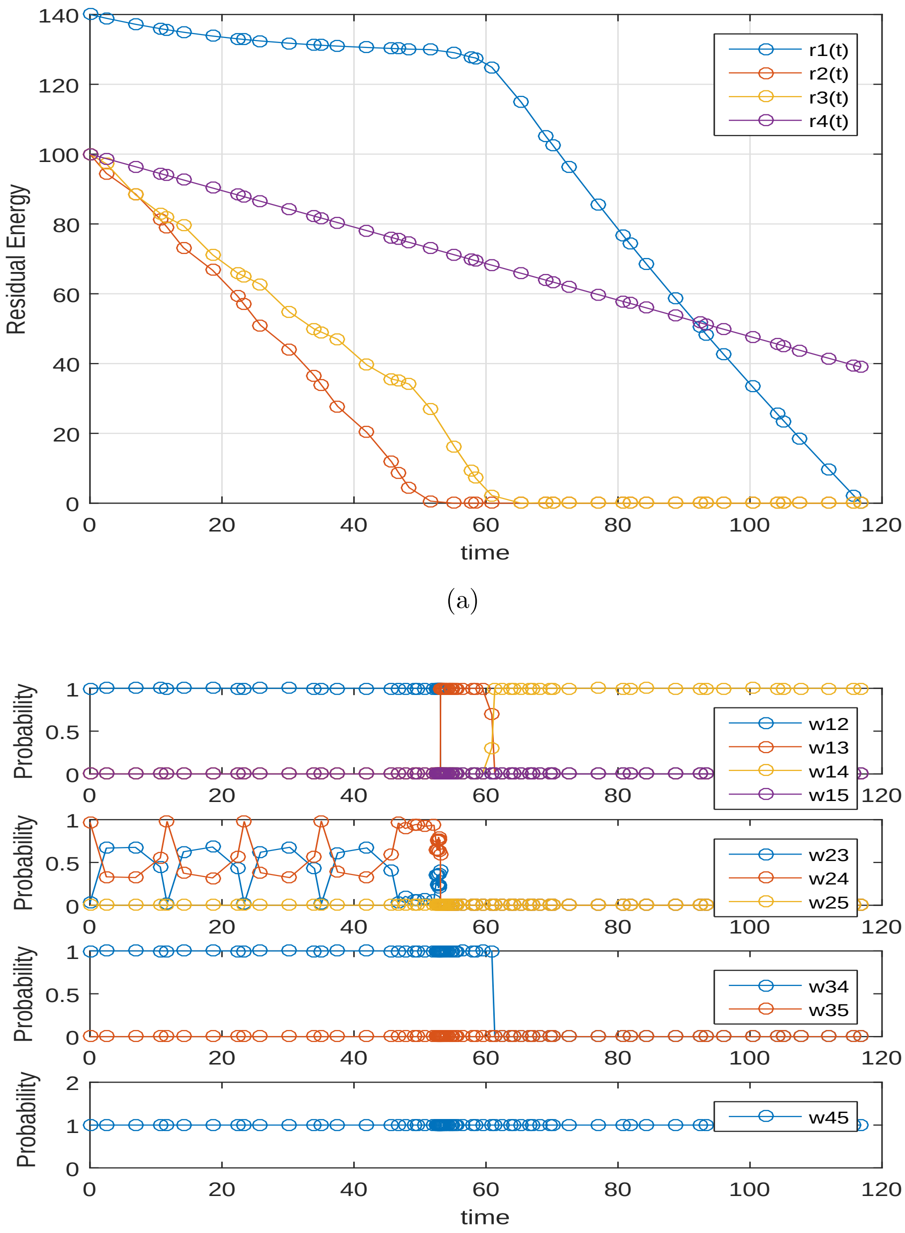


Figure 3·4: 5-node network with mobile source

As observed in Fig. 3·5, in this scenario the source node always sends data packets to the nearest neighbor in order to prolong its lifetime. First, it sends 100% of the generated data to node 2 until it dies at time 51.6. Then, it sends data packets to the

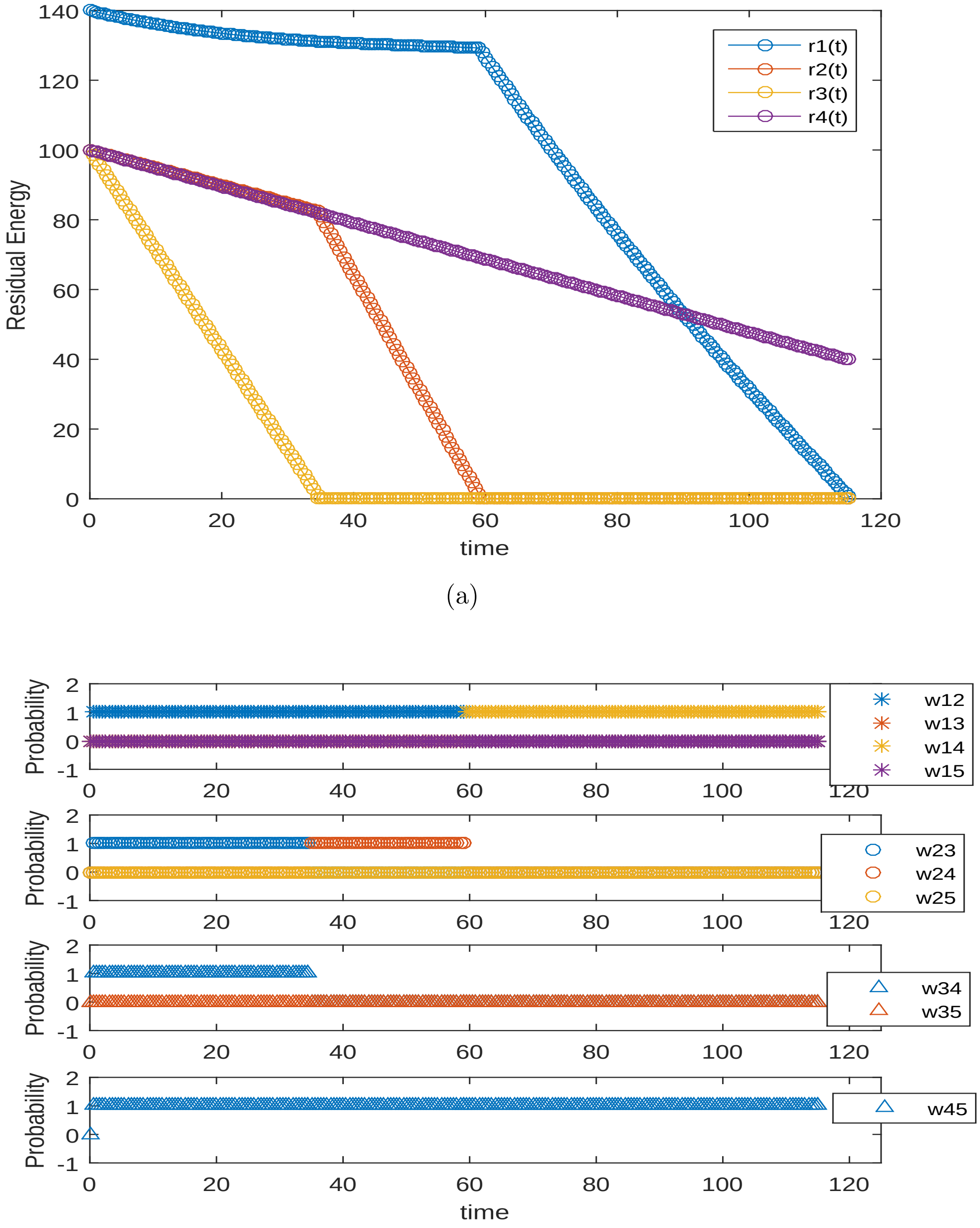


(a)

(b)

Figure 3·5: (a) Residual energies over time during the network lifetime; (b) Optimal routing vector next available nearest relay node, node 3. Once node 3 runs out of energy, $t = 65.3$, the source node transmits data packets to the base via node 4. Finally, at $t = 116.8$ the source node depletes its energy. This optimal solution suggests a greedy policy in which each node sends the inflow of data packets to its available nearest neighbor. Fig. 3·6 shows the routing vector and evolution of residual energies of all nodes under this greedy policy for the same scenario as in Fig. 3·4. It is observed that the greedy policy results in almost the same lifetime for the network.

Next we consider a more interesting example in which the source node travels over a sinusoidal trajectory described through $\dot{x}_0(t) = v_x,\ \dot{y}_0(t) = AB\cos(Bt)$ in (3.44) with $v_x = 1$ and $A = 55$ and $B = 1/15$. Solving the corresponding OCP, Fig. 3·7 shows the network topology and source node trajectory during its lifetime and Fig. 3·8 shows all nodes residual energies as well as the optimal routing vector in this scenario. Unlike the previous example, here the optimal routing vector is such that it prolongs the lifetime of node 3, resulting in extending source node lifetime. In other words, due to the prior knowledge of the source node trajectory, it is optimal that node 3 remains alive for a longer time compared to the scenario shown in Fig. 3·4. Thus node 2 just sends half of its inflow packets to node 3. Applying the nearest-neighbor greedy policy to the same scenario, Fig. 3·9 shows the evolution of residual energies as well as the greedy routing vector. It is observed that the greedy policy is not optimal in this case and results in the network lifetime of $211 < 298.7$ obtained under the optimal policy.



(a)

(b)

Figure 3·6: (a) Residual energies over time during the network lifetime; (b) Routing vector under greedy policy

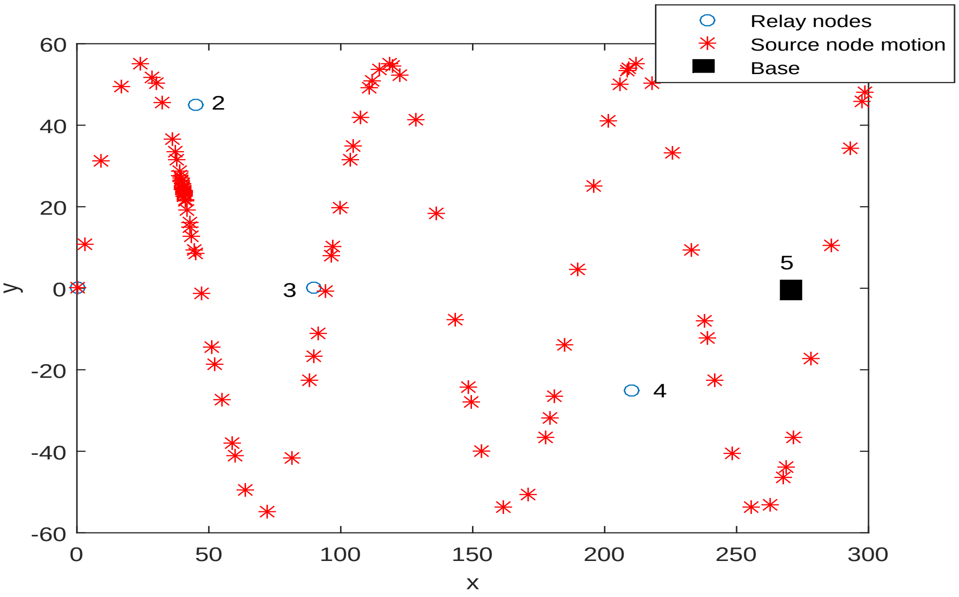
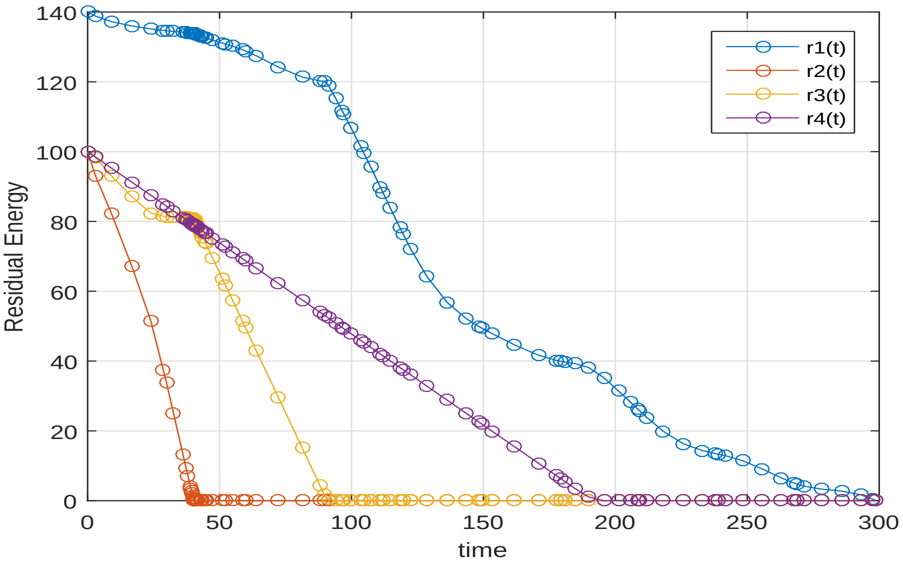
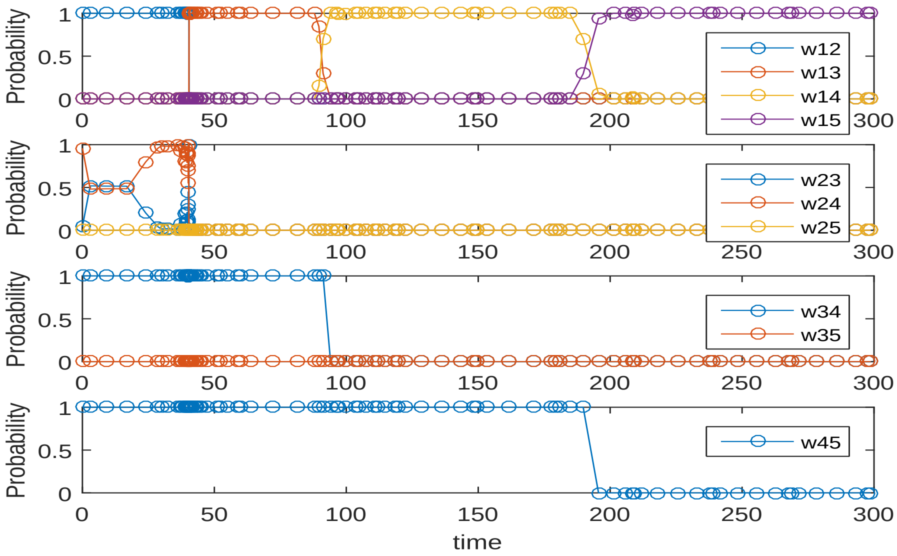


Figure 3·7: 5-node network with mobile source

Finally, we investigate how the prior knowledge of the source node's motion dynamics helps improving network lifetime. To do so, we consider the same sinusoidal trajectory while we assume there is no information about the equation of motion and the source node trajectory evolves with a time step of $\delta = 1$. We then find the network lifetime applying OCPs II and III introduced in Section 3.4. Fig. 3·10 shows the nodes' residual energies over time under the routing policies resulting from both formulations II and III with $T = 112.9$ and $T = 147.3$ respectively. It is observed that the lack of knowledge of the source node trajectory in this case results in a lifetime which is less than half of the optimal value $T^* = 298.7$ obtained with advance knowledge of the source node trajectory.

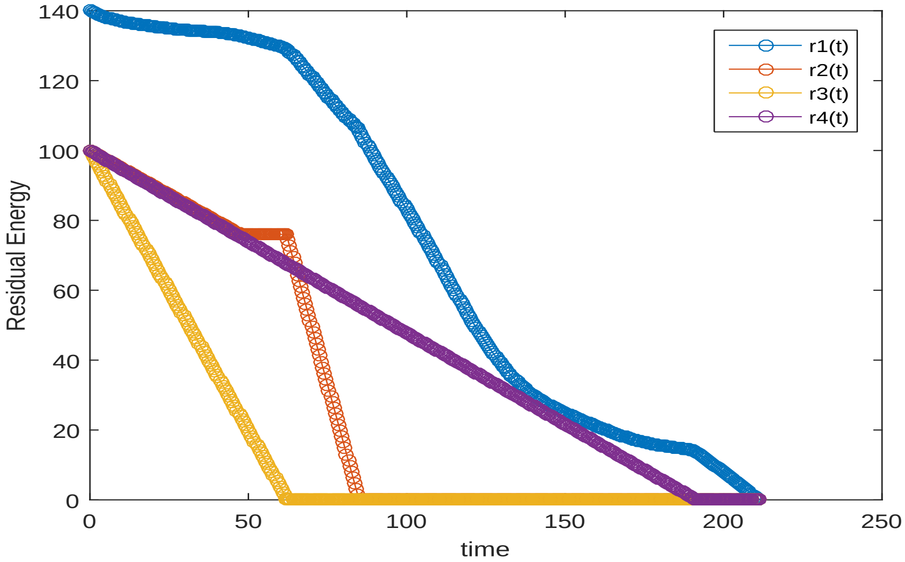


(a)

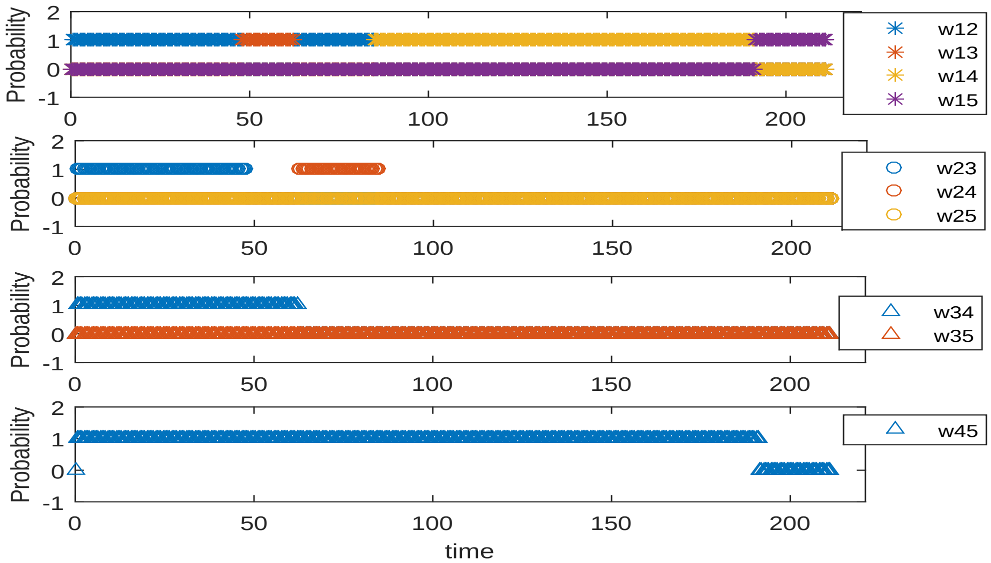


(b)

Figure 3·8: (a) Residual energies over time during the network lifetime; (b) Optimal routing vector for the sinusoidal trajectory

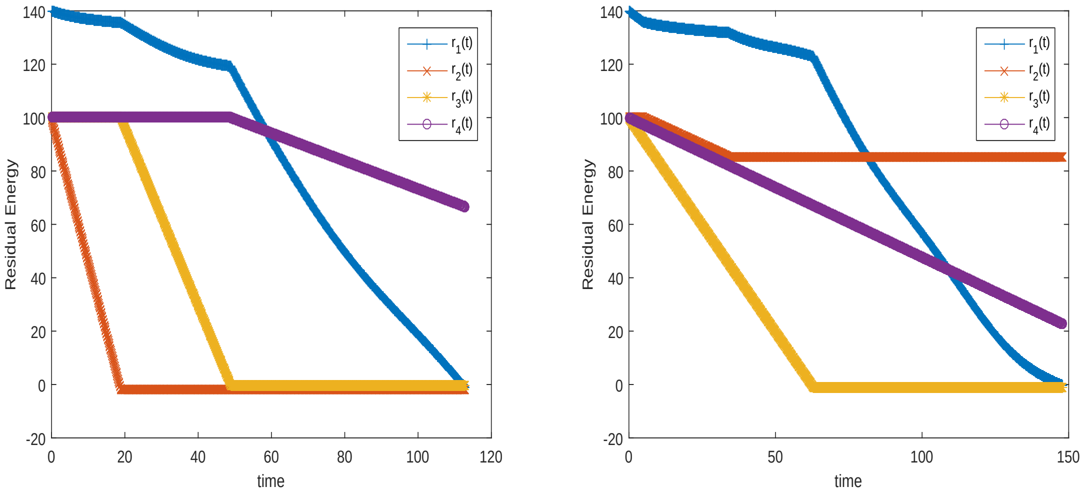


(a)



(b)

Figure 3·9: (a) Residual energies over time during the network lifetime; (b) Routing vector under the greedy policy for the sinusoidal trajectory



(a)

(b)

Figure 3·10: (a) Residual energies over time (Problem II); (b) Residual energies over time (Problem III)

3.6 Summary

We have redefined the lifetime for WSNs with a mobile source node to be the time until the source node runs out of energy. When the mobile node's trajectory is unknown in advance, we have shown that optimal routing vectors can be evaluated as solutions of a sequence of NLPs as the source node trajectory evolves. When the mobile node's trajectory is known in advance, we formulate an optimal control problem which requires an explicit off-line numerical solution. Our examples show that the prior knowledge of the source node's motion dynamics considerably increases the network lifetime.

Chapter 4

Optimal Routing and Charging of Energy-Limited Vehicles in Traffic Networks

4.1 Introduction

As reported by the U.S. Department of Energy, transportation was responsible for almost three-quarters of total U.S. petroleum consumption in 2014. From increasing energy security to reducing emissions of greenhouse gases, Battery-Powered Vehicles, such as Electric Vehicles (EVs), offer a revolutionary pathway to an energy efficient, environmentally friendly transportation system. On the other hand, based on the International Energy Agency (IEA) road-map vision (IEA, 2011), at least 50% of Light Duty Vehicle (LDV) sales worldwide should include Electric and Plug-in Hybrid Electric Vehicles (EVs/PHEVs) by 2050. This significant rise of BPVs in traffic networks has introduced new challenges in classical network routing problems (Laporte, 1992). In particular, by integrating BPVs into traffic flows, we deal with a network routing problem in which the routing decision can be affected by the dynamic behavior of a physical attribute of some entities. Here, the key physical attribute is *energy*. In general, BPVs face significant battery-related challenges which are crucial in routing problems including limited driving range, long recharge time, sparse coverage of charging stations, and the BPV energy recuperation ability (Artmeier et al., 2010) which can be exploited. The Regenerative Braking System (RBS) causes the energy recuperation ability in EVs which can be exploited to extend their cruising range. In this context, (Artmeier et al., 2010) addresses an energy-optimal routing problem. Incorporating the recuperation ability and energy constraints, the general shortest-path algorithm is extended to be adaptable for this problem. Considering both limited energy supply and energy recuperation ability, (Eisner et al., 2011) studies the energy-efficient routing problem for EVs. Employing a generalization of Johnson's potential shifting technique to the famous Dijkstra's algorithm, a computationally efficient route planning algorithm is proposed in this work. This algorithm is applicable to any road network graph whose edge costs represent energy consumption or energy recuperation. In (Sachenbacher et al., 2011) the problem of energy-optimal routing for EVs, subject to specific characteristics such as the energy recuperation, battery capacity limitations and dynamic energy cost, is studied in a graph-theoretic context and a heuristic algorithm is proposed to find the optimal path.

Minimizing the length of the path, (Siddiqi et al., 2011) studies a multi-constrained route optimization problem for EVs. Applying penalty function method, the problem is transformed into an unconstrained optimization problem, then a particle swarm optimization algorithm is proposed to find a suboptimal solution. In (Khuller et al., 2011) algorithms for several routing problems including the shortest path and the Traveling Salesman Problem (TSP) are proposed by incorporating all costs in terms of gas prices. The goal is to find the cheapest route for an origin-destination pair or the cheapest tour in the case of TSP. For the shortest path problem, equivalent to our single vehicle routing problem, a Dynamic Programming (DP) algorithm is proposed to find a least cost path from an origin to a destination in a network with inhomogeneously priced refueling stations. The same problem is revisited by authors in (Sweda and Klabjan., 2012) where the recharging cost is assumed as a nonlinear function of the battery charging level. Again, the goal is to find a minimum-cost path for an

EV. Discretizing the state space, a DP based algorithms is proposed to determine the optimal path. (Schneider et al., 2014) introduces the Electric Vehicle Routing Problem with Time Windows and recharging stations (E-VRPTW). In this paper, the charging scheme simply forces vehicles to be always fully recharged. In (Worley et al., 2012), the problem of locating charging stations and also determining optimal routes for commercial electric vehicles is formulated as an integer programming problem. Combinatorial optimization methods for different aspects of EV management such as energy-efficient routing and facility location problems are studied in (Touati-Moungla and Jost, 2012). In recent work, (He et al., 2014) investigates the user-optimal network flow equilibrium with different scenarios for flow dependency of energy consumption of Battery Electric Vehicles (BEVs).

In this thesis, our objective is to study a vehicle total traveling time minimization problem (including both the time on paths and at charging stations), where an energy constraint is considered so that the vehicle is not allowed to run out of power before reaching its destination. We view this as a network routing problem where vehicles control not only their routes but also times to recharge at various nodes in the network. We First investigate the problem in a network with homogeneous charging nodes. Then, we study a more complicated case in which charging nodes are inhomogeneous meaning that the charging rate is a node-dependent parameter. We address the problem from two different point of views: user-centric vs system-centric. For the user-centric case, first we formulate the problem as a MINLP which is the exact formulation. We then reduce the problem's complexity by decomposing it into two simpler LP problems. Correspondingly, we separately determine route selection through a Linear Programming (LP) problem and then recharging amounts through another LP problem. Since we do not impose full recharging constraints, the solutions obtained are more general than, for example, in (Schneider et al., 2014) and recover full recharging when this is optimal. For the network with homogeneous charging nodes, this problem decomposition doesn't affect the optimality of the solution, however for the network with inhomogeneous charging nodes, the solution of the decomposed LP for route selection is sub-optimal in general. Next we study the system-centric problem in which a traffic flow model is used to incorporate congestion effects. This system-wide optimization problem appears to have not yet attracted much attention. By grouping vehicles into "subflows" we are once again able to decompose the problem into route selection and recharging amount determination, although we can no longer reduce the former problem to an LP. Again, the global optimality of the solution of the decomposed route selection problem is not guaranteed for the network with inhomogeneous charging modes. Moreover, we provide an alternative flow-based formulation such that each subflow is not required to follow a single end-to-end path, but may be split into an optimally determined set of paths. This formulation reduces the computational complexity of the MINLP problem by orders of magnitude with numerical results showing little or no loss in optimality. We further study the "price of anarchy" for the multi-vehicle routing problem so as to determine the difference in performance between selfish routing and system-optimal routing. We then address the issue of selecting the number of subflows, seeking to keep it as small as possible. Finally, We relax the assumption that every arriving vehicle is an EV and consider both EVs with energy constraints and Non-Electric Vehicles (NEVs) in the inflow to the network. We again seek to optimize a system-centric objective by optimally routing NEVs and EVs along with an optimal policy for charging EVs along the way if needed.

The structure of the chapter is as follows. In Section 4.2, we introduce and address the single-vehicle routing problem and identify properties which lead to its decomposition. In Section 4.3, the multi-vehicle routing problem is formulated, first as a MINLP and then as an alternative flow optimization problem. We also investigate the price of anarchy for this problem and provide simulation examples illustrating our approach and giving insights on the relationship between recharging speed and optimal routes. In Section 4.4 we define a criterion and a systematic procedure for the proper selection of the number of subflows. In Section 4.5, the multi-vehicle routing problem in revisited when the inflow to the netwrok contains both EVs and non-EVs.

4.2 Single Vehicle Routing

The single vehicle routing problem represents the "user-centric" point of view in which the objective is to find the optimal path and charging policy for a single EV minimizing its total traveling time. We consider a traffic network modeled as a directed graph $G = (\mathcal{N}, \mathcal{A})$ with $\mathcal{N} = \{1, \dots, n\}$ and $|\mathcal{A}| = m$ (see Fig. 4·1). Node $i \in \mathcal{N}/\{n\}$ represents a charging station and $(i, j) \in \mathcal{A}$ is an arc (link) connecting node i to j. We assume for simplicity that all nodes have a charging capability, although this is not necessary (we can model the network with some nodes without charging capability as a network with inhomogeneous charging nodes). We also define $I(i)$ and $O(i)$ to be the set of start nodes (respectively, end nodes) of arcs that are incoming to (respectively, outgoing from) node i, that is, $I(i) = \{j \in \mathcal{N} | (j, i) \in \mathcal{A}\}$ and $O(i) = \{j \in \mathcal{N} | (i, j) \in \mathcal{A}\}$.

We are first interested in a single-origin-single-destination vehicle routing problem. Nodes 1 and n respectively are defined to be the origin and destination. For each arc $(i, j) \in \mathcal{A}$, there are two cost parameters: the required traveling time τ_{ij} and the required energy consumption e_{ij} on this arc. Note that $\tau_{ij} > 0$ (if nodes i and j are not connected, then $\tau_{ij} = \infty$), whereas e_{ij} is allowed to be negative due to a BPV's potential energy recuperation effect (Artmeier et al., 2010). Letting the vehicle's charge capacity be B, we assume that $e_{ij} < B$ for all $(i, j) \in \mathcal{A}$. Since we are considering a single vehicle's behavior, we assume that it will not affect the overall network's traffic state, therefore, τ_{ij} and e_{ij} are assumed to be fixed depending on given traffic conditions at the time the single-vehicle routing problem is solved. Clearly, this cannot apply to the multi-vehicle case in the next section, where the decisions of multiple vehicle routes affect traffic conditions, thus influencing traveling times and energy consumption. Since the BPV has limited battery energy it may not be able to reach the destination without recharging. Thus, recharging amounts at charging nodes $i \in \mathcal{N}$ are also decision variables.

We denote the selection of arc (i, j) and energy recharging amount at node i by $x_{ij} \in \{0, 1\}$, $i, j \in \mathcal{N}$ and $r_i \geq 0$, $i \in \mathcal{N}/\{n\}$, respectively. Moreover, since we take into account the vehicle's energy constraints, we use E_i to represent the vehicle's residual battery energy at node i. Then, for all E_j, $j \in O(i)$, we have:

$$E_j = \begin{cases} E_i + r_i - e_{ij} & \text{if } x_{ij} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

which can also be expressed as

$$E_j = \sum_{i \in I(j)} (E_i + r_i - e_{ij}) x_{ij}, \quad x_{ij} \in \{0, 1\}$$

The problem objective is to determine a path from 1 to n, as well as recharging amounts, so as to minimize the total elapsed time for the vehicle to reach the destination. Fig. 4·1 is a sample network for this vehicle routing problem. We formulate a MINLP problem as follows:

$$\min_{x_{ij}, r_i,\ i,j \in \mathcal{N}} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \tau_{ij} x_{ij} + \sum_{i=1}^{n} \sum_{j=1}^{n} r_i g x_{ij} \tag{4.2}$$

$$s.t. \quad \sum_{j \in O(i)} x_{ij} - \sum_{j \in I(i)} x_{ji} = b_i, \quad \text{for each } i \in \mathcal{N} \tag{4.3}$$

$$b_1 = 1,\ b_n = -1,\ b_i = 0, \text{ for } i \neq 1, n \tag{4.4}$$

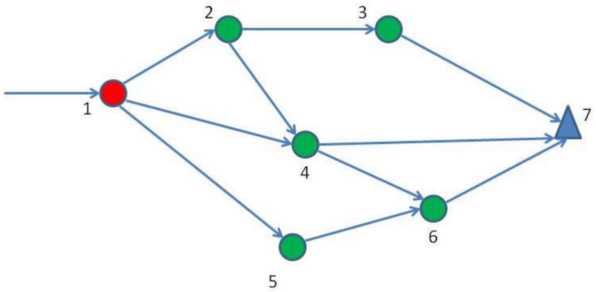


Figure 4·1: A 7-node network example for routing with recharging nodes.

$$E_j = \sum_{i \in I(j)} (E_i + r_i - e_{ij}) x_{ij}, \text{ for } j = 2, \ldots, n \tag{4.5}$$

$$0 \le E_i \le B, \quad E_1 \text{ given}, \text{ for each } i \in \mathcal{N} \tag{4.6}$$

$$x_{ij} \in \{0, 1\}, \quad r_i \ge 0 \tag{4.7}$$

where g is the charging time per energy unit, i.e., the reciprocal of a fixed charging rate. Here, we assume *homogeneous* charging nodes, i.e., the charging rate, g, is identical for all nodes. The constraints (4.3)-(4.4) stand for the flow conservation, which implies that only one path starting from node i can be selected, i.e., $\sum_{j \in O(i)} x_{ij} \le 1$ (Bertsimas and Tsitsiklis, 1997). It is easy to check that this also implies $x_{ij} \le 1$ for all i, j since $b_1 = 1$, $I(1) = \varnothing$. Constraint (4.5) represents the vehicle's energy dynamics. Finally, (4.6) indicates that the vehicle cannot run out of energy before reaching a node or exceed a given capacity B. All other parameters are predetermined according to the network topology. The more general case of inhomogeneous charging nodes will be addressed in Section 4.6.

4.2.1 Properties

Rather than directly tackling the MINLP problem (4.2)-(4.7), we derive some key properties which will enable us to simplify the solution procedure. The main difficulty in this problem lies in the coupling of the decision variables, x_{ij} and r_i, in (4.5).

The following lemma will enable us to exclude r_i from the objective function by showing that the difference between the total recharging energy and the total energy consumption while traveling is given only by the difference between the vehicle's residual energy at the destination and at the origin.

Lemma 1: Given (4.2)-(4.7),

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(r_i x_{ij} - e_{ij}x_{ij}) = E_n - E_1 \tag{4.8}$$

Proof: From (4.5), we sum up both sides to get:

$$\sum_{j=2}^{n} E_j - \sum_{j=2}^{n}\sum_{i\in I(j)} E_i x_{ij} = \sum_{j=2}^{n}\sum_{i\in I(j)} (r_i - e_{ij})x_{ij} \tag{4.9}$$

Moreover, we can write

$$\sum_{j=2}^{n}\sum_{i\in I(j)} E_i x_{ij} = \sum_{i\in I(2)} E_i x_{i2} + \cdots + \sum_{i\in I(n)} E_i x_{in}$$

representing the sum of E_i on the selected path from node 1 to n, excluding E_n. On the other hand, from (4.5) we have $E_i = 0$ for any node i not selected on the path. Therefore, $\sum_{j=2}^{n} E_j$ is the sum of E_i on the selected path from node 1 to n, excluding E_1. It follows that

$$\sum_{j=2}^{n} E_j - \sum_{j=2}^{n}\sum_{i\in I(j)} E_i x_{ij} = E_n - E_1 \tag{4.10}$$

Returning to (4.9), we use (4.10) and observe that all terms in the double sum $\sum_{i=1}^{n}\sum_{j=1}^{n}(r_i - e_{ij})x_{ij}$ are zero except for those with $i \in I(j)$, we get

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(r_i - e_{ij})x_{ij} = \sum_{j=2}^{n}\sum_{i\in I(j)} (r_i - e_{ij})x_{ij} = \sum_{j=2}^{n} E_j - \sum_{j=2}^{n}\sum_{i\in I(j)} E_i x_{ij} = E_n - E_1$$

which proves the lemma.■

In view of Lemma 1, we can replace $\sum_{i=1}^{n}\sum_{j=1}^{n} r_i g x_{ij}$ in (4.2) by $(E_n - E_1)g +$

$\sum_{i=1}^{n}\sum_{j=1}^{n} e_{ij}gx_{ij}$ and eliminate the presence of r_i, $i = 2, \ldots, n-1$, from the objective function. Note that E_1 is given, leaving us only with the task of determining the value of E_n. Now, let us investigate the recharging energy amounts r_i^*, $i = 1, \ldots, n-1$, in an optimal policy. There are two possible cases: (i) $\sum_i r_i^* > 0$, i.e., the vehicle has to get recharged at least once, and (ii) $\sum_i r_i^* = 0$, i.e., $r_i^* = 0$ for all i and the vehicle has adequate energy to reach the destination without recharging. For Case (i), we establish the following lemma.

Lemma 2: If $\sum_i r_i^* > 0$ in the optimal routing policy, then $E_n^* = 0$.

Proof: We use a contradiction argument. Assume we have already achieved an optimal route where $E_n^* > 0$ and the objective function is $J^* = \sum_{i \in P}(\tau_{i,i+1} + r_i^* g)$ for an optimal path denoted by P. Without loss of generality, we re-index nodes so that we may write $P = \{1, \ldots, n\}$. Then, each $i \in P$ such that $i < n$ on this optimal path satisfies:

$$E_{i+1}^* = E_i^* + r_i^* - e_{i,i+1} \tag{4.11}$$

Consider first the case where $r_{n-1}^* > 0$. Let us perturb the current policy as follows: $r_{n-1}^{'} = r_{n-1}^* - \Delta$, and $r_i^{'} = r_i^*$ for all $i < n-1$, where $\Delta > 0$. Then, from (4.11), we have

$$E_n^* = E_1 + \sum_{i=1}^{n-1}(r_i^* - e_{i,i+1})$$

Under the perturbed policy,

$$E_n^{'} = E_1 + \sum_{i=1}^{n-1}(r_i^{'} - e_{i,i+1}) = E_1 + \sum_{i=1}^{n-1}(r_i^* - e_{i,i+1}) - \Delta = E_n^* - \Delta$$

$$E_i^{'} = E_i^*, \text{ for all } i < n$$

and, correspondingly,

$$J^{'} = \sum_{i=1}^{n-1}(\tau_{i,i+1} + r_i^{'} g) = \sum_{i=1}^{n-1}(\tau_{i,i+1} + r_i^* g) - \Delta g = J^* - \Delta g$$

Since $E_n^* > 0$, we may select $\Delta > 0$ sufficiently small so that $E_n^{'} > 0$ and the perturbed policy is still feasible. However, $J^{'} = J^* - \Delta g < J^*$, which leads to a contradiction to the assumption that the original path was optimal.

Next, consider the case where $r_{n-1}^* = 0$. Then, due to $E_n^* > 0$ and $e_{i,i+1} > 0$ for all $i \in P$, we can always find some $j \in P$, $j < n$ such that $E_j^* > 0$, $r_{j-1}^* > 0$ and $r_k^* = 0$ for $k \geqslant j$. Thus, still due to (4.11), we have

$$E_j^* = E_n^* + \sum_{k=j}^{n-1} e_{k,k+1} > 0$$

At this time, since $r_{j-1}^* > 0$, the argument is similar to the case $r_{n-1}^* > 0$, leading again to the same contradiction argument and the lemma is proved. ■

Turning our attention to Case (*ii*) where $r_i^* = 0$ for all $i \in \{1, \ldots, n\}$, observe that the problem (4.2) can be transformed into

$$\min_{x_{ij},\ i,j\in\mathcal{N}} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} \tau_{ij} x_{ij} \tag{4.12}$$

$$s.t. \quad \sum_{j\in O(i)} x_{ij} - \sum_{j\in I(i)} x_{ji} = b_i, \quad \text{for each } i \in \mathcal{N}$$

$$b_1 = 1,\ b_n = -1,\ b_i = 0,\ \text{for } i \neq 1, n$$

$$E_j = \sum_{i\in I(j)} (E_i - e_{ij}) x_{ij},\ \text{for } j = 2, \ldots, n \tag{4.13}$$

$$0 \le E_i \le B, \quad E_0 \text{ given},\ \text{for each } i \in \mathcal{N}$$

$$x_{ij} \in \{0, 1\} \tag{4.14}$$

In this case, the constraint (4.13) gives

$$\sum_{j=2}^{n} E_j - \sum_{j=2}^{n}\sum_{i\in I(j)} E_i = -\sum_{j=2}^{n}\sum_{i\in I(j)} e_{ij} x_{ij}$$

Using (4.10) and $E_i \geq 0$, we have

$$E_n = E_1 - \sum_{j=2}^{n} \sum_{i \in I(j)} e_{ij} x_{ij} \geq 0$$

and it follows that

$$\sum_{i=1}^{n} \sum_{j=1}^{n} e_{ij} x_{ij} \leq E_1 \tag{4.15}$$

With (4.15) in place of (4.13), the determination of x_{ij}^* boils down to an integer *linear* programming problem in which only variables x_{ij}, $i, j \in \mathcal{N}$, are involved, a much simpler problem.

We are normally interested in Case (i), where some recharging decisions must be made, so let us assume the vehicle's initial energy is not large enough to reach the destination. Then, in view of Lemmas 1 and 2, we have the following theorem.

Theorem 1: If $\sum_i r_i^* > 0$ in the optimal policy, then x_{ij}^*, $i, j \in \mathcal{N}$, in the original problem (4.2) can be determined by solving a linear programming problem:

$$\min_{x_{ij},\ i,j \in \mathcal{N}} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} (\tau_{ij} + e_{ij} g) x_{ij} \tag{4.16}$$

$$s.t. \quad \sum_{j \in O(i)} x_{ij} - \sum_{j \in I(i)} x_{ji} = b_i, \quad \text{for each } i \in \mathcal{N}$$

$$b_1 = 1,\ b_n = -1,\ b_i = 0,\ \text{for } i \neq 1, n$$

$$0 \leq x_{ij} \leq 1$$

Proof: Given Lemmas 1 and 2, we know that the optimal solution satisfies $\sum_i \sum_j r_i^* x_{ij}^*$ $= \sum_i \sum_j e_{ij} x_{ij}^* - E_1$. Consequently, we can change the objective (4.2) to the form below without affecting optimality:

$$\min_{x_{ij},\ i,j \in \mathcal{N}} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} (\tau_{ij} + e_{ij} g) x_{ij} - E_1 g$$

Since r_i no longer appears in the objective function and is only contained in the energy dynamics (4.5), we can choose any r_i satisfying the constraints (4.5)-(4.6) without affecting the optimal objective function value. Therefore, x_{ij}^* can be determined by the following problem:

$$\begin{aligned} \min_{x_{ij},\ i,j\in\mathcal{N}} \quad & \sum_{i=1}^{n}\sum_{j=1}^{n}(\tau_{ij}+e_{ij}g)x_{ij} - E_1 g \\ s.t. \quad & \sum_{j\in O(i)} x_{ij} - \sum_{j\in I(i)} x_{ji} = b_i, \quad \text{for each } i\in\mathcal{N} \\ & b_1 = 1,\ b_n = -1,\ b_i = 0,\ \text{for } i\neq 1,n \\ & x_{ij}\in\{0,1\} \end{aligned}$$

which is a typical shortest path problem formulation. Moreover, by the property of minimum cost flow problems (Hillier and Lieberman, 2005), the above integer programming problem is equivalent to the linear programming problem with the integer restriction of x_{ij} relaxed. Finally, since E_1 is given, the problem reduces to (4.16), which proves the theorem. ■

Note that using Theorem 1, an optimal path is determined by solving a LP problem and since this is a convex optimization problem (Boyd and Vandenberghe, 2004), the solution is the global optimum.

Transformation of the single-vehicle routing problem to a shortest path problem

We can show that the optimal path obtained using LP formulation (4.16), is equivalent to the shortest path on the graph weighted by the traveling costs for each arc (i, j). The weight of arc (i, j) is defined as $w_{ij} = \tau_{ij} + e_{ij}g$ which is the total elapsed time for traveling through link (i, j) and getting $r_i = e_{ij}$ unit of charge at node i. A path from the origin to the destination node is denoted by p with an associated cost defined as $W_p = \sum_{(i,j)\in p} w_{ij}$. Clearly, for an EV with no initial energy at the origin node, i.e., $E_1 = 0$, the total elapsed time to travel from the origin node to the destination through path p is W_p (Note that the EV will receive $\sum_{(i,j)\in p} e_{ij}$ unit of energy while traveling and $E_n = 0$ at the destination node). One can observe that W_p is identical to the cost function in (4.16) when corresponding x_{ij} s, $\forall(i,j) \in p$s are equal to one. Similarly, when we find the shortest path, p^*, the optimal cost is calculated as: $W_{p^*} - E_1 * g$.

4.2.2 Determination of optimal recharging amounts r_i^*

Once we determine the optimal route, P, in (4.16), it is relatively easy to find a feasible solution for r_i, $i \in P$, to satisfy the constraint (4.5), which is obviously non-unique in general. Then, we can introduce a second objective into the problem, i.e., the minimization of charging costs on the selected path, since charging prices normally vary over stations. As before, we re-index nodes and define $P = \{1, ..., n\}$. We denote the charging price at node i by p_i. Once an optimal route is determined, we seek to control the energy recharging amounts r_i to minimize the total charging cost dependent on p_i, $i \in \mathcal{N}/\{n\}$. This can be formulated as a multistage optimal control problem:

$$\begin{gathered}
\min_{r_i,\ i\in P} \quad \sum_{i\in P} p_i r_i \qquad (4.17)\\
s.t. \quad E_{i+1} = E_i + r_i - e_{i,i+1}\\
0 \le E_i \le B, \quad E_1 \text{ given}\\
r_i \ge 0 \text{ for all } i \in \mathcal{N}
\end{gathered}$$

This is a simple linear programming problem where E_i and r_i are both decision variables.

Finally, we note that Theorem 1 holds under the assumption that charging nodes are homogeneous in terms of charging speeds (i.e., the charging rate $1/g$ is fixed). However, our analysis allows for inhomogeneous charging prices. The case of node-dependent charging rates will be addressed in Section 4.6 and it will be shown that we still can do the problem decomposition, although the global optimality is not be guaranteed in general. Thus, alternative techniques need to be explored in order to reduce the computational complexity of the problem in that case.

It is important to ensure that a solution to the overall problem is computationally efficient, since it may have to be repeatedly obtained during the course of a vehicle's trip: although we treat the state variable E_i as deterministic, in reality there is noise in the process which may force a re-evaluation of routing and charging at each node when E_{i+1} is observed and may satisfy $E_{i+1} = E_i + r_i - e_{i,i+1} + w_{ij}$ where w_{ij} is a random variable. In this case, one can re-solve the optimal routing and charging problem for the vehicle with new initial conditions at node $i+1$, which is possible as long as we only have to deal with the simple problems (4.16) and (4.17).

4.3 Multiple Vehicle Routing

Next, we investigate the system-centric problem, referred to as the multiple-vehicle routing problem, in a network with homogeneous charging nodes. As opposed to the user-centric policy, here we determine the routing and charging policies so as to optimize a *system-wide objective*. Thus, as discussed in Section 4.2, the first technical difficulty here is the need to incorporate the effect of traffic congestion on both traveling time and energy consumption; therefore, the variables τ_{ij} and e_{ij} no longer have fixed values. A second difficulty is the implementation of an optimal routing policy, which requires signaling mechanisms and possibly incentive structures to enforce desired routes assigned to vehicles. This raises a number of additional research issues which are beyond the scope of this thesis and likely to be addressed by the advent of Connected Automated Vehicles (CAVs).

If we proceed as in the single vehicle case, i.e., determining a path selection through

x_{ij}^k, $i, j \in \mathcal{N}$, and recharging amounts r_i^k, $i \in \mathcal{N}/\{n\}$ for all vehicles $k = 1, \ldots, K$, for some K, then the dimensionality of the solution space is prohibitive. Moreover, the inclusion of traffic congestion effects introduces additional nonlinearities in the dependence of the travel time τ_{ij} and energy consumption e_{ij} on the traffic flow through arc (i, j), which now depend on $x_{ij}^1, \cdots, x_{ij}^K$. Instead, we will proceed by grouping subsets of vehicles into N "subflows" where N may be selected to render the problem manageable (see Section 4.4).

Let all vehicles enter the network at the origin node 1 and let R denote the rate of vehicles arriving at this node. Viewing vehicles as defining a *flow*, we divide them into N *subflows* (we will discuss the effect of N in Section 4.3.4), each of which may be selected so as to include the same type of homogeneous vehicles (e.g., vehicles with the same initial energy). Thus, all vehicles in the same subflow follow the same routing and recharging decisions so that we only consider energy recharging at the subflow level rather than individual vehicles. Note that asymptotically, as $N \to \infty$, we can recover routing at the individual vehicle level.

Clearly, not all vehicles in our system are EVs, in which case these can be treated as uncontrollable interfering traffic and are accommodated in our analysis as long as their flow rates are known. For simplicity, we will assume here that every arriving vehicle is an EV and joins a subflow. However, we will show in Section 4.5 how the problem can be solved by optimizing over both EVs and non-EVs.

Our objective is to determine optimal routes and energy recharging amounts for each subflow of vehicles so as to minimize the total elapsed time of these vehicle flows traveling from the origin to the destination. The decision variables consist of $x_{ij}^k \in \{0, 1\}$ for all arcs (i, j) and subflows $k = 1, \ldots, N$, as well as charging amounts r_i^k for all nodes $i = 1, \ldots, n-1$ and $k = 1, \ldots, N$. Given traffic congestion effects, the time and energy consumption on each arc depends on the values of x_{ij}^k and the fraction of the total flow rate R associated with each subflow k; the simplest such flow allocation (which we will adopt) is one where each subflow is assigned R/N. Let $\mathbf{x_{ij}} = (x_{ij}^1, \cdots, x_{ij}^N)^T$ and $\mathbf{r_i} = (r_i^1, \cdots, r_i^N)^T$. Then, we denote the traveling time (delay) a vehicle will experience through link (i,j) by some nonlinear function $\tau_{ij}(\mathbf{x_{ij}})$. The corresponding energy consumption of the kth vehicle subflow through link (i,j) is a nonlinear function denoted by $e_{ij}^k(\mathbf{x_{ij}})$. As already mentioned, $\tau_{ij}(\mathbf{x_{ij}})$ and $e_{ij}^k(\mathbf{x_{ij}})$ can also incorporate the influence of uncontrollable (non-EV) vehicle flows, which can be treated as parameters in these functions (we discuss this further in Section 4.5). Similar to the single vehicle case, we use E_i^k to represent the residual energy of subflow k at node i, given by the aggregated residual energy of all vehicles in the subflow. If the subflow does not go through node i, then $E_i^k = 0$. The problem formulation is as follows:

$$\min_{\mathbf{x_{ij}},\mathbf{r_i},\ i,j\in\mathcal{N}} \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N}\left(\tau_{ij}(\mathbf{x_{ij}})x_{ij}^k\frac{R}{N} + r_i^k g x_{ij}^k\right) \tag{4.18}$$

$$s.t. \text{ for each } k \in \{1,\dots,N\}:$$

$$\sum_{j\in O(i)} x_{ij}^k - \sum_{j\in I(i)} x_{ji}^k = b_i, \quad \text{for each } i \in \mathcal{N} \tag{4.19}$$

$$b_1 = 1,\ b_n = -1,\ b_i = 0, \text{ for } i \neq 1, n \tag{4.20}$$

$$E_j^k = \sum_{i\in I(j)} (E_i^k + r_i^k - e_{ij}^k(\mathbf{x_{ij}}))x_{ij}^k, \quad j = 2,\dots,n \tag{4.21}$$

$$E_1^k \text{ is given}, \quad E_i^k \geq 0, \quad \text{for each } i \in \mathcal{N} \tag{4.22}$$

$$x_{ij}^k \in \{0,1\}, \quad r_i^k \geq 0 \tag{4.23}$$

Obviously, this MINLP problem is difficult to solve. However, as in the single-vehicle case, we are able to establish some properties that will allow us to simplify it.

4.3.1 Properties

Even though the term $\tau_{ij}(\mathbf{x_{ij}})$ in the objective function is no longer linear in general, for each subflow k the constraints (4.19)-(4.23) are still similar to the single-vehicle case. Consequently, we can derive similar useful properties for this problem in the form of the following two lemmas.

Lemma 3: For each subflow $k = 1, \ldots, N$,

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(r_i^k - e_{ij}^k(\mathbf{x_{ij}}))x_{ij}^k = E_n^k - E_1^k \tag{4.24}$$

Proof: From (4.21), we sum up both sides of the equation as follows:

$$\begin{aligned}
&\text{For each } k: \\
&\sum_{j=2}^{n} E_j^k = \sum_{j=2}^{n}\sum_{i\in I(j)}(E_i^k + r_i^k - e_{ij}^k(\mathbf{x_{ij}}))x_{ij}^k \\
&\Longrightarrow \sum_{j=2}^{n} E_j^k - \sum_{j=2}^{n}\sum_{i\in I(j)} E_i^k x_{ij}^k = \sum_{j=2}^{n}\sum_{i\in I(j)}(r_i^k - e_{ij}^k(\mathbf{x_{ij}}))x_{ij}^k
\end{aligned} \tag{4.25}$$

Moreover, $\sum_{j=2}^{n}\sum_{i\in I(j)} E_i^k x_{ij}^k = \sum_{i\in I(j)} E_i^k \sum_{j=2}^{n} x_{ij}^k$ representing the sum of E_i^k on the selected path from node 1 to n. On the other hand, from (4.21) we have $E_i^k = 0$ for node i not selected on the route. Therefore,

$$\sum_{j=2}^{n} E_j^k - \sum_{j=2}^{n}\sum_{i\in I(j)} E_i^k x_{ij}^k = E_n^k - E_1^k$$

Back to (4.25),

$$\sum_{i}\sum_{j}(r_i^k - e_{ij}^k(\mathbf{x_{ij}}))x_{ij}^k = \sum_{j=2}^{n}\sum_{i\in I(j)}(r_i^k - e_{ij}^k(\mathbf{x_{ij}}))x_{ij}^k = \sum_{j=2}^{n} E_j^k - \sum_{j=2}^{n}\sum_{i\in I(j)} E_i^k x_{ij}^k = E_n^k - E_1^k$$

which proves the result.■

Similar to Lemma 2, we can determine E_n^{k*} when $\sum_i r_i^{k*} > 0$ by Lemma 4:

Lemma 4: If $\sum_{i=1}^{n} r_i^{k*} > 0$ in the optimal routing policy, then $E_n^{k*} = 0$ for all $k = 1, \ldots, N$.

Proof: Assume we have already achieved the optimal routes for these k vehicle subflows such that $E_n^{k*} > 0$ and the contribution of kth subflow to the objective function value

$$J_k^* = \sum_{i=1}^{n} \sum_{j=1}^{n} \tau_{ij}(\mathbf{x_{ij}}) x_{ij}^{k*} \frac{R}{N} + \sum_{i=1}^{n} \sum_{j=1}^{n} r_i^{k*} g x_{ij}^{k*}$$

Since only the second part of the objective function is dependent on r_i^k, we only need to concentrate on the value of $\sum_{i=1}^{n} \sum_{j=1}^{n} r_i^{k*} g x_{ij}^{k*}$. Then each $i < n$ on this route satisfies:

$$E_{i+1}^{k*} = E_i^{k*} + r_i^{k*} - e_{i,i+1} \tag{4.26}$$

where $e_{i,i+1}$ is the value of $e_{ij}^k(\mathbf{x_{ij}})$ on the determined route by x_{ij}^{k*} for all k. Now if $r_{n-1}^{k*} > 0$, then let us perturb the current policy by

$$\begin{aligned} r_{n-1}^{k'} &= r_{n-1}^{k*} - \Delta \\ r_i^{k'} &= r_i^{k*}, \text{ for all } i < n-1 \end{aligned}$$

where $\Delta > 0$. Then according to (4.26), under the perturbed policy

$$\begin{aligned} E_n^{k'} &= E_n^{k*} - \Delta \\ E_i^{k'} &= E_i^{k*}, \text{ for all } i < n \end{aligned}$$

and correspondingly $J_k^{'} = J_k^* - \Delta g$. Since $E_n^{k*} > 0$, then as long as we make Δ small enough such that $E_n^{k'} > 0$, the perturbed policy is still feasible. However, $J_k^{'}$ is smaller than J_k^*, which draws a contradiction to the assumption. Now if $r_{n-1}^{k*} = 0$, then due to $E_n^{k*} > 0$ and $e_{i,i+1} > 0$ for all i, we can always find some $j < n$ such that $E_j^{k*} > 0$,

$r_{j-1}^{k*} > 0$ and $r_l^{k*} = 0$ for $l > j$. Thus, still owing to (4.26), we have

$$E_j^{k*} = E_n^{k*} + \sum_{l=j}^{n-1} e_{l,l+1} > 0$$

At this time, since $r_{j-1}^{k*} > 0$, the argument is similar to the case $r_{n-1}^{k*} > 0$, in which the lemma can be justified by the contradiction argument. Consequently, the lemma is proven. ■

In view of Lemma 3, we can replace $\sum_{i=1}^{n} \sum_{j=1}^{n} r_i^k g x_{ij}^k$ in (4.18) by $(E_n^k - E_1^k)g + \sum_{i=1}^{n} \sum_{j=1}^{n} e_{ij}^k(\mathbf{x_{ij}}) g x_{ij}$ and eliminate, for all $k = 1, \ldots, N$, the presence of r_i^k, $i = 1, \ldots, n-1$, from the objective function similar to the single-vehicle case. Since E_1^k is given, this leaves only the task of determining the value of E_n^k. There are two possible cases: (i) $\sum_i r_i^{k*} > 0$, i.e., the kth vehicle subflow has to get recharged at least once, and (ii) $\sum_i r_i^{k*} = 0$, i.e., $r_i^{k*} = 0$ for all i and the kth vehicle subflow has adequate energy to reach the destination without recharging.

Similar to the derivation of (4.15), Case (ii) results in a new constraint $\sum_i \sum_j e_{ij}^k(\mathbf{x_{ij}}) x_{ij}^k \leq E_1^k$ for subflow k. However, since $e_{ij}^k(\mathbf{x_{ij}})$ now depends on all $x_{ij}^1, \ldots, x_{ij}^N$, the problem (4.18)-(4.23) with all $r_i^k = 0$ is not as simple to solve as was the case with (4.12)-(4.14). Let us instead concentrate on the more interesting Case (i) for which Lemma 4 applies and we have $E_n^{k*} = 0$. Therefore, along with Lemma 3, we have for each $k = 1, \ldots, N$:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} r_i^k x_{ij}^k = \sum_{i=1}^{n} \sum_{j=1}^{n} e_{ij}^k(\mathbf{x_{ij}}) x_{ij}^k - E_1^k$$

Then, proceeding as in Theorem 1, we can replace the original objective function (4.18) and obtain the following new problem formulation to determine x_{ij}^{k*} for all

$i, j \in \mathcal{N}$ and $k = 1, \ldots, N$:

$$\min_{\mathbf{x_{ij}},\, i,j\in\mathcal{N}} \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N}\left(\tau_{ij}(\mathbf{x_{ij}})x_{ij}^{k}\frac{R}{N} + e_{ij}^{k}(\mathbf{x_{ij}})gx_{ij}^{k}\right) \tag{4.27}$$

$$s.t. \text{ for each } k \in \{1, \ldots, N\} :$$

$$\sum_{j\in O(i)} x_{ij}^{k} - \sum_{j\in I(i)} x_{ji}^{k} = b_i, \quad \text{for each } i \in \mathcal{N}$$

$$b_1 = 1,\, b_n = -1,\, b_i = 0, \text{ for } i \neq 1, n$$

$$x_{ij}^{k} \in \{0, 1\}$$

Since the objective function is no longer necessarily linear in x_{ij}^{k}, (4.27) cannot be further simplified into an LP problem as in Theorem 1. The computational effort required to solve this problem heavily depends on the dimensionality of the network and the number of subflows. Nonetheless, from the transformed formulation above, we are still able to separate the determination of routing variables x_{ij}^{k} from recharging amounts r_i^k. Similar to the single-vehicle case, once the routes are determined, we can obtain any r_i^k satisfying the energy constraints (4.21)-(4.22) such that $E_n^k = 0$, thus preserving the optimality of the objective value. To further determine r_i^{k*}, we can introduce a second level optimization problem similar to the single-vehicle case in (4.17). Next, we will present an alternative formulation for the original problem (4.18)-(4.23) which leads to a computationally simpler solution approach.

4.3.2 Flow control formulation

We begin by relaxing the binary variables in (4.23) and letting $0 \leq x_{ij}^{k} \leq 1$. Thus, we switch our attention from determining a single path for any subflow k to several possible paths by treating x_{ij}^{k} as the normalized vehicle flow on arc (i, j) for the kth subflow. This is in line with many network routing algorithms in which fractions x_{ij} of entities are routed from a node i to a neighboring node j using appropriate schemes ensuring that, in the long term, the fraction of entities routed on (i, j) is indeed x_{ij} (Gallager, 1977). Following this relaxation, the objective function in (4.18) is changed to:

$$\min_{\mathbf{x_{ij}},\mathbf{r_i},\ i,j\in\mathcal{N}} \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N}\tau_{ij}(\mathbf{x_{ij}})x_{ij}^{k}\frac{R}{N}+\sum_{i=1}^{n}\sum_{k=1}^{N}r_i^k g$$

Moreover, the energy constraint (4.21) needs to be adjusted accordingly. Let E_{ij}^k represent the fraction of residual energy of subflow k associated with the x_{ij}^k portion of the vehicle flow exiting node i. Therefore, the constraint (4.22) becomes $E_{ij}^k \geq 0$. We can now capture the relationship between the energy associated with subflow k and the vehicle flow as follows:

$$\left[\sum_{h\in I(i)}(E_{hi}^k - e_{hi}^k(\mathbf{x_{ij}})) + r_i^k\right]\frac{x_{ij}^k}{\sum_{h\in I(i)} x_{hi}^k} = E_{ij}^k \tag{4.28}$$

$$\frac{E_{ij}^k}{\sum_{j\in O(i)} E_{ij}^k} = \frac{x_{ij}^k}{\sum_{j\in O(i)} x_{ij}^k} \tag{4.29}$$

In (4.28), the energy values of different vehicle flows entering node i are aggregated and the energy corresponding to each portion exiting a node, E_{ij}^k, $j \in O(i)$, is proportional to the corresponding fraction of vehicle flows, as expressed in (4.29). Clearly, this aggregation of energy leads to an approximation, since one specific vehicle flow may need to be recharged in order to reach the next node in its path, whereas another might have enough energy without being recharged. This approximation foregoes controlling recharging amounts at the individual vehicle level and leads to approximate solutions of the original problem (4.18)-(4.23). Several numerically based comparisons are provided in the next section showing little or no loss of optimality relative to the solution of (4.18).

Adopting this formulation with $x_{ij}^k \in [0, 1]$ instead of $x_{ij}^k \in \{0, 1\}$, we obtain the following simpler nonlinear programming problem (NLP):

$$\min_{\mathbf{x_{ij}},\mathbf{r_i},\ i,j\in\mathcal{N}} \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N}\tau_{ij}(\mathbf{x_{ij}})x_{ij}^k\frac{R}{N}+\sum_{i=1}^{n}\sum_{k=1}^{N}r_i^k g \tag{4.30}$$

$$s.t.\text{ for each } k\in\{1,\dots,N\}:$$

$$\sum_{j\in O(i)}x_{ij}^k-\sum_{j\in I(i)}x_{ji}^k=b_i,\quad\text{for each } i\in\mathcal{N} \tag{4.31}$$

$$b_1=1,\ b_n=-1,\ b_i=0,\ \text{for } i\neq 1,n$$

$$\left[\sum_{h\in I(i)}(E_{hi}^k-e_{hi}^k(\mathbf{x_{ij}}))+r_i^k\right]\frac{x_{ij}^k}{\sum_{h\in I(i)}x_{hi}^k}=E_{ij}^k \tag{4.32}$$

$$\frac{E_{ij}^k}{\sum_{j\in O(i)}E_{ij}^k}=\frac{x_{ij}^k}{\sum_{j\in O(i)}x_{ij}^k} \tag{4.33}$$

$$E_{ij}^k\geq 0, \tag{4.34}$$

$$0\leq x_{ij}^k\leq 1,\quad r_i^k\geq 0 \tag{4.35}$$

As in our previous analysis, we are able to eliminate $\mathbf{r_i}$ from the objective function in (4.30) as follows.

Lemma 5: For each subflow $k=1,\dots,N$,

$$\sum_{i=1}^{n}r_i^k=\sum_{i=1}^{n}\sum_{j=1}^{n}e_{ij}^k(\mathbf{x_{ij}})+\sum_{i\in I(n)}E_{in}^k-\sum_{i\in O(1)}E_{1i}^k$$

Proof: Summing (4.32) over all $i=1,\dots,n$ gives

$$\sum_{i=1}^{n}r_i^k=\sum_{i=1}^{n}\sum_{j=1}^{n}e_{ij}^k(\mathbf{x_{ij}})+\sum_{i=1}^{n}\sum_{j\in O(i)}E_{ij}^k-\sum_{i=1}^{n}\sum_{h\in I(i)}E_{hi}^k$$

and using (4.31),(4.33), we get

$$\sum_{i=1}^{n}r_i^k=\sum_{i=1}^{n}\sum_{j=1}^{n}e_{ij}^k(\mathbf{x_{ij}})+\sum_{i\in I(n)}E_{in}^k-\sum_{i\in O(1)}E_{1i}^k$$

which proves the lemma. ■

Similar to Lemma 3, we can easily see that if $\sum_i r_i^{k*} > 0$ under an optimal routing policy, then $\sum_{i \in I(n)} E_{in}^{k*} = 0$. In addition, $\sum_{i \in O(1)} E_{1i}^k = E_1^k$, which is given. We can now transform the objective function (4.30) into (4.36) and determine the optimal routes x_{ij}^{k*} by solving the following NLP:

$$\min_{\substack{\mathbf{x_{ij}} \\ i,j \in \mathcal{N}}} \quad \sum_{k=1}^{N} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \left[\tau_{ij}(\mathbf{x_{ij}}) x_{ij}^k \frac{R}{N} + e_{ij}^k(\mathbf{x_{ij}}) g \right] - E_1^k \right) \tag{4.36}$$

$$s.t. \text{ for each } k \in \{1, \ldots, N\}:$$

$$\sum_{j \in O(i)} x_{ij}^k - \sum_{j \in I(i)} x_{ji}^k = b_i, \quad \text{for each } i \in \mathcal{N}$$

$$b_1 = 1,\ b_n = -1,\ b_i = 0, \text{ for } i \neq 1, n$$

$$0 \leq x_{ij}^k \leq 1$$

Note that in the above formulation, the nonlinearity appears in the objective function due to the traffic congestion effect on traveling time and energy consumption. Thus, if $\tau_{ij}(\mathbf{x_{ij}}) x_{ij}^k \frac{R}{N}$ and $e_{ij}^k(\mathbf{x_{ij}})$ are convex functions, the NLP is a convex optimization problem and the global optimum can be found generally fast. Once we find the optimal routes, the values of r_i^k, $i = 1, \ldots, n$, $k = 1, \ldots, N$, can be determined so as to satisfy the energy constraints (4.32)-(4.34), and they are obviously not unique. We may then proceed with a second-level optimization problem to determine optimal values similar to Section 4.2.2.

4.3.3 Objective function selection

The selection of $\tau_{ij}(\mathbf{x_{ij}})$ in either (4.27) or (4.36) is based on models originating in the traffic engineering literature. Here, we use a commonly used relationship between speed and density of a vehicle flow as in (Ho and Ioannou, 1996), (Kuhne and Rodiger, 1991), (Haefner and Li, 1998):

$$v(k(t)) = v_f\left(1 - \left(\frac{k(t)}{k_{jam}}\right)^p\right)^q \tag{4.37}$$

where v_f is the reference speed on the road without traffic, $k(t)$ represents the density of vehicles on the road at time t and k_{jam} the saturated density for a traffic jam. Note that we can replace $k(t)/k_{jam}$ in (4.37) with $f(t)/f_{jam}$, where $f(t)$ is the vehicle flow on the road at time t and f_{jam} represents the maximum capacity of the road. The parameters p and q are empirically identified for actual traffic flows. Given a network topology (i.e., a road map), the distances d_{ij} between nodes and the capacity of links, f^{ij}_{jam}, are known. Let us assume EVs enter the network at a rate of R veh./min. We then evenly divide the EV inflow into N subflows and the total flow entering link (i,j) becomes $f_{ij} = \sum_k x^k_{ij}\dfrac{R}{N}$. Then, the time a vehicle spends on link (i,j) becomes

$$\tau_{ij}(\mathbf{x_{ij}}) = \frac{d_{ij}}{v_f(1 - (\frac{f_{ij}}{f^{ij}_{jam}})^p)^q} \tag{4.38}$$

In Chapter 6 we show how to estimate the delay function using real traffic data.

Note that in order to prevent the inflow entering each link from exceeding its capacity, we add the following inequality constraint to the problem formulation (4.27) and (4.36):

$$\sum_k x^k_{ij}\frac{R}{N} \leq f^{ij}_{jam} \tag{4.39}$$

As for $e^k_{ij}(\mathbf{x_{ij}})$, we assume the energy consumption rates of subflows on link (i,j) are all identical, proportional to the distance between nodes i and j, giving $e^k_{ij}(\mathbf{x_{ij}}) = ed_{ij}R/N$.

Therefore, we aim to solve the multi-vehicle routing problem using (4.27) which

Table 4.1: d_{ij} values for network of Fig. 4·1 (*miles*)

d_{12}	d_{14}	d_{15}	d_{23}	d_{24}	d_{46}	d_{56}	d_{37}	d_{47}	d_{67}
5	6.2	7	3.5	5	3.6	4.3	6	6	4

in this case becomes:

$$\min_{\substack{x_{ij}^k \\ i,j \in \mathcal{N}}} \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N}\left(\frac{d_{ij}x_{ij}^k\frac{R}{N}}{v_f(1-(\frac{R/N\sum_k x_{ij}^k}{f_{jam}^{ij}})^p)^q}+egd_{ij}\frac{R}{N}x_{ij}^k\right) \tag{4.40}$$

$$s.t. \text{ for each } k \in \{1,\dots,N\}:$$

$$\sum_{j\in O(i)} x_{ij}^k - \sum_{j \in I(i)} x_{ji}^k = b_i, \quad \text{for each } i \in \mathcal{N}$$

$$b_1 = 1,\ b_n = -1,\ b_i = 0,\ \text{for } i \neq 1, n$$

$$\sum_k x_{ij}^k \frac{R}{N} \leq f_{jam}^{ij},\ \forall (i,j) \in \mathcal{A}$$

$$x_{ij}^k \in \{0,1\}$$

4.3.4 Numerical Examples

For simplicity, we let $v_f = 1$ mile/min, $R = 1$ veh./min, $p = 2$, $q = 2$ and $eg = 1$ and $f_{jam}^{ij} = 1$ veh./min $\forall (i,j) \in \mathcal{A}$. The network topology used is that of Fig. 4·1, where the distance of each link is shown in Tab. 4.1. To solve the nonlinear binary programming problem (4.40), we use the optimization solver *Opti* (MATLAB toolbox for optimization). The results are shown in Tab. 4.2 for different values of $N = 1,\dots,30$. It can be observed that vehicles are mainly distributed through three routes and the traffic congestion effect makes the flow distribution differ from following the shortest path. The number of decision variables (hence, the solution search space) rapidly increases with the number of subflows. However, looking at Fig. 4·2 which gives the performance in terms of our objective function in (4.40) as a function of the number of subflows, one can observe that the optimal objective value quickly converges with no significant fluctuations beyond $N = 10$. Thus, even though the best solution is found when $N = 25$, a near-optimal solution can be determined under a small number of subflows. This suggests that one can rapidly approximate the asymptotic solution of the multi-vehicle problem (dealing with individual vehicles routed so as to optimize a systemwide objective) based on a relatively small value of N.

Table 4.2: Numerical results for sample problem

N	1	2
obj	1.22e9	37.077
routes	$1 \to 4 \to 7$	$1 \to 4 \to 7$ $1 \to 2 \to 3 \to 7$
N	3	4
obj	31.7148	32.8662
routes	$(1 \to 4 \to 7)$ $1 \to 2 \to 3 \to 7$ $1 \to 5 \to 6 \to 7$	$(1 \to 4 \to 7) \times 2$ $1 \to 2 \to 3 \to 7$ $1 \to 5 \to 6 \to 7$
N	5	6
obj	32.1921	31.7148
routes	$(1 \to 4 \to 7) \times 2$ $(1 \to 2 \to 3 \to 7) \times 2$ $1 \to 5 \to 6 \to 7$	$(1 \to 4 \to 7) \times 2$ $(1 \to 2 \to 3 \to 7) \times 2$ $(1 \to 5 \to 6 \to 7) \times 2$
N	10	15
obj	31.5279	31.4851
routes	$(1 \to 4 \to 7) \times 4$ $(1 \to 2 \to 3 \to 7) \times 3$ $(1 \to 5 \to 6 \to 7) \times 3$	$(1 \to 4 \to 7) \times 5$ $(1 \to 2 \to 3 \to 7) \times 5$ $(1 \to 5 \to 6 \to 7) \times 4$ $(1 \to 4 \to 6 \to 7) \times 1$
N	25	30
obj	31.4513	31.4768
routes	$(1 \to 4 \to 7) \times 9$ $(1 \to 2 \to 3 \to 7) \times 8$ $(1 \to 5 \to 6 \to 7) \times 7$ $(1 \to 4 \to 6 \to 7) \times 1$	$(1 \to 4 \to 7) \times 11$ $(1 \to 2 \to 3 \to 7) \times 10$ $(1 \to 5 \to 6 \to 7) \times 8$ $(1 \to 4 \to 6 \to 7) \times 1$

Next, we obtain a solution to the same problem (4.40) using the alternative NLP formulation (4.36) where $0 \leq x_{ij}^k \leq 1$. Since in this example all subflows are identical, solving the NLP relaxed problem results in the same routing probabilities for all subflows, i.e., $x_{ij}^1 = ... = x_{ij}^N$. Therefore, we can further combine all x_{ij}^k over each link (i, j) and formulate the following N-subflow relaxed problem, referred to as N-NLP, giving the total normalized flow on each link, $x_{ij}, \forall (i, j) \in \mathcal{A}$ which is independent of

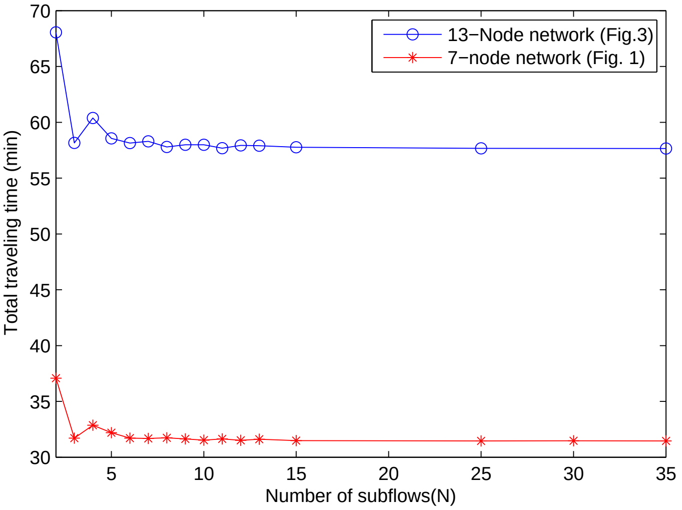


Figure 4·2: Performance as a function of N (No. of subflows)

N:

$$\begin{aligned}
\min_{x_{ij},\ i,j\in\mathcal{N}} \quad & \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{d_{ij}x_{ij}R}{v_f(1-(Rx_{ij}/f_{jam}^{ij})^p)^q}+egd_{ij}Rx_{ij}\right) && (4.41)\\
s.t. \quad & \sum_{j\in O(i)} x_{ij}-\sum_{j\in I(i)} x_{ji}=b_i, \quad \text{for each } i\in\mathcal{N}\\
& b_1=1,\ b_n=-1,\ b_i=0,\ \text{for } i\neq 1,n\\
& x_{ij}R\le f_{jam}^{ij}\\
& 0\le x_{ij}\le 1
\end{aligned}$$

This is a relatively easy to solve NLP problem. It can be readily shown that the objective function is convex. In particular, $\dfrac{d_{ij}x_{ij}R}{v_f(1-(x_{ij})^p)^q}$ is convex over $0\le x_{ij}\le 1$ and $egd_{ij}Rx_{ij}$ is a linear function, therefore, their positive weighted sum is a convex function and (4.41) is a convex optimization problem whose solution is a global optimum. Using the same parameter settings as before, we obtain the objective value of

31.45 mins and the optimal routes are:

35.88% of vehicle flow: $(1 \rightarrow 4 \rightarrow 7)$

31.74% of vehicle flow: $(1 \rightarrow 2 \rightarrow 3 \rightarrow 7)$

27.98% of vehicle flow: $(1 \rightarrow 5 \rightarrow 6 \rightarrow 7)$

4.44% of vehicle flow: $(1 \rightarrow 4 \rightarrow 6 \rightarrow 7)$

Compared to the best solution ($N = 25$) in Tab. 4.2 and Fig. 4·2, the difference in objective values between the integer and flow-based solutions is less than 0.1%. This supports the effectiveness of a solution based on a limited number of subflows in the MINLP problem.

Larger networks. We have also considered a more topologically complex network with 13 nodes and 20 links as shown in Fig. 4·3. The number on each link indicates the distance between adjacent nodes. We assume all other numerical values to be similar to the previous example. Fig. 4·2 shows the performance in terms of the objective function in (4.40) vs the number of subflows for this network. We can see that the optimal objective value converges around $N = 10$.

Now, let us solve the N-subflow relaxed problem (4.41) for this network with the same parameter settings as those in Section 4.3.4 to check for its accuracy. We obtain the optimal objective function value as 57.63 min which is almost equal to the optimal traveling time of 57.65 min obtained for $N = 35$ in the MINLP formulation. The optimal routing probabilities are as follows:

34.77% of vehicle flow: $(1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 13)$

27.52% of vehicle flow: $(1 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13)$

24.89% of vehicle flow: $(1 \rightarrow 6 \rightarrow 10 \rightarrow 7 \rightarrow 8 \rightarrow 13)$

10.81% of vehicle flow: $(1 \rightarrow 6 \rightarrow 3 \rightarrow 8 \rightarrow 13)$

1.71% of vehicle flow: $(1 \rightarrow 9 \rightarrow 10 \rightarrow 7 \rightarrow 8 \rightarrow 13)$

0.31% of vehicle flow: $(1 \rightarrow 6 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 13)$

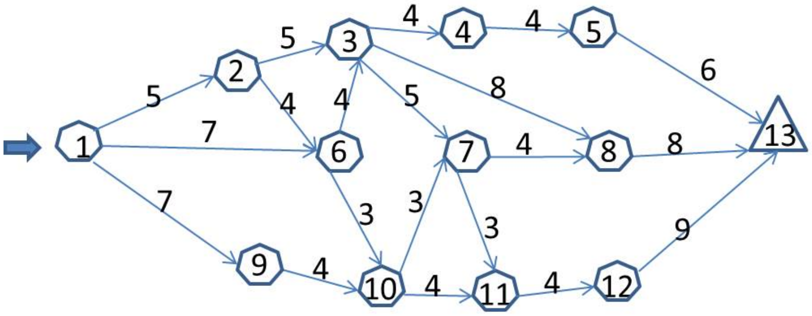


Figure 4·3: A 13-node network example for routing with recharging nodes.

CPU time Comparison. Based on our simulation results we conclude that the flow control formulation is a good approximation of the original MINLP problem. Tab. 4.3 compares the computational effort in terms of CPU time for both formulations to find optimal routes for the two sample networks we have considered. Our results show that the flow control formulation results in a reduction of about 5 orders of magnitude in CPU time with virtually identical objective function values (the difference between objective values of NLP and MINLP with near optimal N is less than 1%) .

Effect of recharging speed on optimal routes. Once we determine the optimal routes, we can also ascertain the total time spent traveling and recharging respectively, i.e., the first and second terms in (4.41). Obviously the value of eg, which captures the recharging speed, determines the proportion of traveling and recharging

Table 4.3: CPU time for sample problems

Fig. 4·1 Net.	MINLP	MINLP	NLP approx.
N	2	10(near opt)	-
obj	37.08	31.53	31.45
CPU time(sec)	312	9705	0.07
Fig. 4·3 Net.	MINLP	MINLP	NLP approx.
N	2	15(near opt)	-
obj	68.05	57.76	57.63
CPU time(sec)	820	10037	0.2

amount as well as the route selection. As shown in Tab. 4.4, the larger the product eg is, the slower the recharging speed, therefore the more weighted the recharging time in the objective function becomes. In this case, flows tend to select the shortest paths in terms of energy consumption. Conversely, if the recharging speed is fast, the routes are selected to prioritize the traveling time on paths.

Table 4.4: Numerical results for different values of eg for network of Fig. 4·1

eg	0.1	1	10
total time	18.94	31.45	154.48
time on paths	17.55	17.58	19.45
time at stations	1.39	13.87	135.03
optimal routes	31.53% : $(1 \to 2 \to 3 \to 7)$ 32.97% : $(1 \to 4 \to 7)$ 28.58% : $(1 \to 5 \to 6 \to 7)$ 5.78% : $(1 \to 4 \to 6 \to 7)$ 1.14% : $(1 \to 2 \to 4 \to 7)$	31.74% : $(1 \to 2 \to 3 \to 7)$ 35.88% : $(1 \to 4 \to 7)$ 27.98% : $(1 \to 5 \to 6 \to 7)$ 4.40% : $(1 \to 4 \to 6 \to 7)$	32.35% : $(1 \to 2 \to 3 \to 7)$ 49.63% : $(1 \to 4 \to 7)$ 18.02% : $(1 \to 5 \to 6 \to 7)$

Price of Anarchy. In order to compare system performance under a user-optimal (single-vehicle routing problem) policy and a system-optimal (multiple-vehicle routing problem) policy, we investigate the Price of Anarchy (PoA) for this problem. To make this comparison, we consider two different scenarios:

1. **A single driver acts selfishly.** We control all vehicles to follow system-optimal paths and assume that a single driver acts selfishly. We then investigate this driver's total traveling time and the possible gain resulting from this deviation.

Let us consider the numerical example in Section 4.3.4 for the network shown in

Fig. 4·1. The system-optimal flows are obtained by solving the NLP problem (4.41). Under these flows, let us calculate the traveling time, τ_{ij}^c, of all links $(i,j) \in \mathcal{A}$ with positive flows using (4.38) as shown in Tab. 4.5. Assuming the energy consumption on each link is equal to the distance of that link, the total traveling time experienced by an individual EV, T_{EV}, depends on the system-optimal path assigned to its subflow, P:

Table 4.5: Traveling time on each link for the network shown in Fig. 4·1 under system-optimal flows

τ_{12}^c	τ_{14}^c	τ_{15}^c	τ_{23}^c	τ_{24}^c	τ_{46}^c	τ_{56}^c	τ_{37}^c	τ_{47}^c	τ_{67}^c
6.18	8.83	8.24	4.33	5	3.61	5.06	7.42	7.90	4.99

$$T_{EV} = \sum_{(i,j)\in P} (\tau_{ij}^c + e_{ij} g)$$

Total Traveling time for a single EV in flow $(1 \to 4 \to 7)$: 28.94 min

Total Traveling time for a single EV in flow $(1 \to 2 \to 3 \to 7)$: 32.43 min

Total Traveling time for a single EV in flow $(1 \to 5 \to 6 \to 7)$: 33.59 min

Total Traveling time for a single EV in flow $(1 \to 4 \to 6 \to 7)$: 31.24 min

Now if we solve the single-vehicle routing problem for a lone EV in the network, the user-optimal path is $(1 \to 4 \to 7)$ with a traveling time of 28.94 min. Thus, in the system-optimal problem, vehicles assigned to the subflows following this path experience the same traveling time as if they act selfishly and follow the user-optimal path. However, vehicles assigned to other subflows will experience longer traveling times in order to reduce the total elapsed time for the whole inflow. For instance, a single EV can gain 13.86% in its traveling time by acting selfishly and deviating from the subflow assigned to path $(1 \to 5 \to 6 \to 7)$ and joining path $(1 \to 4 \to 7)$.

2. **All drivers act selfishly.** In this case, the flow will be in a Nash equilibrium

where no single user can incur a gain by changing its own strategy (Youn and Jeong, 2008). Based on Wardrop's principle, the equilibrium occurs at flows that minimize the potential function (Roughgarden, 2005)

$$\phi(f) = \sum_{(i,j)\in\mathcal{A}} \int_{0}^{f_{ij}} \eta_{ij}(x)dx \tag{4.42}$$

where $\eta_{ij}(x)$ is the travel time incurred by traffic that traverses link (i, j) as a function of the link congestion (Rx_{ij} in (4.41)). The PoA is defined as the ratio of the total system cost (the total elapsed time) under Nash equilibrium to the total cost under the social-optimal flows. Tab. 4.6 shows the normalized Nash-equilibrium flow, x^e_{ij}, on each link (i, j) of the network shown in Fig. 4·1 resulting in the following selfish routing:

46.4% of vehicle flow: $(1 \rightarrow 4 \rightarrow 7)$
30.7% of vehicle flow: $(1 \rightarrow 2 \rightarrow 3 \rightarrow 7)$
21.4% of vehicle flow: $(1 \rightarrow 5 \rightarrow 6 \rightarrow 7)$
1.5% of vehicle flow: $(1 \rightarrow 4 \rightarrow 6 \rightarrow 7)$

Applying Nash-equilibrium flows into the system-wide objective function (4.41), the total traveling time is 32.27 min which is higher than the optimal cost of 31.45 obtained under the social-optimal policy and the price of anarchy is $PoA = 1.038$.

Table 4.6: Normalized Nash-equilibrium flows

x^e_{12}	x^e_{14}	x^e_{15}	x^e_{23}	x^e_{24}	x^e_{46}	x^e_{56}	x^e_{37}	x^e_{47}	x^e_{67}
30.7	47.9	21.4	30.7	0	1.5	21.4	30.7	46.4	22.9

4.4 Selection of the Number of Subflows

We begin with the observation that the objective function as well as the constraints of the flow control formulation (NLP) (4.36) are the same as those of the MINLP formulation (4.27), except for the relaxed binary constraints, i.e., $0 \leq x_{ij}^k \leq 1$. Thus, in general, the optimal objective value of the NLP problem will be equal or lower than that of the MINLP problem. We seek the best value of N to render the problem computationally manageable.

Similar to the numerical examples in Section 4.3.4, we focus on the case where we divide the total vehicle inflow, R, into N sublows each with a rate of R/N. In this case, solving the NLP problem results in the same routing probabilities for all subflows, i.e., $x_{ij}^1 = ... = x_{ij}^N$. Therefore, we can combine them and reformulate the problem as an N-subflow relaxed problem, referred to as "N-NLP", giving the total normalized flow on each link, x_{ij}, $(i, j) \in \mathcal{A}$ (see (4.41)).

In Section 4.3.4, the numerical results show that the optimal objective value quickly converges for a small value of N. Thus, even though the best solution may be found for a larger N, a near-optimal solution can be determined under a small number of subflows. This suggests that we can approximate the asymptotic solution of the multi-vehicle problem based on a relatively small value of N. Our goal is to find a lower bound, N^*, for the number of subflows, such that by selecting any $N \geq N^*$, we can guarantee that the N-NLP solution will be in a given neighborhood of the MINLP solution. To do so, first we proceed as follows.

Let π be the number of different paths from the origin node to the destination node in a given graph, and let x_p denote the normalized amount of flow through path $p,\ p = 1, ..., \pi$ determined using the solution of the N-NLP problem (4.36). Based on that, we define *active paths* to be those with non-zero flow, i.e., paths with $x_p > 0$. Let us assume there are $q \leq \pi$ active paths; then, let $\tilde{x}_p$ denote the normalized flow on active path p, $p = 1, ..., q$.

Next, we define $N_p(N)$ to be the number of subflows assigned to active path p obtained by the MINLP solution with N subflows. We write $N_p(N)$ to emphasize that the MINLP solution depends on the choice of N. Then, $\frac{N_p(N)}{N}$ is the normalized flow on path p obtained by solving the MINLP problem with N subflows. Noting that N is an integer, the best N is the one that minimizes the deviation from the normalized flows obtained by solving the NLP problem (lower bounds to the MINLP solutions), i.e.,

$$\min_{N\geq 1} |\frac{N_p(N)}{N} - x_p|, \ p = 1, ..., \pi$$

Since the computational complexity of the MINLP problem increases with N, the selection of N is a trade-off between a near-optimal solution and the computational effort required to solve the problem. To address this trade-off, let us define the *average deviation* between the optimal routing probabilities of the active paths obtained by solving the N-NLP problem, $\tilde{x}_p$, $p = 1, ..., q$, and the normalized flows obtained by solving the MINLP problem, $N_p(N)/N$, as a near-optimality metric, i.e., $\frac{1}{q}\sum_{p=1}^{q} |\frac{N_p(N)}{N} - \tilde{x}_p|$. Then, we define a "desired accuracy", δ, as the upper bound for this metric and seek to determine values of N that satisfy:

$$\frac{1}{q}\sum_{p=1}^{q} |\frac{N_p(N)}{N} - \tilde{x}_p| \leq \delta \tag{4.43}$$

Based on (4.43), we seek the critical N^* such that by selecting $N \geq N^*$ the average deviation between the N-NLP and MINLP solutions does not exceed δ, i.e.,

$$N \geq \frac{\sum_{p=1}^{q} |N_p(N) - N\tilde{x}_p|}{q\delta} \tag{4.44}$$

We define:

$$N^* = \lceil \frac{\max_{N\geq 1}(\sum_{p=1}^{q} |N_p(N) - N\tilde{x}_p|)}{q\delta} \rceil \tag{4.45}$$

Since the numerator of N^* is an upper bound for the numerator in the right hand side of (4.44) and noting that δ and q are constants, choosing $N \geq N^*$ guarantees that the average deviation between the NLP and MINLP solutions never exceeds our desired accuracy, δ. However, since $N_p(N)$ is a function of N, finding a closed-form expression for $\max_{N\geq 1}(\sum_{p=1}^{q} |N_p(N) - N\tilde{x}_p|)$ in the numerator of (4.45) is not easy. To address this issue, we propose a method which efficiently and accurately estimates the MINLP solution. Then, using these estimates, to be referred as $\hat{N}_p(N),\ p = 1, .., \pi$, for a large range of the number of subflows, N, we can find $\max_{N\geq 1}(\sum_{p=1}^{q} |\hat{N}_p(N) - N\tilde{x}_p|$ and select the proper N^* using (4.45).

```
Algorithm 1 MINLP Solution Estimation Algorithm
Input: $N$
Output: estimation for MINLP solution, $\hat{N}_p$, $p = 1, .., \pi$
Initialization: Set $\hat{N}_p = 0,\ p = 1, .., \pi$.
1: Solve $N$-NLP problem and identify active paths and corresponding $\tilde{x}_p$, $p = 1, .., q$
2: Form the set of all possible combinations for assigning $N$ subflows to $q$ active
   paths, $\mathcal{S}_N$, for each $i \in \mathcal{S}_N$, $N_p^i$ is the number of subflows allocated to path $p$ for
   the $i$th such assignment.
3: Find the best assignment, $i^*$, so that
   $i^* = \arg\min_{i\in\mathcal{S}_N}(\sum_{p=1}^{q} |N_p^i/N - \tilde{x}_p|)/q$
4: Set $\hat{N}_p = N_p^{i^*}$ for active paths.
End
```

As described in Algorithm 1, first, we solve the N-NLP problem and find the optimal objective value and optimal normalized flow on each link (i, j), x_{ij}. Next, we determine the active paths and their corresponding optimal normalized flow, $\tilde{x}_p,\ p = 1, ..., q$. Since the objective functions of the MINLP and NLP problems are the same, the MINLP solution for each N, assigns the subflows to the active paths such that the deviation between the corresponding normalized flows, $N_p(N)/N$, and the N-NLP solution, $\tilde{x}_p,\ p = 1, .., q$, is minimized. Therefore, to estimate the MINLP solution, $\hat{N}_p$, for each value of N, we consider all possible combinations of assigning N subflows to the q active paths and form a set $\mathcal{S}_N$. There are $\begin{pmatrix} N+q-1 \\ N \end{pmatrix}$ different such assignments (cardinality of set $\mathcal{S}_N$). Let N_p^i denote the number of subflows allocated to path p, $p = 1, ..., q$ in the ith such assignment, $i \in \mathcal{S}_N$. For each assignment i, the equivalent normalized flows on active paths become $[N_1^i/N, ..., N_q^i/N]$ where $\sum_{p=1}^q N_p^i = N$. For each value of N, i^* is the "best assignment" if it results in the minimum average deviation from the N-NLP solution among all $i \in \mathcal{S}_N$. i.e.,

$$i^* = \arg\min_{i \in \mathcal{S}_N} \frac{1}{q} \sum_{p=1}^{q} |\frac{N_p^i}{N} - \tilde{x}_p| \tag{4.46}$$

For each value of N, we set the i^* assignment as the estimate of the optimal routing of subflows (MINLP solution), $\hat{N}_p = N_p^{i^*}$, $p = 1, .., q$ and $\hat{N}_p = 0$ for non-active paths. Finally, for a given graph one can create a lookup table of the estimates of the MINLP solution for a range of N and find $\max_{N \geq 1}(\sum_{p=1}^q |\hat{N}_p(N) - N\tilde{x}_p|)$ to calculate N^* for a desired δ using (4.45).

Remark 1 - A simple way of intuitively determining the critical N is as follows: defining $\hat{x} = \min_p \tilde{x}_p$, $\hat{x}$ is the least fraction of inflow obtained by the N-NLP problem to flow through an individual path. If we choose $N = \lceil \frac{1}{\hat{x}} \rceil$, the MINLP solution will have a chance to send at least one subflow through the same path and the normalized flows obtained by the MINLP will be close to the NLP solution. Obviously, it may not be the best N and there is no guarantee for such N to satisfy the bound defined in (4.43). However, our simulation results show that it is a *good* "rule of thumb" for selecting N.

4.4.1 Numerical Example

Consider the 7-node graph shown in Fig. 4·1 with the same parameter values as in Section 4.3.4. The N-NLP solution is shown in Tab. 4.7.

Using the optimal x_{ij}s we determine each active path and corresponding $\tilde{x}_p$:

$$\tilde{x}_1 = 31.73\% \quad \text{Path}_1 : (1 \to 2 \to 3 \to 7)$$

Table 4.7: Optimal Normalized flow on each link (x_{ij}) obtained by solving N-NLP problem

x_{12}	x_{14}	x_{15}	x_{23}	x_{24}
31.73%	40.28%	27.98%	31.73 %	0
x_{46}	x_{56}	x_{37}	x_{47}	x_{67}
4.40%	27.98%	31.73%	35.88%	32.39%

$$\begin{aligned} \tilde{x}_2 &= 4.40\% \quad \text{Path}_2 : (1 \rightarrow 4 \rightarrow 6 \rightarrow 7) \\ \tilde{x}_3 &= 35.88\% \quad \text{Path}_3 : (1 \rightarrow 4 \rightarrow 7) \\ \tilde{x}_4 &= 27.98\% \quad \text{Path}_4 : (1 \rightarrow 5 \rightarrow 6 \rightarrow 7) \end{aligned}$$

In this example, there are 4 active paths. Fig. 4·4 shows the average deviation between the NLP solution and the normalized flows obtained by the estimated MINLP solutions for $N = 1, ..., 72$. Tab. 4.8 shows the best assignments in the form $[\hat{N}_1\ \hat{N}_2\ \hat{N}_3\ \hat{N}_4]$ (corresponds to [Path_1 Path_2 Path_3 Path_4]) for different values of N. In Fig. 4·4, we observe that the minimum average deviation occurs for $N = 25$ with the closest objective value to the NLP problem and the following normalized flows:

$$\begin{aligned} N_1/N &= 8/25 = 32\% \quad \text{Path}_1 : (1 \rightarrow 2 \rightarrow 3 \rightarrow 7) \\ N_2/N &= 1/25 = 4\% \quad \text{Path}_2 : (1 \rightarrow 4 \rightarrow 6 \rightarrow 7) \\ N_3/N &= 9/25 = 36\% \quad \text{Path}_3 : (1 \rightarrow 4 \rightarrow 7) \\ N_4/N &= 7/25 = 28\% \quad \text{Path}_4 : (1 \rightarrow 5 \rightarrow 6 \rightarrow 7) \end{aligned}$$

which are almost identical to the NLP solution, $\tilde{x}_p,\ p = 1, ..., 4$. In this particular example $\hat{x} = \min_{p=1,..,4} \tilde{x}_p = 0.044$ which suggests the same number of subflows based on the simple "rule of thumb" in Remark 1.

Finally, we investigate the correctness of the bound defined in (4.44) for different

Table 4.8: Estimates for the MINLP solution for different values of N

N	Estimated $\hat{N}_1, ..., \hat{N}_4$	N	Estimated $\hat{N}_1, ..., \hat{N}_4$	N	Estimated $\hat{N}_1, ..., \hat{N}_4$
1	0 0 1 0	25	8 1 9 7	49	15 2 18 14
2	1 0 1 0	26	8 1 10 7	50	16 2 18 14
3	1 0 1 1	27	9 1 10 7	51	16 2 19 14
4	1 0 2 1	28	9 1 10 8	52	16 2 19 15
5	2 0 2 1	29	9 1 11 8	53	17 2 19 15
6	2 0 2 2	30	10 1 11 8	54	17 3 19 15
7	2 0 3 2	31	10 1 11 9	55	18 2 20 15
8	3 0 3 2	32	10 1 12 9	56	18 2 20 16
9	3 0 3 3	33	11 1 12 9	57	18 3 20 16
10	3 0 4 3	34	11 1 12 10	58	18 3 21 16
11	4 0 4 3	35	11 1 13 10	59	19 3 21 16
12	4 1 4 3	36	11 2 13 10	60	19 3 21 17
13	4 0 5 4	37	12 2 13 10	61	19 3 22 17
14	4 1 5 4	38	12 2 13 11	62	20 3 22 17
15	5 1 5 4	39	12 2 14 11	63	20 3 22 18
16	5 1 6 4	40	13 2 14 11	64	20 3 23 18
17	5 1 6 5	41	13 2 15 11	65	21 3 23 18
18	6 1 6 5	42	13 2 15 12	66	21 3 24 18
19	6 1 7 5	43	14 2 15 12	67	21 3 24 19
20	6 1 7 6	44	14 2 16 12	68	22 3 24 19
21	7 1 7 6	45	14 2 16 13	69	22 3 25 19
22	7 1 8 6	46	15 2 16 13	70	22 3 25 20
23	7 1 8 7	47	15 2 17 13	71	23 3 25 20
24	8 1 8 7	48	15 2 17 14	72	23 3 26 20

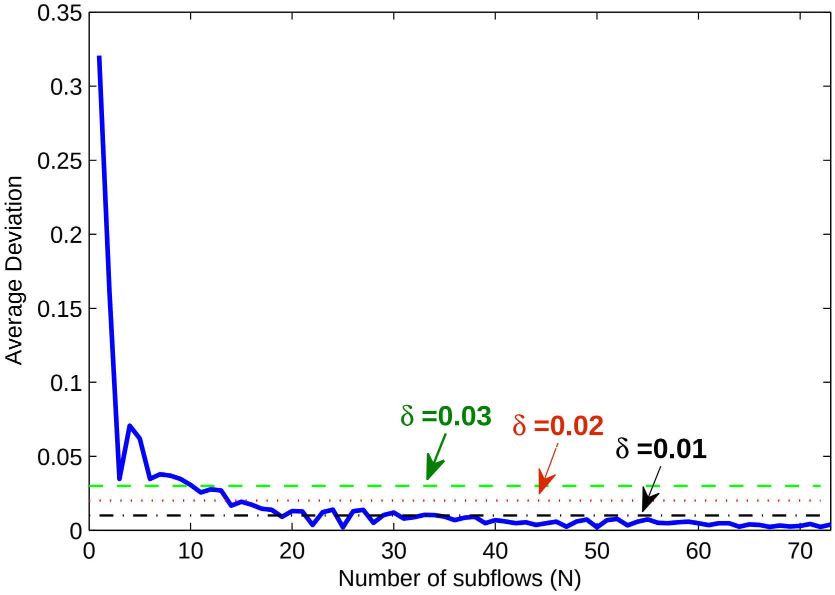


Figure 4·4: Average deviation between the solution of the NLP and estimated solution of the MINLP problem for different values of N

values of δ. It can be seen in Fig. 4·4 that by selecting $N \geq N^*$, calculated for different values of δ in Tab. 4.9, the average deviation never exceeds our desired accuracy, δ, which shows the validity of the proposed criterion in (4.45). Increasing the upper bound of the average deviation allows us to select smaller N and consequently the problem size and associated computational complexity decreases. This demonstrates the trade-off between proximity to optimality and computational effort required to solve the problem.

Table 4.9: Critical number of subflows, N^*, for different values of δ

$\delta = 0.01$	$N^* = 33$
$\delta = 0.02$	$N^* = 17$
$\delta = 0.03$	$N^* = 11$

Our numerical results show that the optimal routing obtained by solving the MINLP problem is exactly the same as our estimate, i.e., the best assignment corresponding to the minimum average deviation with the NLP solution for each N. This can be verified by comparing the results in Tab. 4.2 and Tab. 4.8 for different values of N.

4.5 Multiple-Vehicle Routing Problem in the Presence of Non-Electric Vehicle Flows

In this section, we extend our approach by involving both EV and Non-EV (NEV) flows. Let all vehicles enter the network at node 1 and let R denote the rate of vehicles arriving at this node. Assuming a fraction P of NEVs in the inflow, NEVs and EVs enter the network with flow rates given by RP and $R(1-P)$ respectively. We propose two different ways to incorporate the effect of NEV flows. In the first method, we assume the flow of NEVs on each link (i,j), f_{ij}^{NEV}, is known (e.g., Nash equilibrium flows or socially-optimal flows are determined) and we can calculate the *residual* capacity for each link accordingly, i.e., $f_{jam}^{ij} - f_{ij}^{NEV}$. Thus, the problem is reduced to the multi-vehicle routing problem with all arriving vehicles as EVs with the residual capacity for links which has already been discussed.

Our second method is to reformulate an optimization problem in order to control both EV and NEV flows. Similar to our approach in Section 4.3, we group EVs as well as NEVs into subflows. In particular we divide the inflow of NEVs into a fixed number of subflows, M (e.g., the number of distinct paths from the origin to the destination node) and the inflow of EVs into N subflows. Our objective is to determine optimal routes for NEV subflows and optimal routes, as well as energy recharging amounts, for each EV subflow so as to minimize the total elapsed time of these subflows from origin to destination. Note that for NEVs, we do not consider the refueling process as part of this optimization problem. The decision variables consist of (i) $x_{ij}^{k} \in \{0,1\},\ k = 1,..,M$ and $y_{ij}^{l} \in \{0,1\},\ l = 1,..,N$, corresponding to the selection of link (i,j) by NEV and EV subflows respectively, and (ii) charging amounts r_i^l for EV subflows for all nodes $i = 1, \dots, n-1$ and subflows $l = 1, \dots, N$. Given traffic congestion effects, the time and energy consumption on each link depends on the values of x_{ij}^k, y_{ij}^l and the fraction of the total flow rate R associated with the kth NEV subflow or the lth EV subflow. As in Section 4.3, the simplest such flow allocation is to assign each subflow the same rate, i.e., every NEV subflow $k = 1, .., M$ is assigned a rate RP/M and every EV subflow $l = 1, \dots, N$ is assigned a rate $R(1-P)/N$. Let $\mathbf{x_{ij}} = (x_{ij}^1, \cdots, x_{ij}^M, y_{ij}^1, \cdots, y_{ij}^N)^T$ and $\mathbf{r_i} = (r_i^1, \cdots, r_i^N)^T$ where r_i^l is the amount of charge selected by the lth EV subflow at node i. Similar to Section 4.3, we denote the traveling time (delay) a vehicle will experience through link (i,j) by some nonlinear function $\tau_{ij}(\mathbf{x_{ij}})$. The corresponding energy consumption for the lth subflow of EVs through (i,j) is a nonlinear function denoted by $e_{ij}^l(\mathbf{x_{ij}})$. Finally, E_i^l represents the residual energy of subflow l of EVs at node i, given by the aggregated residual energy of all EVs in the subflow. The optimization problem is formulated as follows:

$$\min_{\mathbf{x_{ij}}, \mathbf{r_i},\, i,j \in \mathcal{N}} \quad [\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{M} \tau_{ij}(\mathbf{x_{ij}}) x_{ij}^k \frac{RP}{M} + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{N} \left(\tau_{ij}(\mathbf{x_{ij}}) y_{ij}^l \frac{R(1-P)}{N} + r_i^l g y_{ij}^l \right)] \tag{4.47}$$

$$s.t. \text{ for each } k \in \{1, \dots, M\} :$$

$$\sum_{j \in O(i)} x_{ij}^k - \sum_{j \in I(i)} x_{ji}^k = b_i, \quad \text{for each } i \in \mathcal{N} \tag{4.48}$$

$$b_1 = 1,\, b_n = -1,\, b_i = 0, \text{ for } i \neq 1, n \tag{4.49}$$

$$\text{for each } l \in \{1, \dots, N\} :$$

$$\sum_{j \in O(i)} y_{ij}^l - \sum_{j \in I(i)} y_{ji}^l = b_i, \quad \text{for each } i \in \mathcal{N} \tag{4.50}$$

$$b_1 = 1,\, b_n = -1,\, b_i = 0, \text{ for } i \neq 1, n \tag{4.51}$$

$$E_j^l = \sum_{i \in I(j)} (E_i^l + r_i^l - e_{ij}^l(\mathbf{x_{ij}}))y_{ij}^l, \quad j = 2, \ldots, n \tag{4.52}$$

$$E_1^l \text{ is given}, \quad E_i^l \geq 0, \quad \text{for each } i \in \mathcal{N} \tag{4.53}$$

$$x_{ij}^k \in \{0, 1\}, \quad y_{ij}^l \in \{0, 1\}, \quad r_i^l \geq 0 \tag{4.54}$$

In the above formulation, (4.47) is the objective function which for NEVs is the first sum representing the overall traveling time from origin to destination by adding the link traveling times $\tau_{ij}(\mathbf{x_{ij}})$ when $x_{ij}^k = 1$. For EVs, the second sum includes the charging times $r_i^l g$ when $y_{ij}^l = 1$ and an EV subflow selects node l for charging. The constraints (4.48)-(4.49) and (4.50)-(4.51) represent flow conservation for NEV and EV subflows respectively, while (4.52)-(4.53) shows the energy dynamics for each EV subflow. This is a Mixed Integer Non-Linear Programming Problem (MINLP) with $(M + N)m + 2(n - 1)N$ variables. Similar to our discussion in Section 4.3.1, one can exploit some properties of the optimal solution and energy dynamics in order to decompose this problem into route selection and recharging amount determination and reduce the problem dimensionality. We omit numerical results, which lead to observations similar to those presented in Section 4.3.4, including a behavior of performance as a function of the number of subflows similar to that of Fig. 4·2.

4.6 Routing of energy-aware vehicles in networks with inhomogeneous charging nodes

In this section, we relax the homogeneity assumption for charging stations and investigate the routing problem for BPVs through a network of "inhomogeneous" charging nodes, i.e., charging rates at different nodes are not identical. In fact, depending on an outlet's voltage and current, charging an EV battery could take anywhere from minutes to hours and the Society of Automotive Engineering (SAE) classifies charging stations into three categories (Joos et al., 2010), (Bai et al., 2010), (J17, 2012) as shown in Tab. 4.10. Thus, charging rates and times are highly dependent on the class of the charging station and they clearly affect the solution of our optimization problem.

Table 4.10: Classification of charging stations (Bai et al., 2010)

Charge Method	Nominal Supply Voltage(volts)	Max. Current (Amps)	Miles per every hour charging
AC Level 1	120 VAC, 1-phase	12 A	< 5
AC Level 2	208-240 VAC, 1-phase	32 A	up to 62
DC Charging	300 - 460VDC	400 A Max.	up to 300

In the sequel, we study both user-centric and system-centric routing problems in a network with inhomogeneous charging nodes. Using similar frameworks as those provided for a network with homogeneous charging nodes, we generalize our formulations such that they recover the homogeneous case as well.

4.6.1 Single Vehicle Routing

The network model is similar to the one introduced in Section 4.2 in which each node $i \in \mathcal{N}/\{n\}$ represents a charging station with g_i denoting the charging time per unit of energy at node i. For the network with homogeneous charging nodes, $g_i = g_j \; \forall i, j \in \mathcal{A}$. In contrast, here g_i, $i = 1, \ldots, n$, are node-dependent parameters and not identical. Without loss of generality, we assume all nodes have a charging capability (if node i does not have such capability, we can simply set $g_i = \infty$).

Considering τ_{ij} and e_{ij} as fixed parameters as explained in Section 4.2, we formulate the user-centric problem as follows:

$$\min_{x_{ij}, r_i,\; i,j \in \mathcal{N}} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \tau_{ij} x_{ij} + \sum_{i=1}^{n} \sum_{j=1}^{n} r_i g_i x_{ij} \tag{4.55}$$

$$s.t. \quad \sum_{j \in O(i)} x_{ij} - \sum_{j \in I(i)} x_{ji} = b_i, \quad \text{for each } i \in \mathcal{N} \tag{4.56}$$

$$b_1 = 1,\; b_n = -1,\; b_i = 0, \;\text{ for } i \neq 1, n \tag{4.57}$$

$$E_j = \sum_{i \in I(j)} (E_i + r_i - e_{ij}) x_{ij}, \text{ for } j = 2, \ldots, n \tag{4.58}$$

$$0 \leq E_i \leq B, \quad E_1 \text{ given, for each } i \in \mathcal{N} \tag{4.59}$$

$$x_{ij} \in \{0, 1\}, \quad r_i \geq 0 \tag{4.60}$$

This is a Mixed Integer Non-Linear Programming (MINLP) problem with $m+2(n-1)$ variables and it will be referred to as **P1**. A crucial difference between **P1** and MINLP (4.2)-(4.7), for the homogeneous case, is that here the charging rates g_i in (4.55) are node-dependent.

Properties

Similar to our approach in Section 4.2.1, we reduce the computational complexity of **P1** by deriving some key properties of an optimal solution. Applying these properties we obtain a lower-dimensional problem with $m + (n-1)$ variables.

Lemma 6: Given (4.55)-(4.60), an optimal solution $\{x_{ij}, r_i, E_i\}$, $i, j \in \mathcal{N}$ satisfies:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} (r_i x_{ij} - e_{ij} x_{ij}) g_i = \sum_{i=1}^{n} \sum_{j=1}^{n} (E_j - E_i) g_i x_{ij} \tag{4.61}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} E_j (g_i - g_j) x_{ij} - E_1 g_1 \tag{4.62}$$

Proof: Multiplying both sides of (4.1) by g_i gives:

$$E_j g_i = \begin{cases} (E_i + r_i - e_{ij}) g_i & \text{if } x_{ij} = 1, \\ 0 & \text{otherwise .} \end{cases}$$

which can be expressed as

$$\sum_{i \in I(j)} E_j g_i x_{ij} = \sum_{i \in I(j)} (E_i + r_i - e_{ij}) g_i x_{ij}$$

Summing both sides over $j = 2, \ldots, n$ and rearranging yields:

$$\begin{aligned}
&\sum_{j=2}^{n} \sum_{i \in I(j)} E_j g_i x_{ij} - \sum_{j=2}^{n} \sum_{i \in I(j)} E_i g_i x_{ij} \\
&= \sum_{j=2}^{n} \sum_{i \in I(j)} (r_i - e_{ij}) g_i x_{ij}
\end{aligned}$$

Based on (4.1), $E_i = 0$ for all nodes which are not in the selected path. Thus we can rewrite the equation above as

$$\sum_{i=1}^{n} \sum_{j=1}^{n} (r_i x_{ij} - e_{ij} x_{ij}) g_i = \sum_{i=1}^{n} \sum_{j=1}^{n} (E_j - E_i) g_i x_{ij}$$

which establishes (4.61). Finally, (4.62) follows by observing that if P is an optimal path we can re-index nodes so that $P = \{1, ..., n\}$ with $g_n = 0$. Thus, we have $\sum_{i=1}^{n} \sum_{j=1}^{n} E_i g_i x_{ij} = E_1 g_1 + \ldots + E_{n-1} g_{n-1}$ which can also be written as $E_1 g_1 + \sum_{i=2}^{n} \sum_{j=2}^{n} E_j g_j x_{ij}$ where $x_{ij} = 0$ for all (i, j) not in the optimal path. Therefore,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} (E_j - E_i) g_i x_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{n} E_j (g_i - g_j) x_{ij} - E_1 g_1$$

which proves (4.62).■

Lemma 7: If $\sum_i r_i^* > 0$ in the optimal routing policy, then $E_n^* = 0$.

Proof: This is the same as the homogeneous charging node case; see Lemma 2 in Section 4.2.1.

Using Lemma 6, we replace $\sum_{i=1}^{n} \sum_{j=1}^{n} r_i g_i x_{ij}$ in (4.55) and eliminate the presence of r_i, $i = 2, \ldots, n-1$, from the objective function and the constraints. Thus, **P1** is reduced to the following MINLP problem referred to as **P2**:

$$\min_{\substack{x_{ij}, E_i \\ i,j \in \mathcal{N}}} \sum_{i=1}^{n} \sum_{j=1}^{n} \big(\tau_{ij} x_{ij} + e_{ij} g_i x_{ij} + E_j (g_i - g_j) x_{ij} \big) - E_1 g_1 \tag{4.63}$$

$$s.t \quad \sum_{j \in O(i)} x_{ij} - \sum_{j \in I(i)} x_{ji} = b_i \tag{4.64}$$

$$b_1 = 1,\ b_n = -1,\ b_i = 0 \quad \text{for} \quad i \neq 1, n \tag{4.65}$$

$$0 \leqslant E_j - (E_i - e_{ij})x_{ij} \leqslant B \quad \forall\ i, j \in N \tag{4.66}$$

$$0 \leqslant E_i \leqslant B,\ E_1 \text{ given},\ \forall\ i \in N \tag{4.67}$$

$$x_{ij} \in \{0, 1\} \tag{4.68}$$

Constraint (4.66) is derived from (4.58). Assuming $x_{ij} = 1$, i.e., arc (i, j) is part of the optimal path, we can recover $r_i = E_j - E_i + e_{ij}$ and constraint (4.66) is added to prevent any vehicle from exceeding its capacity B in an optimal path. Solving this problem gives both an optimal path and residual battery energy at each node.

Although **P2** has fewer decision variables, it is still a MINLP which is hard to solve for large networks. Specifically, the computation time is highly dependent on the number of nodes and arcs in the network. In what follows we introduce a *locally optimal* charging policy, leading to a simpler problem, by arguing as follows. Looking at (4.63), the term $\sum_{i=1}^{n} \sum_{j=1}^{n} E_j(g_i - g_j)x_{ij}$ is minimized by selecting each E_j depending on the sign of $(g_i - g_j)$:

Case 1: $g_i - g_j < 0$, i.e., node i has a faster charging rate than node j. Therefore, E_j should get its maximum possible value, which is $B - e_{ij}$. This implies that the vehicle must be maximally charged at node i.

Case 2: $g_i - g_j \geqslant 0$, i.e., node j has a faster or same charging rate as node i. In this case, E_j should get its minimum value $E_j = 0$. This implies that the vehicle should get the minimum charge needed at node i in order to reach node j.

We define $\pi_{\mathbf{C}}$ to be the charging policy specified as above and note that it does not guarantee the global optimality of E_i thus selected in (4.63) which can easily be checked by a counterexample. However, it allows us to decompose the optimal routing problem from the optimal charging problem. If, in addition, we consider only solutions for which the vehicle is recharged at least once (otherwise, the vehicle is not energy-constrained and the problem is of limited interest), we can obtain the following result.

Theorem 2: If $\sum_i r_i^* > 0$ (i.e., the vehicle has to be recharged at least once), then under charging policy $\pi_{\mathbf{C}}$, the solution x_{ij}^*, $i, j \in \mathcal{N}$, of the original problem (4.55) can be determined by solving the LP problem:

$$\min_{x_{ij}, i,j \in \mathcal{N}} \sum_{i=1}^{n} \sum_{j=1}^{n} \big(\tau_{ij} + e_{ij} g_i + K(g_i - g_j)\big) x_{ij} \tag{4.69}$$

$$K = \begin{cases} B - e_{ij} & \text{if } g_i < g_j, \\ 0 & \text{otherwise .} \end{cases} \tag{4.70}$$

$$s.t. \quad \sum_{j \in O(i)} x_{ij} - \sum_{j \in I(i)} x_{ji} = b_i \tag{4.71}$$

$$b_1 = 1, b_n = -1, b_i = 0 \quad for i \neq 1, n \tag{4.72}$$

$$0 \leqslant x_{ij} \leqslant 1 \tag{4.73}$$

Proof: Applying charging policy $\pi_{\mathbf{C}}$ in (4.63) we change the objective function to $\sum_{i=1}^{n} \sum_{j=1}^{n} \big(\tau_{ij} + e_{ij} g_i + K(g_i - g_j)\big) x_{ij} - E_1 g_1$ where K is as in (4.70). Therefore, x_{ij}^* can be determined by the following problem:

$$\min_{x_{ij}, i,j \in \mathcal{N}} \sum_{i=1}^{n} \sum_{j=1}^{n} \big(\tau_{ij} + e_{ij} g_i + K(g_i - g_j)\big) x_{ij} - E_1 g_1$$

$$K = \begin{cases} B - e_{ij} & \text{if } g_i < g_j, \\ 0 & \text{otherwise .} \end{cases}$$

$$s.t. \quad \sum_{j \in O(i)} x_{ij} - \sum_{j \in I(i)} x_{ji} = b_i, \quad \text{for each } i \in \mathcal{N}$$

$$b_1 = 1,\ b_n = -1,\ b_i = 0,\ \text{for } i \neq 1, n$$

$$x_{ij} \in \{0, 1\}$$

which is a typical shortest path problem formulation. Then, similar to our argument in proof of Theorem 1, the above integer programming problem is equivalent to the LP with the integer restriction on x_{ij} relaxed. Finally, since E_1 and g_1 are given, the problem reduces to (4.69), which proves the theorem. ■

Note that If $g_i = g_j$ for all i, j in (4.69), the problem reduces to the homogeneous charging node case with the same optimal LP formulation as in Theorem 1. With $g_i \neq g_j$ however, the LP formulation cannot guarantee global optimality, although the routes obtained through Theorem 2 may indeed be optimal, in which case the optimal charging amounts are obtained as described next.

Determination of optimal recharging amounts r_i^*

Once we determine an optimal route P, it is relatively easy to find a feasible solution for r_i, $i \in P$, to satisfy the constraint (4.5) and minimize the total charging time on the selected path. It is obvious that the optimal charging amounts r_i^* are non-unique in general. Without loss of generality we re-index nodes so that we may write $P = \{1, ..., n\}$. Then, the problem resulting in an optimal charging policy is

$$\min_{r_i,\ i\in P} \quad \sum_{i\in P} g_i r_i \tag{4.74}$$

$$s.t. \quad E_{i+1} = E_i + r_i - e_{i,i+1}$$

$$0 \leq E_i \leq B, \quad E_1 \text{ given}$$

$$r_i \geq 0 \text{ for all } i \in \mathcal{N}$$

This is an LP where E_i and r_i are decision variables. Unlike the homogeneous charging node problem where the objective function includes charging prices p_i associated with nodes, i.e., $\sum_{i\in P} p_i r_i$, this is not the case here, since there is a tradeoff between selecting faster-charging nodes and possible higher costs at such nodes. However, the advantage of the decoupling approach is that if an optimal path is determined, an additional cost minimization problem can be formulated to determine optimal charging times at nodes on this path.

4.6.2 Multiple Vehicle Routing

Next, we investigate the system-centric problem in a network with inhomogeneous charging nodes. As in Section 4.3, we formulate the problem by grouping subsets of vehicles into N subflows.

We define R as the EV flow rate entering the network at node 1. Similar to our approach for the network with homogeneous charging nodes, we divide it into N subflows and formulate the problem at the subflow-level by assuming that all vehicles in the same subflow follow the same routing and recharging decisions. Clearly, it is not realistic to consider all vehicles in the system as EVs. In Section 4.5 we have addressed the routing problem for vehicle flows including both Electric Vehicles (EVs) and Non-Electric Vehicles (NEVs) for a network with homogeneous charging nodes. In (Pourazarm and Cassandras, 2015) we have shown that a similar framework and analysis as for the problem with the assumption of all inflow of vehicles as EVs, are applicable. Thus, in this section we focus on routing of EVs while the NEV flows are not part of our optimization process. Instead, we treat them as uncontrollable interfering traffic and assume that their flow rates are known. Our goal is to minimize the total elapsed time (latency) of the EVs traveling from origin to destination by determining optimal routes and energy recharging amounts for each vehicle subflow. Defining the state and decision variables as we did in Section 4.3, the problem is formulated as the following MINLP with $N(m + 2(n-1))$ variables:

$$\min_{\mathbf{x_{ij}},\mathbf{r_i},\ i,j\in\mathcal{N}} \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N}\left(\tau_{ij}(\mathbf{x_{ij}})x_{ij}^{k}\frac{R}{N} + r_i^k g_i x_{ij}^k\right) \tag{4.75}$$

$$s.t. \text{ for each } k \in \{1,\dots,N\}:$$

$$\sum_{j\in O(i)} x_{ij}^k - \sum_{j\in I(i)} x_{ji}^k = b_i, \quad \text{for each } i \in \mathcal{N} \tag{4.76}$$

$$b_1 = 1,\ b_n = -1,\ b_i = 0,\ \text{for } i \neq 1, n \tag{4.77}$$

$$E_j^k = \sum_{i\in I(j)} (E_i^k + r_i^k - e_{ij}^k(\mathbf{x_{ij}}))x_{ij}^k, \quad j = 2, \ldots, n \tag{4.78}$$

$$E_1^k \text{ is given}, \quad E_i^k \geq 0, \quad \text{for each } i \in \mathcal{N} \tag{4.79}$$

$$x_{ij}^k \in \{0, 1\}, \quad r_i^k \geq 0 \tag{4.80}$$

The difference from the MINLP formulated in Section 4.3 is that we consider different charging rates g_i in the objective function. In the sequel, we discuss some properties of the optimal solution allowing us to reduce the complexity of this MINLP problem as we did for the user-centric case.

4.6.3 Properties

It can be seen that for each subflow k, the constraints (4.76)-(4.80) are similar to those in the user-centric case, though the term $\tau_{ij}(\mathbf{x_{ij}})$ in the objective function is no longer linear in general. Consequently, we can derive similar useful properties in the form of the following lemmas (proofs are very similar to those of the user-centric case and are omitted).

Lemma 8: An optimal solution $\{\mathbf{x_{ij}}, \mathbf{r_i}\}$, $i, j \in \mathcal{N}$satisfies:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N} r_i^k g_i x_{ij}^k - \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N} e_{ij}^k(\mathbf{x_{ij}}) g_i x_{ij}^k \tag{4.81}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N} (E_j^k - E_i^k) g_i x_{ij}^k$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N} E_j^k (g_i - g_j) x_{ij}^k - \sum_{k=1}^{N} E_1^k g_1 \tag{4.82}$$

Lemma 9: If $\sum_i r_i^{k*} > 0$ in the optimal routing policy, then $E_n^{k*} = 0$ for $k = 1, ..., N$.

Using Lemma 3, we replace $\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N} r_i^k g_i x_{ij}^k$ in (4.75) through (4.82) and r_i^k, $i = 1, \ldots, n-1$, $k = 1, \ldots, N$, is eliminated from the objective function (4.75). The term $\sum_{k=1}^{N} E_1^k g_1$ is also removed because it has a fixed value. Thus, a new MINLP formulation with $N(m + (n-1))$ variables is obtained to determine x_{ij}^{k*} and E_i^{k*} for all $i, j \in \mathcal{N}$ and $k = 1, \ldots, N$ as follows:

$$\min_{\substack{x_{ij}^k, E_i^k \\ i,j \in \mathcal{N}}} \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N} \left[\tau_{ij}(\mathbf{x_{ij}}) x_{ij}^k \frac{R}{N} + (e_{ij}^k(\mathbf{x_{ij}}) g_i + E_j^k (g_i - g_j)) x_{ij}^k\right] \tag{4.83}$$

$$s.t. \text{ for each } k \in \{1, \ldots, N\}:$$

$$\sum_{j \in O(i)} x_{ij}^k - \sum_{j \in I(i)} x_{ji}^k = b_i \tag{4.84}$$

$$b_1 = 1,\ b_n = -1,\ b_i = 0 \quad \text{for } i \neq 1, n$$

$$0 \leq E_j^k - (E_i^k - e_{ij}^k(\mathbf{x_{ij}})) x_{ij}^k \leq B^k \quad \forall (i,j) \in A \tag{4.85}$$

$$E_1^k \text{is given,} \quad E_i^k \geq 0, \quad \text{for each } i \in N \tag{4.86}$$

$$x_{ij}^k \in \{0, 1\} \tag{4.87}$$

Note that inequality (4.85) is derived from (4.78). Assuming $x_{ij}^k = 1$, i.e., arc (i,j) is part of the optimal path for the kth subflow, $r_i^k = e_{ij}^k(\mathbf{x_{ij}}) + E_j^k - E_i^k$. Thus, (4.85) ensures the optimal solution E_i^{k*} results in a feasible charging amount for the kth subflow, $0 \leq r_i^k \leq B^k$ where B^k is the maximum charging amount kth subflow can get. This value should be predetermined for each subflow based on the vehicle types and the fraction of total inflow in it. Similar to our approach for the single-vehicle case, once we determine E_i^{k*} we can simply calculate optimal charging amounts using (4.78). Although this new formulation has fewer decision variables than the exact MINLP, its complexity still highly depends on the network size and number of subflows. Similar to the charging policy $\pi_{\mathbf{C}}$ used in Theorem 2, we introduce a charging policy by arguing as follows. Looking at (4.83), the term $\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N} E_j^k (g_i - g_j) x_{ij}^k$ is minimized by selecting each E_j^k depending on the sign of $(g_i - g_j)$:

Case 1: $g_i < g_j$, i.e., node i has faster charging rate than node j. Therefore, E_j^k should get its maximum value, i.e., the kth subflow should get its maximum charge at node i.

Case 2: $g_i \geqslant g_j$, i.e., the charging rate of node j is greater than or equal to node i. Therefore, E_j^k should get its minimum value of 0. This implies that the kth subflow should get the minimum charge needed at node i in order to reach node j.

Applying this policy in (4.83) and changing the objective function accordingly, we achieves the problem decomposition as follows:

$$\min_{x_{ij}^k} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{N} \left[\tau_{ij}(\mathbf{x_{ij}}) x_{ij}^k \frac{R}{N} + (e_{ij}^k(\mathbf{x_{ij}}) g_i + K(g_i - g_j)) x_{ij}^k\right] \tag{4.88}$$

$$K = \begin{cases} B^k - e_{ij}^k(\mathbf{x_{ij}}) & \text{if } g_i < g_j, \\ 0 & \text{otherwise} \end{cases} \tag{4.89}$$

$$s.t. \text{ for each } k \in \{1, \dots, N\}:$$

$$\sum_{j \in O(i)} x_{ij}^k - \sum_{j \in I(i)} x_{ji}^k = b_i \tag{4.90}$$

$$b_1 = 1, b_n = -1, b_i = 0 \quad \text{for } i \neq 1, n$$

$$x_{ij}^k \in \{0, 1\} \tag{4.91}$$

Unlike the user-centric case, the objective function is no longer necessarily linear in x_{ij}^k, therefore, (4.88) cannot be further simplified into an LP problem as in Theorem 2. Nonetheless, we are still able to decompose the original problem into two smaller problems: a MINLP to determine routing variables x_{ij}^k and a NLP to find recharging amounts r_i^k over the optimal routes. Similar to the single-vehicle case, once the optimal routes for all subflows, $P^k,\ k = \{1, ...N\}$, are determined, we can obtain r_i^k by formulating a corresponding NLP which minimizes $\sum_{k=1}^{N} \sum_{i \in P^k} r_i^k g_i$ while satisfying the energy constraints (4.78)-(4.79). The computational effort required to solve this problem with Nm decision variables, depends on the dimensionality of the network and the number of subflows.

On should note that if $g_i = g_j$ for all i, j in (4.88), the problem reduces to the homogeneous charging node case with the exact same MINLP formulation as in 4.3 for obtaining an optimal path. However, the decomposed problem here, cannot guarantee an optimal solution because of the locally optimal charging policy $\pi_{\mathbf{C}}$ which may not be feasible in a globally optimal solution (x_{ij}^{k*}, E_i^{k*}).

Next, we present an alternative formulation of (4.75)-(4.80) leading to a computationally simpler solution approach.

4.6.4 Flow control formulation

Similar to the flow control formulation for the network with homogeneous charging stations, we begin by relaxing the binary variables in (4.80) by letting $0 \leq x_{ij}^k \leq 1$. Following this relaxation, the objective function in (4.75) is changed to:

$$\min_{\mathbf{x_{ij}}, \mathbf{r_i},\ i,j \in \mathcal{N}} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{N} \tau_{ij}(\mathbf{x_{ij}}) x_{ij}^k \frac{R}{N} + \sum_{i=1}^{n} \sum_{k=1}^{N} r_i^k g_i$$

We then adjust the energy constraints and dynamics according to (4.28) and (4.29) and obtain the following simpler nonlinear programming problem (NLP):

$$\min_{\mathbf{x_{ij}}, \mathbf{r_i},\ i,j \in \mathcal{N}} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{N} \tau_{ij}(\mathbf{x_{ij}}) x_{ij}^k \frac{R}{N} + \sum_{i=1}^{n} \sum_{k=1}^{N} r_i^k g_i \tag{4.92}$$

$$s.t. \text{ for each } k \in \{1, \ldots, N\} :$$

$$\sum_{j \in O(i)} x_{ij}^k - \sum_{j \in I(i)} x_{ji}^k = b_i, \quad \text{for each } i \in \mathcal{N} \tag{4.93}$$

$$b_1 = 1,\ b_n = -1,\ b_i = 0,\ \text{for } i \neq 1, n$$

$$\left[\sum_{h\in I(i)}(E_{hi}^k - e_{hi}^k(\mathbf{x_{ij}})) + r_i^k\right] \cdot \frac{x_{ij}^k}{\sum_{h\in I(i)} x_{hi}^k} = E_{ij}^k \tag{4.94}$$

$$\frac{E_{ij}^k}{\sum_{j\in O(i)} E_{ij}^k} = \frac{x_{ij}^k}{\sum_{j\in O(i)} x_{ij}^k} \tag{4.95}$$

$$E_{ij}^k \geq 0, \tag{4.96}$$

$$0 \leq x_{ij}^k \leq 1, \quad r_i^k \geq 0 \tag{4.97}$$

As in our previous analysis, we are able to eliminate $\mathbf{r_i}$ from the objective function in (4.92) as follows.

Lemma 10: For each subflow $k = 1, \ldots, N$,

$$\begin{aligned}\sum_{i=1}^n r_i^k g_i &= \sum_{i=1}^n \sum_{j=1}^n e_{ij}^k(\mathbf{x_{ij}})g_i + \sum_{i=1}^n \sum_{j\in O(i)} E_{ij}^k g_i - \sum_{i=1}^n \sum_{h\in I(i)} E_{hi}^k g_i \\ &= \sum_{i=1}^n \sum_{j=1}^n e_{ij}^k(\mathbf{x_{ij}})g_i + \sum_{i=1}^n \sum_{j\in O(i)} E_{ij}^k (g_i - g_j)\end{aligned}$$

Proof: Multiplying (4.94) by g_i and summing over all $i = 1, \ldots, n$, then using (4.93) and (4.95) proves the lemma. ■

Using Lemma 5 we change the objective function (4.92) to:

$$\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^N \left(\tau_{ij}(\mathbf{x_{ij}})x_{ij}^k \frac{R}{N} + e_{ij}^k(\mathbf{x_{ij}})g_i\right) + \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^N E_{ij}^k(g_i - g_j) \tag{4.98}$$

Once again, we adopt a charging policy $\pi_{\mathbf{C}}$ as follows:

Case 1: If $g_i < g_j$, then E_{ij}^k gets its maximum value $(B^k - e_{ij}^k(\mathbf{x_{ij}}))x_{ij}^k$.

Case 2: If $g_i \geq g_j$, then E_{ij}^k gets its minimum value 0.

Applying this policy in (4.98) we can transform the objective function (4.92) to (4.99) and determine near-optimal routes x_{ij}^{k*} by solving the following NLP:

$$\min_{\substack{\mathbf{x_{ij}}\\ i,j\in\mathcal{N}}} \sum_{k=1}^{N}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[\tau_{ij}(\mathbf{x_{ij}})x_{ij}^{k}\frac{R}{N}+e_{ij}^{k}(\mathbf{x_{ij}})g_i+K(g_i-g_j)\right] \tag{4.99}$$

$$K=\begin{cases}(B^k-e_{ij}^k(\mathbf{x_{ij}}))x_{ij}^k & \text{if } g_i<g_j,\\ 0 & \text{otherwise}\end{cases}$$

$$s.t. \text{ for each } k\in\{1,\dots,N\}:$$

$$\sum_{j\in O(i)} x_{ij}^k-\sum_{j\in I(i)} x_{ji}^k=b_i, \quad \text{for each } i\in\mathcal{N}$$

$$b_1=1,\ b_n=-1,\ b_i=0, \text{ for } i\neq 1,n$$

$$0\le x_{ij}^k\le 1$$

Once again, there is no guarantee of global optimality due to applying charging policy $\pi_{\mathbf{C}}$. The values of r_i^k, $i=1,\dots,n$, $k=1,\dots,N$, can then be determined so as to satisfy the energy constraints (4.94)-(4.96), and minimizing $\sum_{k=1}^{N}\sum_{i\in P^k}^{n} r_i^k g_i$. Note that in the above formulation, the nonlinearity appears in the objective function due to the traffic congestion effect on traveling time and energy consumption. Thus, if $\tau_{ij}(\mathbf{x_{ij}})x_{ij}^k$ and $e_{ij}^k(\mathbf{x_{ij}})$ are convex functions, the NLP is a convex optimization problem and its solution can be found generally fast (Note that this is not in general the global optimum of the main problem). Finally, if $g_i=g_j$ for all i,j in (4.99), the problem reduces to the homogeneous charging node case with the same exact NLP flow control formulation as in 4.3.

4.6.5 Objective function selection using actual traffic data

We now explain how to estimate the delay function, $\tau_{ij}(\mathbf{x_{ij}})$, in either (4.75) or (4.99) using the same actual traffic dataset from the Eastern Massachusetts transportation network and based on the analysis given in (Zhang et al., 2016). We assume that the delay functions have the following form (Bertsimas et al., 2014):

$$\tau_{ij}\left(f_{ij}\right) = t_{ij}^{0} h\left(\frac{f_{ij}}{C_{ij}}\right), \tag{4.100}$$

where t_{ij}^{0} is the free-flow travel time of link $(i,j) \in \mathcal{A}$, $h(\cdot)$ is strictly increasing and continuously differentiable on $\mathbb{R}_+$, and f_{ij} and C_{ij} denote the flow and the effective capacity of link $(i,j) \in \mathcal{A}$ respectively. The goal is to estimate $h(\cdot)$ functions based on actual traffic data.

The dataset at our disposal, provided by the Boston Region Metropolitan Planning Organization (MPO), includes spatial average speeds and the flow capacity for each road segment of major roadways and arterial streets of Eastern Massachusetts (see Fig. 4·5).

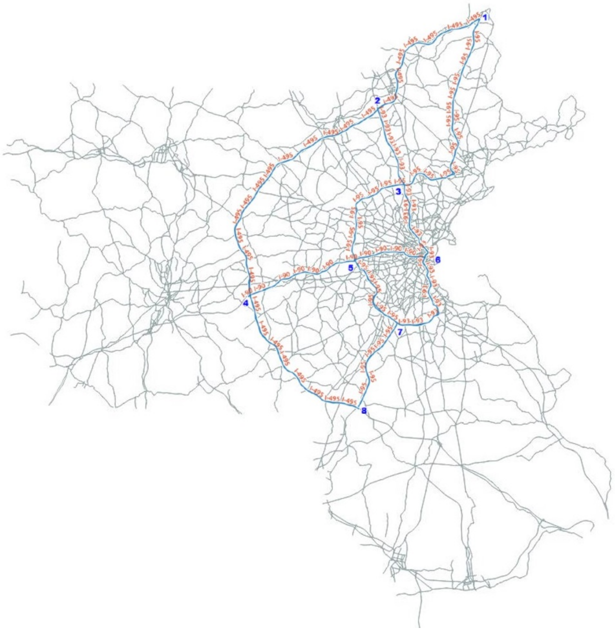
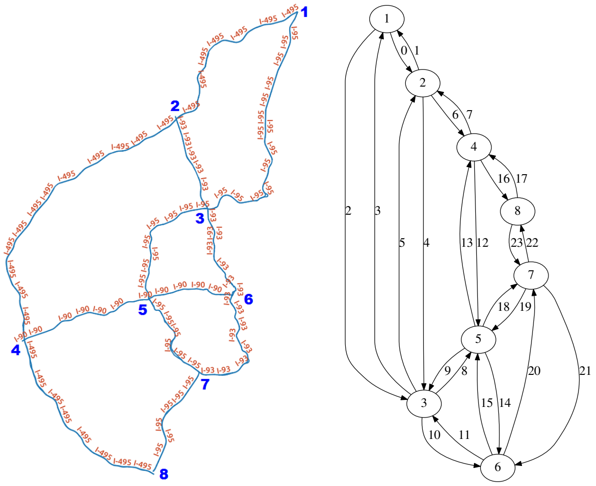


Figure 4·5: Road map of Eastern Massachusetts

We consider an interstate highway subnetwork, as shown in Fig. 4·6, and assume that the observed traffic data correspond to user (Wardrop) equilibria. Applying Greenshield's traffic flow model (Greenshields et al., 1935), we first convert the spatial



(a) (b)

Figure 4·6: (a) An interstate highway sub-network of Eastern Massachusetts (the blue numbers indicate node indices); (b) The topology of the sub-network

average speed data into equilibrium flows for each road segment. Then, by adopting the estimated traffic flows we obtain Origin-Destination (O-D) demand matrices. Finally, we formulate appropriate inverse problems (Bertsimas et al., 2014) to recover the per-road cost (congestion) functions determining user route selection for each month and time-of-day period (details are provided in (Zhang et al., 2016)). Applying polynomial kernels in the corresponding Quadratic Programming (QP) problem (Zhang et al., 2016), we estimate cost functions $h(\cdot)$ as polynomial functions. We estimate the cost functions for different scenarios: AM (7 am – 9 am), MD (11 am – 1 pm), PM (5 pm – 7 pm), and NT (9pm – 11 pm) for each day of January, April, July, and October, all in 2012.

The estimated $h(\cdot)$ functions corresponding to five different time periods for month April are shown in Fig. 4·7. We observe that the costs for the AM/PM peaks are much more sensitive to traffic flows than for the other three time periods (MD, NT, and weekend). This can be explained by taking into account the traffic condition during a day: from a congested road network in the AM/PM period to an uncongested road network during MD, NT, or weekend periods.

Let us consider the estimated equilibrium flow on each link as the uncontrolled NEV flow. Then, our goal is to determine system-optimal routes and charging policies for the EV flow entering the network. Let us assume that EVs enter the network at a rate of R veh./hr. We then evenly divide the EV inflow into N subflows and the total flow entering link (i, j) becomes:

$$f_{ij} = \sum_k x_{ij}^k \frac{R}{N} + f_{ij}^{eq} \tag{4.101}$$

where the first term represents the assignment of EV subflows to link (i, j) and the second term is the equilibrium flow for NEVs inferred from the average speed data. Therefore, the time a vehicle spends on link (i, j) becomes

$$\tau_{ij}(\mathbf{x_{ij}}) = t_{ij}^0 h \left(\frac{\sum_k (x_{ij}^k \frac{R}{N}) + f_{ij}^{eq}}{C_{ij}} \right) \tag{4.102}$$

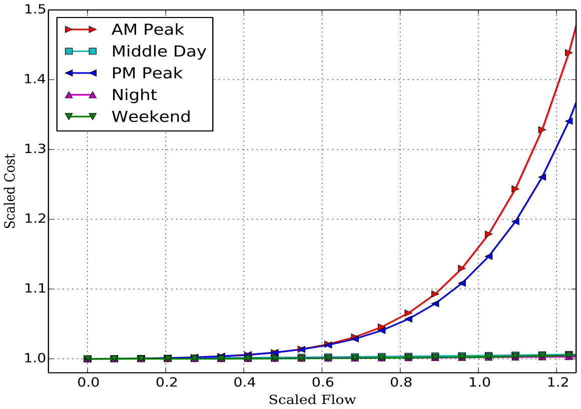


Figure 4·7: Comparison of the estimated cost functions corresponding to different time periods.

As for $e_{ij}^k(\mathbf{x_{ij}})$, we assume that the energy consumption rates of subflows on link

(i, j) are all identical, proportional to the distance between nodes i and j, giving

$$e_{ij}^{k}(\mathbf{x_{ij}}) = \alpha d_{ij} \frac{R}{N}$$

Therefore, the objective function in problems (4.88)-(4.91) and (4.99) in this case becomes

$$\min_{\substack{x_{ij}^{k} \\ i,j \in \mathcal{N}}} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{N} \left[t_{ij}^{0} h \left(\frac{\sum_k (x_{ij}^{k} \frac{R}{N}) + f_{ij}^{eq}}{C_{ij}} \right) x_{ij}^{k} \frac{R}{N} + \alpha d_{ij} g_i x_{ij}^{k} \frac{R}{N} + K(g_i - g_j) \right] \tag{4.103}$$

4.6.6 Numerical examples for the Eastern Massachusetts transportation network.

We consider the same sub-network shown in Fig. 4·6. Our goal is to determine system-optimal routes and charging policies for the flow of EVs traveling from node 1 to node 8 while the effect of NEV flows on the traffic congestion should be included in the cost function. As discussed in Section 4.6.5, we use real traffic data to calculate the uncontrolled NEV flow on each link. To do so, we use the average speed data on each road segment and infer the average flow data on that using the Greenshield's traffic flow model. Next, the calculated flow for all segments composing a link are aggregated in order to calculate the uncontrolled NEV flow on each link (Zhang et al., 2016). Tab. 4.11 shows the calculated average flow on each link of the sub-network on April 3 during AM period and we consider them as the user equilibrium flow, f_{ij}^{eq}, in (4.101). We then use the data-driven estimated cost function in our formulations. As stated earlier, the cost function is in polynomial form since we apply polynomial kernels in the corresponding QP problem. For the April-Workday-AM period, the

Table 4.11: Uncontrolled NEV flow on each link during AM period [No. veh/hr]

f_{12}^{eq}	f_{21}^{eq}	f_{13}^{eq}	f_{31}^{eq}	f_{23}^{eq}	f_{32}^{eq}	f_{24}^{eq}	f_{42}^{eq}	f_{35}^{eq}	f_{53}^{eq}	f_{36}^{eq}	f_{63}^{eq}
1287	1271	1725	1740	1713	3757	2231	172	3352	4870	3678	4218
f_{45}^{eq}	f_{54}^{eq}	f_{56}^{eq}	f_{65}^{eq}	f_{48}^{eq}	f_{84}^{eq}	f_{57}^{eq}	f_{75}^{eq}	f_{67}^{eq}	f_{76}^{eq}	f_{78}^{eq}	f_{87}^{eq}
4171	875	2464	2039	863	2099	4154	2802	3661	3776	1796	560

estimated $h(\cdot)$ function in (4.102) has the following form (red curve in Fig. 4·7):

$$\begin{aligned} h(x) =& 0.11x^8 - 0.4705x^7 + 0.946x^6 - 0.9076x^5 + 0.6238x^4 \\ & - 0.1973x^3 + 0.057x^2 - 0.0032x + 1 \end{aligned} \tag{4.104}$$

Applying this function in (4.102), we obtain the delay function for each link (i, j), τ_{ij}, based on actual traffic data. For the energy consumption function we set $\alpha = 0.3$ and distances between nodes are as shown in Tab. 4.12.

Table 4.12: Distance [mile] of all links in the sub-network in Fig. 4·6

d_{12}	d_{21}	d_{13}	d_{31}	d_{23}	d_{32}	d_{24}	d_{42}	d_{35}	d_{53}	d_{36}	d_{63}
21.49	22.83	32.03	32.7	10.08	11.98	37.63	38.67	16.21	16.8	12.88	12.94
d_{45}	d_{54}	d_{56}	d_{65}	d_{48}	d_{84}	d_{57}	d_{75}	d_{67}	d_{76}	d_{78}	d_{87}
6.94	17.16	10.77	10.68	24.43	24.37	12.51	12.33	16.51	16.3	13.92	14.19

We assume the network has inhomogeneous charging nodes with a level 2 charging station at node 3 (charging rate of $g^2 = 1/6$ [hr/kWh]) and level 1 charging stations (charging rate of $g^1 = 41.67/60$ [hr/kWh]), for the rest, i.e., $G = [g^1\ g^1\ g^2\ g^1\ g^1\ g^1\ g^1]$. In our approach, we need to identify N subflows and we do so by evenly dividing the entire vehicle inflow into N subflows, each of which has R/N vehicles per unit time. In order to verify the accuracy of different formulations, we numerically solve the optimal (exact MINLP) and near-optimal problems (decomposed MINLP). Let us set $R = 1492$ [Veh./hr] as the flow of EVs traveling from node 1 to node 8. Tab. 4.13 shows both optimal routes and suboptimal routes obtained by solving both formulations for different values of $N \in [1, \ldots, 11]$ and $G = [\frac{41.67}{60} \ \frac{41.67}{60} \ \frac{1}{6} \ \frac{41.67}{60} \ \frac{41.67}{60} \ \frac{41.67}{60} \ \frac{41.67}{60}]$. We observe that vehicles are mainly distributed through two routes and the traffic congestion effect makes the flow distribution differ from the shortest path, $1 \to 3 \to 5 \to 7 \to 8$. The number of decision variables (hence, the solution search space) rapidly increases with the number of subflows. However, looking at Fig. 4·8 which gives the performance in terms of our objective functions in (4.75) and (4.88) as a function of the number of subflows, observe that the optimal objective value (exact MINLP) quickly converges around $N = 3$. Thus, even though the best solution is found when $N = 11$, a near-optimal solution can be determined under a small number of subflows. This suggests that one can rapidly approximate the asymptotic solution of the multiple-vehicle problem (dealing with individual vehicles routed so as to optimize a system-wide objective) based on a relatively small value of N.

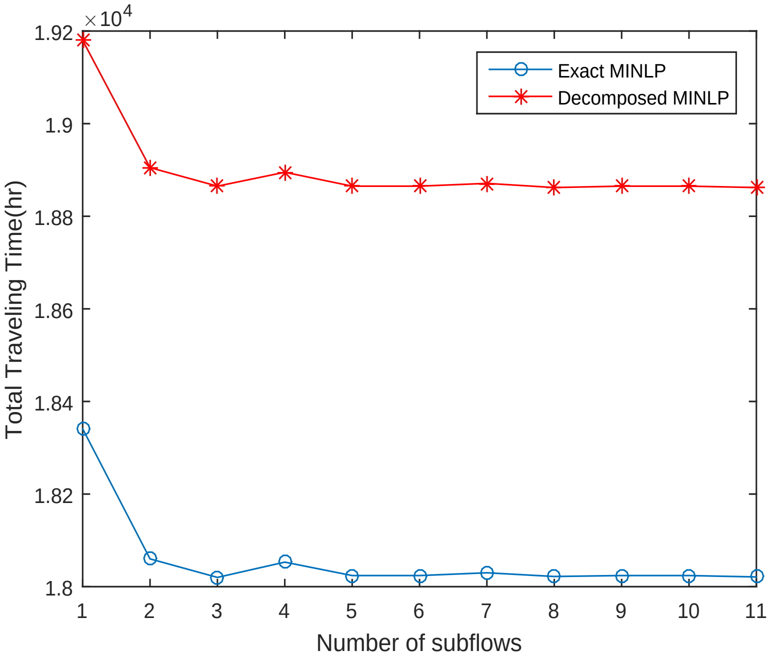


Figure 4·8: Performance as a function of N (No. of subflows)

Another observation is that although the decomposed problem is a suboptimal

Table 4.13: Numerical results for sample problem

	P3	**P5**
N	1	1
obj	1.8341e4	1.9182e4
routes	$1 \to 3 \to 5 \to 7 \to 8$	$1 \to 3 \to 5 \to 7 \to 8$
N	2	2
obj	1.806e4	1.8905e4
routes	$1 \to 3 \to 5 \to 7 \to 8$ $1 \to 3 \to 6 \to 7 \to 8$	$1 \to 3 \to 5 \to 7 \to 8$ $1 \to 3 \to 6 \to 7 \to 8$
N	3	3
obj	1.8024e4	1.8865e4
routes	$1 \to 3 \to 5 \to 7 \to 8(\times 2)$ $1 \to 3 \to 6 \to 7 \to 8$	$1 \to 3 \to 5 \to 7 \to 8(\times 2)$ $1 \to 3 \to 6 \to 7 \to 8$
N	4	4
obj	1.8053e4	1.8895e4
routes	$1 \to 3 \to 5 \to 7 \to 8(\times 3)$ $1 \to 3 \to 6 \to 7 \to 8$	$1 \to 3 \to 5 \to 7 \to 8(\times 3)$ $1 \to 3 \to 6 \to 7 \to 8$
N	5	5
obj	1.8024e4	1.8865e4
routes	$1 \to 3 \to 5 \to 7 \to 8(\times 3)$ $1 \to 3 \to 6 \to 7 \to 8(\times 2)$	$1 \to 3 \to 5 \to 7 \to 8(\times 3)$ $1 \to 3 \to 6 \to 7 \to 8(\times 2)$
N	6	6
obj	1.8024e4	1.8865e4
routes	$1 \to 3 \to 5 \to 7 \to 8(\times 4)$ $1 \to 3 \to 6 \to 7 \to 8(\times 2)$	$1 \to 3 \to 5 \to 7 \to 8(\times 4)$ $1 \to 3 \to 6 \to 7 \to 8(\times 2)$
N	7	7
obj	1.803e4	1.8871e4
routes	$1 \to 3 \to 5 \to 7 \to 8(\times 4)$ $1 \to 3 \to 6 \to 7 \to 8(\times 3)$	$1 \to 3 \to 5 \to 7 \to 8(\times 4)$ $1 \to 3 \to 6 \to 7 \to 8(\times 3)$
N	8	8
obj	1.8022e4	1.8862e4
routes	$1 \to 3 \to 5 \to 7 \to 8(\times 5)$ $1 \to 3 \to 6 \to 7 \to 8(\times 3)$	$1 \to 3 \to 5 \to 7 \to 8(\times 5)$ $1 \to 3 \to 6 \to 7 \to 8(\times 3)$
N	9	9
obj	1.8024e4e4	1.8865e4
routes	$1 \to 3 \to 5 \to 7 \to 8(\times 6)$ $1 \to 3 \to 6 \to 7 \to 8(\times 3)$	$1 \to 3 \to 5 \to 7 \to 8(\times 6)$ $1 \to 3 \to 6 \to 7 \to 8(\times 3)$
N	10	10
obj	1.8024e4	1.8865e4
routes	$1 \to 3 \to 5 \to 7 \to 8(\times 6)$ $1 \to 3 \to 6 \to 7 \to 8(\times 4)$	$1 \to 3 \to 5 \to 7 \to 8(\times 6)$ $1 \to 3 \to 6 \to 7 \to 8(\times 4)$
N	11	11
obj	1.8021e4	1.8862e4
routes	$1 \to 3 \to 5 \to 7 \to 8(\times 7)$ $1 \to 3 \to 6 \to 7 \to 8(\times 4)$	$1 \to 3 \to 5 \to 7 \to 8(\times 7)$ $1 \to 3 \to 6 \to 7 \to 8(\times 4)$

formulation it results in the same paths as those obtained by solving the exact MINLP. Next, we obtain a solution to the same problem using the NLP formulation (4.99) with $0 \leq x_{ij}^k \leq 1$. Since in this example all subflows are identical, we can further combine all x_{ij}^k over each (i,j), leading to the N-subflow relaxed problem:

$$\begin{aligned}
\min_{\substack{x_{ij} \\ i,j\in\mathcal{N}}} \sum_{i=1}^{n}\sum_{j=1}^{n} \Big[& t_{ij}^0 h\left(\frac{x_{ij}R + f_{ij}^{eq}}{C_{ij}}\right) x_{ij}R + \\
& \alpha d_{ij} g_i x_{ij} R + K(g_i - g_j)\Big] \qquad (4.105) \\
& K = \begin{cases} (B - e d_{ij} R) x_{ij} & \text{if } g_i < g_j, \\ 0 & \text{otherwise} \end{cases} \\
s.t. \quad & \sum_{j\in O(i)} x_{ij} - \sum_{j\in I(i)} x_{ji} = b_i, \quad \text{for each } i \in \mathcal{N} \\
& b_1 = 1,\ b_n = -1,\ b_i = 0,\ \text{for } i \neq 1, n \\
& 0 \leq x_{ij} \leq 1
\end{aligned}$$

This is a relatively easy to solve NLP problem. It is obvious from Fig. 4·7 that the $h(\cdot)$ function during AM period is a strictly convex function, thus the solution of this NLP is a unique global optimum. Using the same parameter settings as before, we obtain the objective value of $1.8862e45$ hrs and the optimal routes are:

63.24% of vehicle flow:	$(1 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8)$
36.76% of vehicle flow:	$(1 \rightarrow 3 \rightarrow 6 \rightarrow 7 \rightarrow 8)$

Compared to the best solution ($N = 11$) in Fig. 4·8, the difference in objective values between the integer and flow-based solutions is less than 4.7%. This supports the effectiveness of a solution based on a limited number of subflows in the MINLP problem.

Effect of EV inflow on optimal routes. Tab. 4.14 shows the optimal paths obtained by solving both exact MINLP and the NLP relaxed problems (4.105) for different values of EV inflow, R. It is observed that the optimal routes will change with the inflow rate. For lower flows, e.g., $R = 300$, it is optimal that all EVs travel through the shortest path which means the corresponding change in $\frac{f_{ij}}{C_{ij}}$ in (4.100) is negligible. However, higher flows may cause congestion in some links, i.e., larger values of $\frac{f_{ij}}{C_{ij}}$ in (4.100), resulting in larger delays on those links τ_{ij}. Consequently, some EVs should deviate from the shortest path in the optimal routing.

Table 4.14: Effect of flow rate, R, on Optimal routes

<table>
<tr><td colspan="3">R=300 [Veh./hr]</td></tr>
<tr><td><b>P3</b></td><td>$N = 1$</td><td>$N = 2$</td></tr>
<tr><td>routes</td><td>$1 \to 3 \to 5 \to 7 \to 8$</td><td>$1 \to 3 \to 5 \to 7 \to 8 (\times 2)$</td></tr>
<tr><td colspan="3"><b>NLP</b> ⇒ 100% of EV flow:$(1 \to 3 \to 5 \to 7 \to 8)$</td></tr>
<tr><td colspan="3">R=1492 [Veh./hr]</td></tr>
<tr><td><b>P3</b></td><td>$N = 1$</td><td>$N = 11$</td></tr>
<tr><td>routes</td><td>$1 \to 3 \to 5 \to 7 \to 8$</td><td>$1 \to 3 \to 5 \to 7 \to 8 (\times 7)$
$1 \to 3 \to 6 \to 7 \to 8 (\times 4)$</td></tr>
<tr><td colspan="3"><b>NLP</b> ⇒ 63.24% of EV flow:$(1 \to 3 \to 5 \to 7 \to 8)$
36.76% of EV flow:$(1 \to 3 \to 6 \to 7 \to 8)$</td></tr>
<tr><td colspan="3">R=2984 [Veh./hr]</td></tr>
<tr><td><b>P3</b></td><td>$N = 1$</td><td>$N = 7$</td></tr>
<tr><td>routes</td><td>$1 \to 3 \to 5 \to 7 \to 8$</td><td>$1 \to 3 \to 5 \to 7 \to 8 (\times 3)$
$1 \to 3 \to 6 \to 7 \to 8 (\times 3)$
$1 \to 2 \to 3 \to 5 \to 7 \to 8$</td></tr>
<tr><td colspan="3">42.21% of EV flow:$(1 \to 3 \to 5 \to 7 \to 8)$
<b>NLP</b> ⇒ 41.32% of EV flow:$(1 \to 3 \to 6 \to 7 \to 8)$
16.47% of EV flow:$(1 \to 2 \to 3 \to 5 \to 7 \to 8)$</td></tr>
</table>

CPU time Comparison. Tab. 4.15 compares the computational effort in terms of CPU time for exact and decomposed MINLP problems and the flow control formulation to find optimal routes for the sample network shown in Fig. 4·6. Our results show that the flow control formulation provides a reduction of about 3 orders of magnitude in CPU time with almost the same solution as the optimal solution.

Selection of the number of subflows. Since the problem size increases with

Table 4.15: CPU time for sample problem

Fig.4·6	**P3**	**P5**	NLP approx.
N	8(near opt)	8(near opt)	-
CPU time(sec)	48179	10267	54

the number of subflows, N, a proper selection of this number is essential to render the problem computationally manageable and reflects a trade-off between proximity to optimality and computational effort needed to solve the problem. Our numerical results have shown that a small number of subflows are adequate to obtain convergence to near-optimal solutions. In Section 4.4 we have proposed a criterion and procedure for appropriate choice of the number of subflows for the network with homogeneous charging stations. In brief, the key idea is based on the fact that the decomposed MINLP problem (4.88)-(4.91) obtains the optimal solution for the homogeneous network ($g_i = g_j\ \forall i, j$), thus the corresponding relaxed NLP, i.e., problem (4.105) with $g_i = g_j$, gives a lower bound for the optimal objective value. We then defined a critical number of subflows, N^*, which guarantees near optimality and showed that by selecting N so that $N > N^*$, the average deviation between NLP solution and MINLP solution with N subflows never exceeds a predefined upper bounds.

For the network with inhomogeneous charging stations one should note that adopting a locally optimal charging policy, the decomposed MINLP (4.88)-(4.91) is suboptimal in general. Therefore, the corresponding relaxed NLP does not give a lower bound for the optimal objective value, though it does for the decomposed suboptimal MINLP. Nevertheless, since the routes obtained by solving the decomposed MINLP are near-optimal and the relaxed NLP gives a lower bound for its objective value, we may still use the same procedure as in Section 4.4 for selecting a “good” N. In our numerical results, it is observed that both exact and decomposed MINLP problems result in the same solutions for different values of N. Furthermore, for the value N

with the lowest objective value, $N = 11$, the normalized flow on each path, that is $4/11 = 36.36\%$ for $1 \rightarrow 3 \rightarrow 6 \rightarrow 7 \rightarrow 8$ and $7/11 = 63.64\%$ for $1 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8$, has the least deviation from the solution of the NLP problem which is the main idea in the selection of N.

4.7 Summary

We have introduced energy constraints into vehicle routing in traffic networks and studied the problem of minimizing the total elapsed time for vehicles to reach their destinations by determining routes, as well as recharging amounts, when there is no adequate energy for the entire journey. We have studied the problem in two different settings: in network with homogeneous charging nodes vs in network with inhomogeneous charging nodes. For a single vehicle problem (user-centric problem), we have shown how to decompose this problem into two simpler problems. For a multi-vehicle problem (system-centric problem), we solved the problem by aggregating vehicles into subflows and seeking optimal routing decisions for each such subflow. One critical factor in this problem is the selection of the number of subflows. Our numerical results showed that a small number of subflows is adequate to obtain convergence to near-optimal solutions. Thus, we defined a critical number of subflows which guarantees near-optimality. In particular, we show that by selecting the number of subflows to be equal to or larger than a critical number N^*, i.e., $N \geq N^*$, the average deviation never exceeds the predefined upper bound. Therefore, by selecting a desired complexity needed to solve the problem. We also reformulated the multi-vehicle routing problem in order to incorporate the effect of NEVs on traffic congestion. We then applied real traffic data from the Eastern Massachusetts transportation network and investigated the user-optimal vs social-optimal routing policies for different scenarios.

Chapter 5

Optimal Routing of Electric Vehicles in Networks with Charging Nodes: A Dynamic Programming Approach

5.1 Introduction

In Chapter 4, we studied the vehicle total traveling time minimization problem in a network containing inhomogeneous charging nodes. For the single EV routing problem, formulated as a MINLP, we proved certain optimality properties allowing us to reduce the dimensionality of the original problem. Further, by adopting a locally optimal charging policy, we derived a Linear Programming (LP) formulation through which near-optimal solutions are obtained. For a multi-vehicle problem, where traffic congestion effects are included and a system-wide objective is considered, a similar approach was used by grouping vehicles into subflows. Despite the properties of the problem that we have exploited, its solution remains computationally demanding for real-time applications. This motivates the study of alternative solution techniques.

Thus, in this chaper we formulate the single EV routing problem as a Dynamic Programming (DP) problem by discretizing vehicle residual energy at each node. This model is identical for both homogeneous and inhomogeneous charging nodes and allows us to find an optimal routing and charging policy for both cases in CPU time which is about two orders of magnitude lower compared to solving the MINLP problem introduced in Chapter 4. We then study the much more challenging multi-

EV routing problem, where a traffic flow model is used to incorporate congestion effects. Similar to our approach in Chapter 4, by grouping vehicles into subflows we are able to reduce the complexity of the original problem and provide a DP-based algorithm to determine optimal routing and charging policies at the EV subflow level. In this case, the problem size significantly increases with the number of subflows and the DP algorithm is eventually outperformed by our earlier MINLP approach as the number of subflows increases.

The structure of the chapter is as follows. In Section 5.2, we address the single-EV routing problem in a network with inhomogeneous charging nodes and formulate it as a DP problem. We then derive an iterative algorithm to solve it recursively. In Section 5.3, the multi-EV routing problem is also formulated as a DP. Simulation examples are included illustrating our approach and providing insights on the relationship between recharging speed and optimal routes.

5.2 Single Vehicle Routing

Similar to Chapter 4, we assume that a network is defined as a directed graph $G = (\mathcal{N}, \mathcal{A})$ with $\mathcal{N} = \{1, \dots, n\}$ and $|\mathcal{A}| = m$. Node $i \in \mathcal{N}/\{n\}$ represents a charging station and $(i, j) \in \mathcal{A}$ is an arc connecting node i to j. First, we deal with a single-origin-single-destination vehicle routing problem in a network of inhomogeneous charging stations. Nodes 1 and n respectively are defined to be the origin and destination. For each arc $(i, j) \in \mathcal{A}$, there are two cost parameters: the required traveling time τ_{ij} and the energy consumption e_{ij}. Letting the vehicle's charge capacity be B, we assume that $e_{ij} < B$ for all $(i, j) \in \mathcal{A}$. As explained in Chapter 4, τ_{ij} and e_{ij} are fixed depending on given traffic conditions at the time the single-vehicle routing problem is solved. Since the EV has limited battery energy, it may not be able to reach the destination without recharging. Thus, recharging amounts at charging nodes $i \in \mathcal{N}$ are also decision variables.

5.2.1 Dynamic Programming Formulation

In Chapter 4, first we formulated this problem as a MINLP which is computationally expensive. We then proceeded by decomposing it into two linear programming (LP) problems obtainng a near-optimal solution (for networks with inhomogeneous charging nodes). Here, we formulate the same problem in a DP setting and obtain optimal (not just near-optimal) solutions. The algorithm is based on the following formulation.

We define $Q(i, E_i)$ to be the minimum elapsed time, including traveling and recharging times, to the destination node when starting at node i with E_i units of energy. Our goal, therefore, is to determine $Q(1, E_1)$ where E_1 is given. Assuming the EV maximum charging capacity is B, we have to consider all possible values of $E_i \in [0, B]$. To do so, we discretize the range $[0, B]$ and form a set of all possible values for E_i. Our algorithm is centered on the standard principle of optimality (Bertsekas, 2012) based on which, $Q(i, E_i)$ is obtained using the following iterative equation:

$$Q(i, E_i) = \min_{\substack{j \in O(i),\ 0 \leq E_j \leq B \\ \text{s.t } 0 \leq E_j - E_i + e_{ij} \leq B}} \big[\overbrace{Q(j, E_j)}^{\text{Cost to go}} + \overbrace{\tau_{ij} + (E_j - E_i + e_{ij}) g_i}^{\text{One step cost}} \big] \tag{5.1}$$

where the state is $[i, E_i]$ and there are two control variables: the amount to charge at each state, r_i, and the next node to route the EV to, $j \in O(i)$, dictated by the graph topology. The charge amount r_i is constrained by the energy dynamics, $E_j = E_i + r_i - e_{ij}$ and by $0 \leq r_i \leq B$. This iterative process leads to the optimal solution because when an optimal policy is found from state $[j, E_j]$ to the destination for all feasible values of j and E_j, then the route from node i to the destination node via node j will also be optimal. Under proper technical conditions, the iterative process generated through (5.1) converges to the optimal value of Q. The detailed steps of the DP algorithm for this problem are given next.

Initialization: We have shown in previous chapter that if an EV receives any positive charge in the optimal path, i.e., $\sum_i r_i^* > 0$, the EV residual energy at the destination is zero, i.e., $E_n^* = 0$. Therefore, the cost value at the destination node is $Q^{(0)}(n, 0) = 0$. Motivated by Dijkstra's algorithm for the shortest path problem, we set the initial elapsed time for all other states to infinity, i.e.,

$$Q^{(0)}(n, E_n) = \begin{cases} 0 & \text{if } E_n = 0, \\ \infty & \text{if } E_n > 0. \end{cases} \tag{5.2}$$

$$Q^{(0)}(i, E_i) = \infty \quad \forall i \in \mathcal{N} \setminus n, \quad 0 \le E_i \le B \tag{5.3}$$

Iteration steps: The update of Q values can be carried out starting from any state. For convenience, we start at the source node, i.e., $[1, E_1]$. At the kth iteration, the Q values are updated as follows: $Q^{(k)}(n, E_n) = Q^{(0)}(n, E_n)$ and

$$Q^{(k)}(i, E_i) = \min_{\substack{j \in O\{i\},\ 0 \le E_j \le B \\ \text{s.t } 0 \le E_j - E_i + e_{ij} \le B}} [Q^{(k-1)}(j, E_j) + \tau_{ij} + (E_j - E_i + e_{ij}) g_i]$$

$$\forall i \in \mathcal{N} \setminus n, \quad 0 \le E_i \le B \tag{5.4}$$

We seek $\lim_{k \to \infty} Q^{(k)}(i, E_i) = Q^*$, therefore, the algorithm stops when $Q^{(k)}(i, E_i) = Q^{(k-1)}(i, E_i)$ for all $i \in \mathcal{N}$, $0 \le E_i \le B$. The optimal route can then be determined by choosing the next state, minimizing $Q(i, E_i)$. Without loss of generality, we re-index nodes so that we may write the optimal path as $P = \{1, ..., m\}$. Then, the optimal charging amount at each node on the optimal path is calculated through $r_{i-1} = E_i - E_{i-1} + e_{i-1,i}$ with $i = 2, ..., m$, and E_1 given.

5.2.2 Numerical Example

To investigate the effectiveness of the DP algorithm, we consider a grid graph with 49 nodes and 84 edges as shown in Fig. 5·1, where the traveling time, τ_{ij}, and energy consumption, e_{ij} on each edge are shown in red and blue numbers respectively. Fig. 5·1 shows the optimal path for the network with homogeneous charging stations $(G = [g_1, ..., g_{n-1}], \quad g_i = 1 \quad \forall\ i)$ and inhomogeneous charging stations $G =$[1 1 1 1 1 1 1 5 5 1 1 1 1 1 1 5 1 1 11 11 5 11 1 1 1 1 5 1 1 1 1 1 1 5 5 5 1 1 1 1 1 1 5 5 5] as the green and red routes respectively. For the network

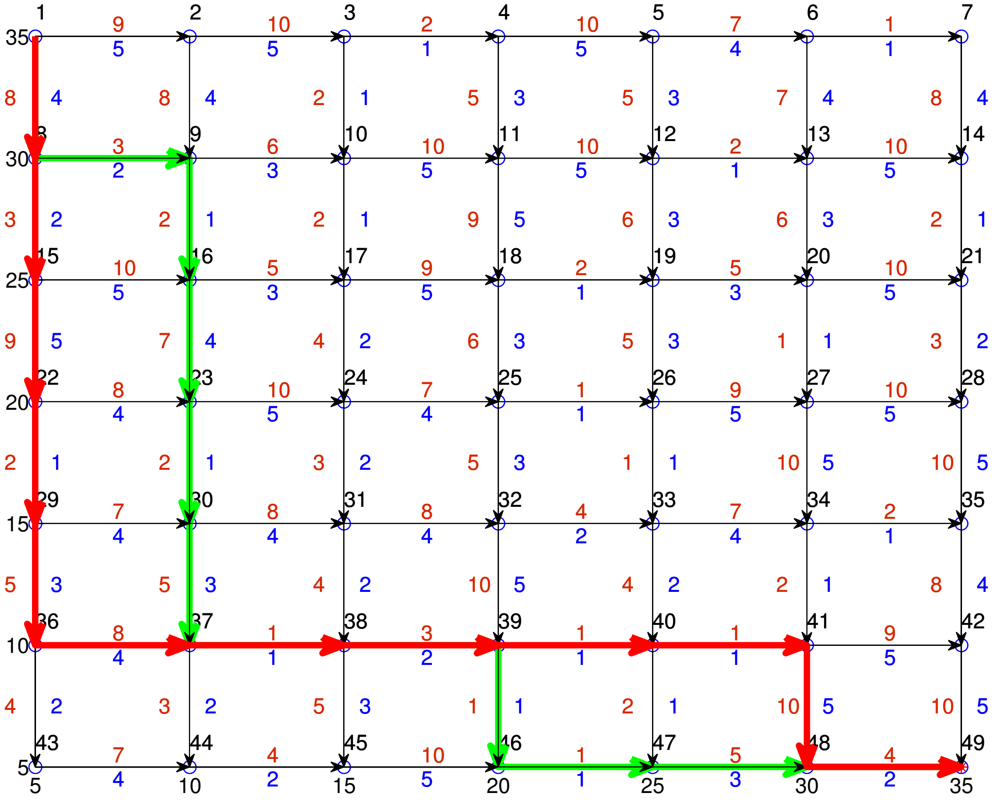


Figure 5·1: A 49-node grid network with inhomogeneous charging nodes.

with homogeneous stations, the optimal charging policy suggests that the EV requires just enough charge at each node to reach the next node on the optimal path, e.g., if $E_1 = 0$ then $r_i^* = e_{i,i+1}$, $i \in P$. In contrast, for the network with inhomogeneous charging nodes with a G vector as above, the optimal charging amount at each node on the optimal path is as follows:

$$\begin{array}{llllll} r_1 = 5 & r_2 = 5 & r_3 = 1 & r_{10} = 1 & r_{17} = 2 & r_{24} = 2 \\ r_{31} = 5 & r_{38} = 0 & r_{39} = 0 & r_{40} = 6 & r_{47} = 0 & r_{48} = 0 \end{array}$$

The algorithm execution is very fast for this graph and converges to the optimal solution in 13 iterations in less than 10 sec for both homogeneous and inhomogeneous charging nodes. In contrast, a MINLP solver requires more than 1000 sec to find the optimal solution for the same graph with homogeneous charging nodes.

5.3 Multiple Vehicle Routing

The results obtained for the single vehicle routing problem pave the way for the investigation of multi-vehicle routing, where we seek to optimize a *system-wide* objective by routing and charging vehicles through some network topology. This is a much more challenging problem, the main technical difficulty being the need to consider the influence of traffic congestion on both traveling time and energy consumption.

As in Chapter 4, we proceed by grouping subsets of vehicles into N subflows where N may be selected to render the problem manageable. Let all vehicles enter the network at node 1 and let R denote the rate of vehicles arriving at this node. Viewing vehicles as defining a flow, we divide them into N subflows. Thus, all vehicles in the same subflow follow the same routing and recharging decisions so that we only consider control at the subflow level rather than individual vehicles. Our objective is to determine optimal routes and energy recharging amounts for each vehicle subflow so as to minimize the total elapsed time of these flows from origin to destination.

5.3.1 Dynamic Programming Formulation

Our goal here is to develop a DP algorithm to solve this problem and compare its computational cost to the solution methods in Chapter 4. Note that the problem size dramatically increases with the number of subflows, N. Our first step is to construct a new graph at the subflow level, $G_{sf} = (\mathcal{N}_{sf}, \mathcal{A}_{sf})$, given a road network $G = (\mathcal{N}, \mathcal{A})$ and the number of subflows, N. In this graph, each node in $\mathcal{N}_{sf}$ represents a feasible combination of nodes in G among which all subflows may be distributed. To make this clear, consider the road network shown in Fig. 5·2. In order to map the original graph G into the subflow-level graph G_{sf}, we define each of its nodes as $\mathbf{Y}^i = (y_1^i ..., y_N^i)$ where $i = 1, 2, \ldots$ indexes these nodes and y_k^i is the location of the kth subflow in G. Fig. 5·3 is the subflow-level graph G_{sf} constructed from Fig. 5·2 when the total inflow, R, is divided into 2 subflows ($N = 2$). In this case, G_{sf} consists of 25 nodes. As an example, in Fig. 5·3 node 3(2 4) represents a node with index $i = 3$ mapping the first and second subflows to nodes 2 and 4 in G respectively, i.e., it represents a routing decision at node 1 in G for sublow 1 to travel from 1 to 2 and for sublow 2 to travel from 1 to 4, noting that $O(1) = \{2, 4, 5\}$. Clearly, G_{sf} is much larger than the original road network G, even for $N = 2$. Table 5.1 shows the number of nodes and edges in the subflow-level graph for different values of N for the road network in Fig. 5·2. As we did for the single EV, we need to consider all possible

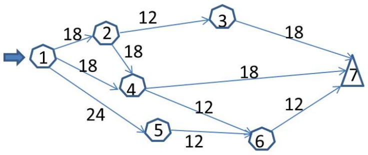


Figure 5·2: A 7-node road network with inhomogeneous charging nodes.

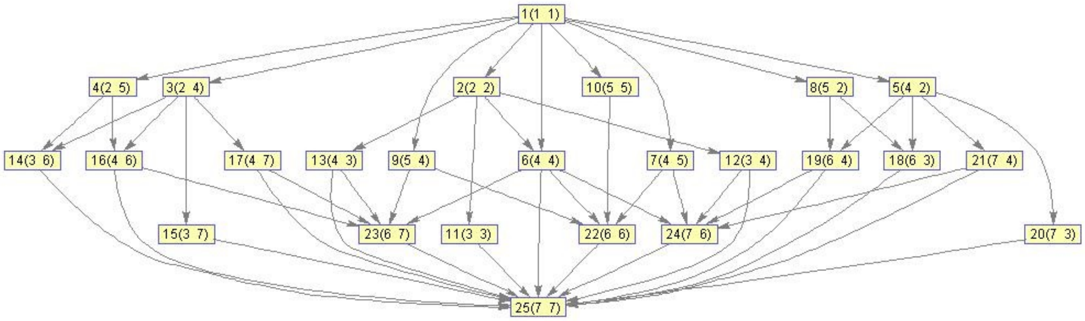


Figure 5·3: Subflow-level graph showing all feasible combination of nodes via which subflows may travel

Table 5.1: Subflow-level graph size for different number of subflows for road network shown in Fig. 5·1

Number of subflows (N)	Number of nodes	Number of edges
2	25	54
3	91	268
6	4825	31914

combinations for the residual energies at each node in the subflow-level graph. Thus, we define $\mathbf{E}^i = [E^i_1, ..., E^i_N]$. When a decision is made at a node in G_{sf}, we need to calculate its effect on the travelling time and energy consumption over each edge $(i, j) \in \mathcal{A}$ due to the potential traffic added to this edge. This requires information on the number of subflows routed through each link (i, j). Recalling that $|\mathcal{A}| = m$, let us index all edges $(i, j) \in \mathcal{A}$ as $\{1, ..., m\}$. Next, we define an auxiliary vector for each pair $(\mathbf{Y}^i, \mathbf{E}^i)$ in G_{sf} denoted by $\mathbf{S}^i = [s^i_1, ..., s^i_m]$ where s^i_l is the number of subflows through the lth edge in $G = (\mathcal{N}, \mathcal{A})$ starting from node $\mathbf{Y}^i \in \mathcal{N}_{sf}$ with residual energies $\mathbf{E}^i$, i.e., $s^i_l \in \{0, 1, ..., N\}$ and $l = 1, ..., m$. In other words, $\mathbf{S}^i$ is a function of the state variables $(\mathbf{Y}^i, \mathbf{E}^i)$ and includes the data required to calculate traveling time and energy consumption amounts on each edge. Specifically, traveling from node $\mathbf{Y}^i$ to node $\mathbf{Y}^j$ in the G_{sf}, $\tau^k_{y_i,y_j}$ and $e^k_{y_i,y_j}$ represent the traveling time and energy consumption on the edge (y_k^i, y_k^j) in the original graph for the kth subflow respectively and their values depend on the traffic congestion on the edge which is a function of $s_l^i \in \mathbf{S}^i$. More precisely, we define the "edge indexing operation", $\delta(y_k^i, y_k^j)$, assigning a single edge index l to a pair of node indices (y_k^i, y_k^j), i.e., $\delta(y_k^i, y_k^j) = l$. Note that s_l^i is updated based on the decision made at node $\mathbf{Y}^i$ which determines the next node, $\mathbf{Y}^j$, and residual energy $\mathbf{E}^j$. Clearly,

$$s_l^i = s_l^j + \sum_{k=1}^{N} \mathbf{1}[\delta(y_k^i, y_k^j) = l] \tag{5.5}$$

where $\mathbf{1}[.]$ is the indicator function. Thus, the term $\sum_{k=1}^{N} \mathbf{1}[\delta(y_k^i, y_k^j) = l]$ captures the added congestion imposed by edge $(\mathbf{Y}^i, \mathbf{Y}^j) \in \mathcal{A}_{sf}$ on the lth edge in $\mathcal{A}$. Let $\mathbf{S}(i,j)$ be defined as the m-dimensional vector with the qth element $s_q(i,j) = \sum_{k=1}^{N} \mathbf{1}[\delta(y_k^i, y_k^j) = q]$ and $q = 1, ..., m$. Therefore, (5.5) can be written in vector form as: $\mathbf{S}^i = \mathbf{S}^j + \mathbf{S}(i,j)$. It's worth to mention that, in contrast with the single-EV problem where we assume fixed parameters for the traveling time and energy consumption on each edge, for the multiple-EV problem these parameters are dependent on the traffic congestion (routing decision) which makes the problem much harder.

We define $Q(\mathbf{Y}^i, \mathbf{E}^{\mathbf{i}})$ to be the minimum total elapsed time to the destination node in G_{sf} starting from node $\mathbf{Y}^i = (y_1^i, ..., y_N^i) \in \mathcal{N}_{sf}$ with $\mathbf{E}^i = (E_1^i, ..., E_N^i)$ units of energy. Our goal then is to determine $Q(\mathbf{Y}^1, \mathbf{E}^1)$ where $\mathbf{Y}^1 = (1, ..., 1)$ and $\mathbf{E}^1$ is a given amount of energy for the whole inflow (divided among suubflows) to the network. Let B^k be the maximum charging amount subflow k can receive based on its vehicle type. Then, we need to consider all possible feasible values of $\mathbf{E}^i = (E_1^i, ..., E_N^i)$ such that $E_j^i \in [0, B^j]$, $\forall j = 1, .., N$ and $\forall i$. To do so we need to dicretize this range accordingly.

The algorithm works based on the following DP formulation over the subflow-level graph. Similar to (5.1), $Q(\mathbf{Y}^i, \mathbf{E}^i)$ is calculated using the iterative equation

$$Q(\mathbf{Y}^i, \mathbf{E}^i) = \min_{\substack{\mathbf{Y}^j \in O\{\mathbf{Y}^i\},\ 0 \le E_k^j \le B^k \\ \text{s.t } 0 \le E_k^j - E_k^i + e_{y_i,y_j}^k(\mathbf{S}^j) \le B^k \\ k=1,\ldots,N}} \Big[Q(\mathbf{Y}^j, \mathbf{E}^j) + \sum_{k=1}^{N} \tau_{y_i,y_j}^k(\mathbf{S}^j) + \sum_{k=1}^{N} (E_k^j - E_k^i + e_{y_i,y_j}^k(\mathbf{S}^j)) g_k^i + C(\mathbf{Y}^i, \mathbf{Y}^j, \mathbf{S}^j) \Big] \tag{5.6}$$

In (5.6), $Q(\mathbf{Y}^j, \mathbf{E}^j)$ denotes the minimum cost to go from node $\mathbf{Y}^j = (y_1^j, ..., y_N^j)$ with residual energies $\mathbf{E}^j = (E_1^j, ..., E_N^j)$ to the destination node. The one-step cost consists of three parts. The first term, $\sum_{k=1}^{N} \tau_{y_i,y_j}^k(\mathbf{S}^j)$, is the total elapsed time to travel from $\mathbf{Y}^i$ to $\mathbf{Y}^j$ in G_{sf}. The second term, $\sum_{k=1}^{N}(E_k^j - E_k^i + e_{y_i,y_j}^k(\mathbf{S}^j))g_k^i$, shows the total recharging time, and the third term, $C(\mathbf{Y}^i, \mathbf{Y}^j, \mathbf{S}^j)$ is necessary to evaluate the added edge travel times and energy consumption resulting from the specific routing decision. Note that $\tau_{y_i,y_j}^k(\mathbf{S}^j)$ and $e_{y_i,y_j}^k(\mathbf{S}^j)$ are computed based on the corresponding s_l^i (updated based on a decision at node $\mathbf{Y}^i$). Adding the edge $(\mathbf{Y}^i, \mathbf{Y}^j) \in \mathcal{A}_{sf}$, may change the travel times on the arcs previously used in computing $Q(\mathbf{Y}^j, \mathbf{E}^j)$, and it should be modified accordingly. To do so, we add the term $C(\mathbf{Y}^i, \mathbf{Y}^j, \mathbf{S}^j)$:

$$C(\mathbf{Y}^i, \mathbf{Y}^j, \mathbf{S}^j) = \sum_{l \in A_{ij}} (s_l^j)[\tau_l(s_l^i) - \tau_l(s_l^j)] \tag{5.7}$$

where $A_{ij} = \{l : s_l^j > 0 \text{ and } s_l(i,j) > 0\}$ for $l = 1, ..., m$, is a set containing the intersection between edges in the route from node $\mathbf{Y}^j$ to the destination and edges in $(\mathbf{Y}^i, \mathbf{Y}^j) \in A_{sf}$.

Recall that the energy dynamics on the optimal path for each subflow are $E_k^j = E_k^i + r_i^k - e_{y_i,y_j}^k(\mathbf{S}^j)$, $k = 1, ..., N$, the constraint $0 \le E_k^j - E_k^i + e_{y_i,y_j}^k(\mathbf{S}^j) \le B^k$ is the feasibility constraint for amount of charge subflow k may receive at each node, r_i^k.

We seek $\lim_{k\to\infty} Q^{(k)}(\mathbf{Y}^i, \mathbf{E}^i) = Q^*$. In the sequel, we describe the detailed steps of the DP algorithm.

Initialization: Based on our analysis in Chapter 4, we know that if subflow k gets charge on the optimal path, the optimal residual energy at the destination for that subflow is zero. Therefore, assuming all subflows will get charge in their journey, it is obvious that the only option for the cost value at the destination node is $Q^{(0)}(\mathbf{Y}^D, \mathbf{E}^D = \mathbf{0}) = 0$ where D is the index of the destination node in the subflow-level graph, e.g., node 25 in Fig. 5·3. For the other nodes, motivated by Dijkstra's algorithm for the shortest path problem, we set the initial traveling time for all other cases to infinity, i.e.,

$$Q^{(0)}(\mathbf{Y}^D, \mathbf{E}^D) = \begin{cases} 0 & \text{if } \mathbf{E}^D = \mathbf{0}, \\ \infty & \text{if } \mathbf{E}^D > \mathbf{0}. \end{cases} \tag{5.8}$$

$$Q^{(0)}(\mathbf{Y}^i, \mathbf{E}^i) = \infty \quad \forall \mathbf{Y}^i \in \mathcal{N}_{sf} \setminus \mathbf{Y}^D, \quad 0 \le E^i_k \le B^k \tag{5.9}$$

Iteration Steps: The update of Q values can be carried out starting from any node. However, we start it at source node. The Q values are updated as follows: $Q^{(k)}(\mathbf{Y}^D, \mathbf{E}^D) = Q^{(0)}(\mathbf{Y}^D, \mathbf{E}^D)$ and

$$\forall \mathbf{Y}^i \in \mathcal{N}_{sf} \setminus \mathbf{Y}^D, \quad 0 \le E^i_k \le B^k :$$

$$\begin{aligned} Q^{(k)}(\mathbf{Y}^i, \mathbf{E}^i) = \min_{\substack{\mathbf{Y}^j \in O\{\mathbf{Y}^i\},\ 0 \le E^j_k \le B^k \\ \text{s.t } 0 \le E^j_k - E^i_k + e^k_{y_i,y_j}(\mathbf{S}^j) \le B^k \\ k=1,\dots,N}} & \Big[Q^{(k-1)}(\mathbf{Y}^j, \mathbf{E}^j) + \sum_{k=1}^{N} \tau^k_{y_i,y_j}(\mathbf{S}^j) \\ & + \sum_{k=1}^{N} (E^j_k - E^i_k + e^k_{y_i,y_j}(\mathbf{S}^j)) g^i_k + C(\mathbf{Y}^i, \mathbf{Y}^j, \mathbf{S}^j) \Big] \end{aligned} \tag{5.10}$$

The algorithm stops as soon as
$Q^{(k)}(\mathbf{Y}^i, \mathbf{E}^i) = Q^{(k-1)}(\mathbf{Y}^i, \mathbf{E}^i) \quad \forall \mathbf{Y}^i \in \mathcal{N}_{sf}, \quad 0 \le E^i_k \le B^k \quad k = 1, \dots, N.$

5.3.2 Numerical Examples

Consider the 7-node road network in Fig. 5·2 where the distance of each edge is shown. In order to model traffic congestion, the relationship between the speed and density of a vehicle flow is estimated as

$$v(k(t)) = v_f\left(1 - \left(\frac{k(t)}{k_{jam}}\right)^p\right)^q \tag{5.11}$$

where v_f is the reference speed on the road without traffic, $k(t)$ represents the density of vehicles on the road at time t and k_{jam} denotes the saturated density for a traffic jam. The parameters p and q are empirically identified for actual traffic flows. we assume the energy consumption rates of subflows on arc $(i,j) \in \mathcal{A}$ are all identical, proportional to the distance between nodes i and j in the road network, giving $e^k_{y_i,y_j} = e \cdot d_l \cdot \dfrac{R}{N}$, where (y^i_k, y^j_k) corresponds to the lth edge in $\mathcal{A}$.

For simplicity we divide the total inflow R into N identical subflows, each of which has R/N vehicles per unit of time. Fig. 5·3 shows the subflow-level graph for this example for $N = 2$. Now in the subflow-level graph, the time subflow k spends on arc (y^i_k, y^j_k) becomes:

$$\tau^k_{y_i,y_j} = \big(d_l \cdot \frac{R}{N}\big)\big(v_f(1 - (\frac{s^i_l}{N})^p)^q\big)^{-1}$$

s^i_l determines the number of subflows (density) through this edge starting from node $\mathbf{Y}^i$ to the destination node $\mathbf{Y}^D$.

In order to examine the efficiency of the DP algorithm, we solve the problem for the network with homogeneous charging nodes with $g_i = 1 \ \forall i \in \mathcal{N}$ for different value of N. Table. 5.2 compares the solution and CPU times (computational effort) for different values of number of subflows. It is obvious from our results that as number of subflows, N, increases, DP loses its efficiency and will be computationally more expensive than MINLP. On the other hand, our analysis and numerical examples in

Chapter 4 show that our proposed *flow control formulation* for the same problem results in a reduction of about 4 orders of magnitude in CPU time with near optimal solution. We have also shown that using the NLP solution and its objective value as a lower bound for the optimal objective value, we can find a good value for the number of subflows, N, which render the problem computationally manageable.

Table 5.2: Numerical results for sample problem

	MINLP	**DP**
N	2	2
obj	116.67	116.67
routes	$1 \rightarrow 2 \rightarrow 3 \rightarrow 7$ $1 \rightarrow 4 \rightarrow 7$	$1 \rightarrow 2 \rightarrow 3 \rightarrow 7$ $1 \rightarrow 4 \rightarrow 7$
CPU time (sec)	1674.2	79.17
N	3	3
obj	99.68	99.68
routes	$1 \rightarrow 2 \rightarrow 3 \rightarrow 7$ $1 \rightarrow 4 \rightarrow 7$ $1 \rightarrow 5 \rightarrow 6 \rightarrow 7$	$1 \rightarrow 2 \rightarrow 3 \rightarrow 7$ $1 \rightarrow 4 \rightarrow 7$ $1 \rightarrow 5 \rightarrow 6 \rightarrow 7$
CPU time (sec)	1752.5	5534.6
N	6	6
obj	99.68	NA
routes	$1 \rightarrow 2 \rightarrow 3 \rightarrow 7(\times 2)$ $1 \rightarrow 4 \rightarrow 7(\times 2)$ $1 \rightarrow 5 \rightarrow 6 \rightarrow 7(\times 2)$	NA
CPU time (sec)	2579	NA

5.4 Conclusions and future work

We have studied the problem of minimizing the total elapsed time for energy-constrained vehicles to reach their destinations, including recharging when there is no adequate energy for the entire journey. In this chapter, we have formulated both user-centric and system-centric problems as DP problems. For a single vehicle problem (user-centric), this approach is very efficient and determines an optimal solution in seconds. For a multi-vehicle problem (system-centric), where traffic congestion effects are considered, we used a similar approach by aggregating vehicles into subflows and seeking optimal routing decisions for each such subflow. In this case, our DP algorithm works well for a small number of subflows but as the number of subflows increases, it loses its efficiency.

Chapter 6

The Price of Anarchy in Transportation Networks Using Actual Traffic Data

6.1 Introduction

A transportation (traffic) network is a system with non-cooperative agents (drivers) in which each driver seeks to minimize her own cost by choosing the best route (resources) to reach her destination without taking into account the overall system performance. In these systems, the cost for each agent depends on the resources it chooses as well as the number of agents choosing the same resources (Wang et al., 2015). In such a non-cooperative setting, one often observes convergence to a Nash equilibrium, a point where no agent can benefit by altering its actions assuming that the actions of all the other agents remain fixed (Youn and Jeong, 2008). However, it is known that the Nash equilibrium is not always the best strategy from the system's point of view and results in a suboptimal behavior compared to the *socially optimal* policy. In a transportation network with selfish drivers, each agent (driver) follows the path (we will use "path" and "route" interchangeably) derived from a *user optimal* policy. In order to quantify the social suboptimality of selfish driving, we use the Price of Anarchy (POA) as a measure to compare system performance under a user-optimal policy vs. a system-optimal policy.

The equilibrium flow in traffic networks, known as "Wardrop equilibrium," is the solution of the *Traffic Assignment Problem (TAP)* (Dafermos and Sparrow, 1969), (Patriksson, 2015). In the transportation science literature, the TAP, which will be termed "forward problem" in what follows, has been extensively explored; see, e.g., (Patriksson, 2015) and the references therein. To solve the TAP, we need to know *a priori* the specific cost function, as well as the Origin-Destination (O-D) demand matrix.

Recent developments in data-driven inverse optimization techniques (Bertsimas et al., 2014) enable the estimation of the cost (usually, the travel time) functions given the observations of the equilibrium flows from a large-scale transportation network. This facilitates a better understanding of the underlying dynamics of the transportation system itself. In addition, with cost function estimates at our disposal, we can address the issue of improving a traffic network's performance by controlling traffic flows, hence, contributing to the design of better transportation systems that serve Smart Cities.

In this chapter, we leverage actual traffic data provided to us by the Boston Region Metropolitan Planning Organization (MPO). Applying a traffic flow model, we first infer equilibrium flows on each segment from the spatial average speed data. Then, by adopting the estimated traffic flows we obtain O-D demand matrices. Finally, we formulate a system-centric problem in which agents, here drivers, cooperate to optimize the overall system performance. This allows us to estimate the POA for a sub-network so as to determine the difference in network performance between selfish routing (non-cooperative) and system-optimal routing (cooperative).

The rest of this chapter is organized as follows. In Sec. 6.2 we present the models and methods we apply to the traffic data. In Sec. 6.3 we provide descriptions of the datasets we use. We elaborate on data processing tasks in Sec. 6.4. We quantify the POA for the transportation network in Sec. 6.5, where numerical results for a subnetwork are included to illustrate our approaches.

6.2 Models and Methods

In this section we describe the network model and provide the formulations for optimization problems obtaining the Nash flows and social optimum flows.

6.2.1 Transportation network model

Consider a directed graph, denoted by $(\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{A}$ the set of links. Assume it is strongly connected. Let $\mathbf{N} \in \{0, 1, -1\}^{|\mathcal{V}| \times |\mathcal{A}|}$ be the node-link incidence matrix, and $\mathbf{e}_a$ the vector with an entry equal to 1 corresponding to link a and all the other entries equal to 0.

Let $\mathbf{w} = (w_s, w_t)$ denote an origin-destination (O-D) pair and

$$\mathcal{W} = \{\mathbf{w}_i : \mathbf{w}_i = (w_{si}, w_{ti}), \, i = 1, \ldots, |\mathcal{W}|\}$$

the set of all O-D pairs. Denote by $d^{\mathbf{w}} \geq 0$ the amount of the flow demand from w_s to w_t. Let $\mathbf{d}^{\mathbf{w}} \in \mathbb{R}^{|\mathcal{V}|}$ be the vector which is all zeros, except for a $-d^{\mathbf{w}}$ in the coordinate corresponding to node w_s and a $d^{\mathbf{w}}$ in the coordinate corresponding to node w_t.

Let $\mathcal{R}_i$ be the index set of simple routes (a *simple route* is a route without cycles) connecting O-D pair i. For all $a \in \mathcal{A}$, $r \in \mathcal{R}_i$, and $i \in \{1, \ldots, |\mathcal{W}|\}$, define the link-route incidence by

$$\delta_{ra}^{i} = \begin{cases} 1, & \text{if route } r \in \mathcal{R}_i \text{ uses link } a, \\ 0, & \text{otherwise.} \end{cases}$$

Let x_a be the total link flow on link $a \in \mathcal{A}$ and $\mathbf{x}$ the vector of these flows. Let $t_a(\mathbf{x}) : \mathbb{R}_+^{|\mathcal{A}|} \to \mathbb{R}_+$ be the cost function for link $a \in \mathcal{A}$; in particular, when $t_a(\mathbf{x})$ only depends on x_a, we also write $t_a(\mathbf{x})$ as $t_a(x_a)$. In addition, denote by $\mathbf{t}(\mathbf{x})$ the vector-valued function whose ath component is $t_a(\mathbf{x})$.

Let us assume that the cost functions have the following form (Bertsimas et al., 2014),(Branston, 1976):

$$t_a\left(x_a\right) = t_a^0 g\left(\frac{x_a}{m_a}\right), \tag{6.1}$$

where t_a^0 is the free-flow travel time of $a \in \mathcal{A}$, $g(\cdot)$ is strictly increasing and continuously differentiable on $\mathbb{R}_+$, and m_a is the effective flow capacity of $a \in \mathcal{A}$.

Let $\mathcal{F}$ be the set of feasible flow vectors defined by

$$\mathcal{F} = \left\{\mathbf{x} : \exists \mathbf{x}^{\mathbf{w}} \in \mathbb{R}_+^{|\mathcal{A}|} \text{ s.t. } \mathbf{x} = \sum_{\mathbf{w}\in\mathcal{W}} \mathbf{x}^{\mathbf{w}},\ \mathbf{N}\mathbf{x}^{\mathbf{w}} = \mathbf{d}^{\mathbf{w}},\ \forall \mathbf{w} \in \mathcal{W}\right\},$$

where $\mathbf{x}^{\mathbf{w}}$ is the flow vector attributed to O-D pair $\mathbf{w}$.

6.2.2 Selfish routing

As it is discussed, a transportation network is a system with non-cooperative agents (drivers) competing to optimize their own costs (traveling time) by utilizing resources (link capacity). In such setting when every agent acts selfishly, one often observes convergence to a Nash equilibrium. In transportation networks the equilibrium flow is known as “Wardrop equilibrium”.

Wardrop equilibrium. A feasible flow $\mathbf{x}^* \in \mathcal{F}$ is a *Wardrop equilibrium* if for every O-D pair $\mathbf{w} \in \mathcal{W}$, and any route connecting (w_s, w_t) with positive flow in $\mathbf{x}^*$, the cost of traveling along that route is less than or equal to the cost of traveling along any other route that connects (w_s, w_t). Here, the cost of traveling along a route is the sum of the costs of each of its constituent links (Bertsimas et al., 2014).

It is a well-known fact that the Wardrop equilibrium is the solution of the Traffic Assignment Problem (TAP). TAP can be formulated as the following optimization problem (Dafermos and Sparrow, 1969), (Patriksson, 2015):

$$\min_{\mathbf{x}\in\mathcal{F}} \sum_{a\in\mathcal{A}} \int_0^{x_a} t_a\left(s\right) ds. \tag{6.2}$$

6.2.3 Socially optimal routing

Assuming a network with multiple O-D pairs, the total latency of the network is defined as follows:

$$L(\mathbf{x}) = \sum_{a \in \mathcal{A}} x_a t_a(x_a). \tag{6.3}$$

In order to find socially optimum flows we formulate the following optimization problem (Patriksson, 2015) (Pourazarm and Cassandras, 2016):

$$\min_{\mathbf{x} \in \mathcal{F}} \sum_{a \in \mathcal{A}} x_a t_a(x_a). \tag{6.4}$$

The problem above is a Non-Linear Programming (NLP) problem in which the nonlinearity comes from the cost function $t_a(x_a)$. The solution of this problem, x_a^*, obtains the optimal flow on each link minimizing the total latency in the network.

6.2.4 Price of Anarchy

The POA is a measure to quantify the system performance under user-optimal policy vs. system-optimal policy. In a traffic network it is defined as the ratio between the total latency, i.e., the total travel time over all drivers in different O-D pairs, obtained under Wardrop flows and that obtained under social-optimal flows. Let now $\mathbf{x}^*$ and $\mathbf{x}^{ne}$ denote the socially optimum and the Wardrop link flow vectors respectively. Then, the POA is defined as

$$\text{POA} = \frac{L(\mathbf{x}^{ne})}{L(\mathbf{x}^*)}. \tag{6.5}$$

6.3 Data Set Description

6.3.1 Speed dataset description

The actual traffic data provided by the MPO is a dataset of 51.2 GB consisting of 861 CSV files, each with more than 1 million lines of data. The dataset includes the spatial average speeds for major roadways and arterial streets in Eastern Massachusetts for the year 2012. The average speed within a given unit of spatial reference is calculated by aggregating observed speeds from billions of data points. Specifically, it is derived by combining data from physical traffic sensors (e.g., induction loop sensors, toll tag readers, etc), as well as all available data from probe vehicles (equipped with on-board GPS devices returning speed and location back to a central system) that fall within a specific segment of a road for a particular time window.

The dataset includes traffic data for more than 13,000 road segments (with the average distance of 0.7 miles; see Fig. 6·1) of Eastern Massachusetts, covering the average speed for every minute of the year 2012.

For each road segment, identified with a unique *tmc (traffic message channel)* code, the dataset provides information such as speed data (instantaneous, average and free-flow speed) in mph, date and time, and traveling time (minute) through that segment. Table 6·2 shows a few sample lines of the speed dataset. Note that a road typically consists of many segments.

6.3.2 Capacity dataset description

The flow capacity (vehicles/hour) dataset, provided by the MPO, includes capacity data – vehicle counts for each road segment – for more than 100,000 road segments (average distance of 0.13 miles) in Eastern Massachusetts. In particular, the capacity data is given for four different time periods (AM: 6 am – 9 am, MD: 9 am – 3 pm, PM: 3 pm – 6 pm, and NT: 6 pm – 6 am) in a day. For each time period, the total

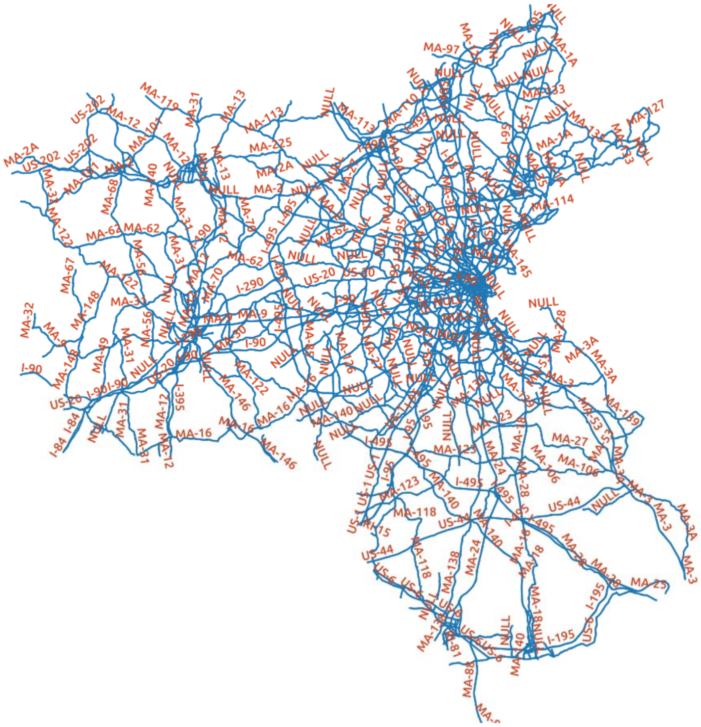
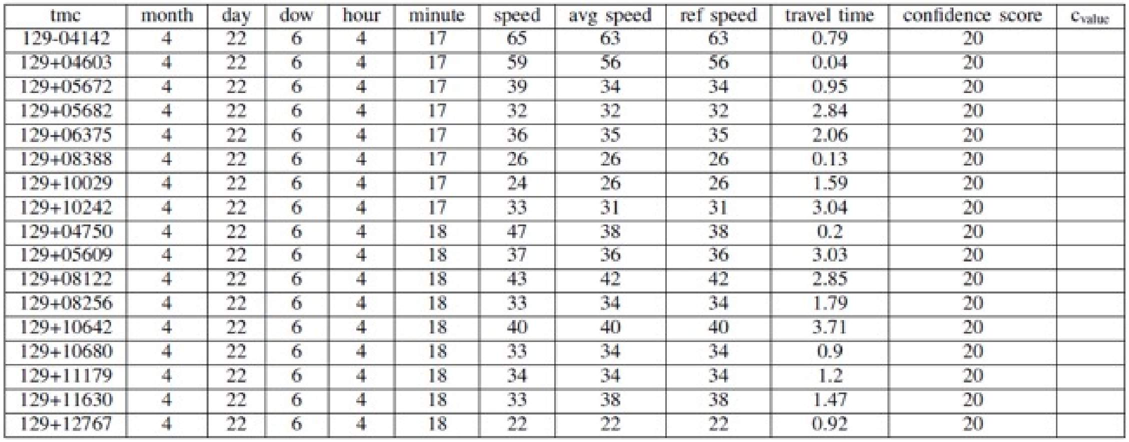


Figure 6·1: All available road segments in the road map of Eastern Massachusetts.



tmc	month	day	dow	hour	minute	speed	avg speed	ref speed	travel time	confidence score	c_{value}
129-04142	4	22	6	4	17	65	63	63	0.79	20	
129+04603	4	22	6	4	17	59	56	56	0.04	20	
129+05672	4	22	6	4	17	39	34	34	0.95	20	
129+05682	4	22	6	4	17	32	32	32	2.84	20	
129+06375	4	22	6	4	17	36	35	35	2.06	20	
129+08388	4	22	6	4	17	26	26	26	0.13	20	
129+10029	4	22	6	4	17	24	26	26	1.59	20	
129+10242	4	22	6	4	17	33	31	31	3.04	20	
129+04750	4	22	6	4	18	47	38	38	0.2	20	
129+05609	4	22	6	4	18	37	36	36	3.03	20	
129+08122	4	22	6	4	18	43	42	42	2.85	20	
129+08256	4	22	6	4	18	33	34	34	1.79	20	
129+10642	4	22	6	4	18	40	40	40	3.71	20	
129+10680	4	22	6	4	18	33	34	34	0.9	20	
129+11179	4	22	6	4	18	34	34	34	1.2	20	
129+11630	4	22	6	4	18	33	38	38	1.47	20	
129+12767	4	22	6	4	18	22	22	22	0.92	20	

Figure 6·2: Sample lines from speed dataset

roadway capacity for all available lanes for that time period is given. These values are calculated based on the share of daily traffic counts in each hour of that time period. For each time period there exists a period capacity factor applied to represent peak hour conditions within that period. These factors are as follows: 2.5 for AM, 4.75 for MD, 2.5 for PM, and 7 for NT. Then, the total roadway capacity for a time period is the product of the capacity/lane/hour, the number of lanes, and the capacity factor. In our experiments, we need flow capacity on each segment in vehicle counts per hour. Thus, for each time period we scale the given vehicle counts by the inverse of the corresponding capacity factor.

6.3.3 Matching capacity data with speed data

Note that, in the capacity dataset, the ID for a road segment is named *road inventory ID*, and the segments are not absolutely identical with those in the speed dataset. Based on the geographic longitude and latitude, we have built up a dictionary mapping segments with *tmc* code to capacity dataset *road inventory ID*, through which we can read the capacity data for each road segment in the speed dataset.

6.4 Data Processing

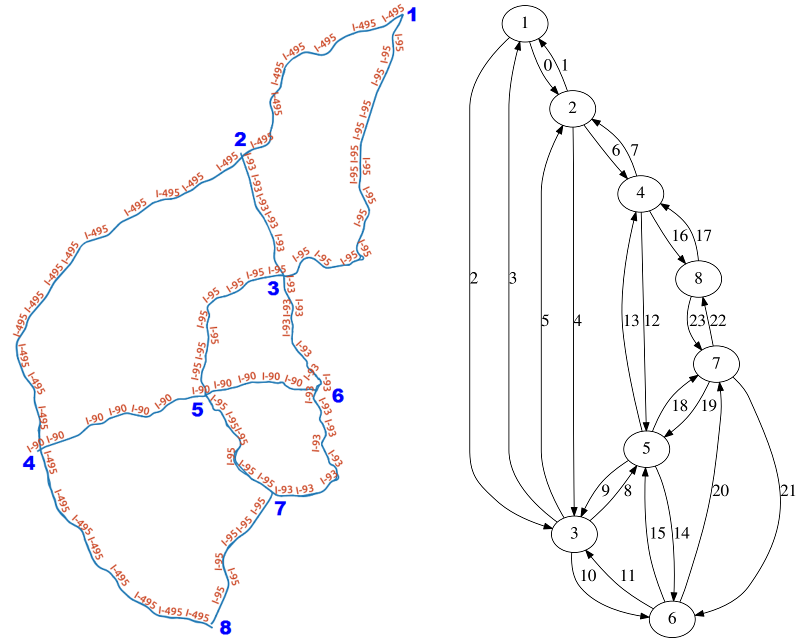
6.4.1 Preprocessing

Calculating average speed and free-flow speed

First, we select the time instances set $\mathcal{T}$ consisting of each minute of AM (7 am – 9 am), MD (11 am – 1 pm), PM (5 pm – 7 pm), and NT (9 pm – 11 pm) for each day of January, April, July, and October, all in 2012. Note that the selected AM (resp., MD, PM, NT) period is a subinterval of the AM (resp., MD, PM, NT) period in the capacity dataset. Then, we calculate the average speed for each segment separately for the four time periods, each of which lasts 120 minutes. Finally, for each segment, we compute a reliable proxy of the free-flow speed by using the 85th-percentile point of the observed speeds on that segment for all the time instances belonging to $\mathcal{T}$.

Selecting a sub-network

To reduce the computational burden while capturing the key elements of the Eastern Massachusetts road network, we only consider a representative interstate highway sub-network as shown in Fig. 6·3(a), where there are 701 road segments, composing a road network with 8 nodes and 24 links. We depict the topology of this sub-network in Fig. 6·3(b).



(a) (b)

Figure 6·3: (a) An interstate highway sub-network of Eastern Massachusetts (a sub-map of Fig. 6·1; the blue numbers indicate node indices); (b) The topology of the sub-network (the numbers beside arrows are link indices, and the numbers inside ellipses are node indices).

Inferring flow data from the speed dataset

In order to infer flow data from the speed data for each link, we use the a macroscopic traffic flow model. Macroscopic models represent how the behavior of one parameter of traffic flow (density, speed and flow) changes with respect to another. In particular we use the Greenshield's traffic model (Greenshields et al., 1935) which assumes a linear speed-density relationship as illustrated in Fig. 6·4(a). The equation for this relationship is as below:

$$v = v_f(\frac{v_f}{k_j}).k \tag{6.6}$$

where v is the mean speed at density k, v_f is the free speed and k_j is the jam density. Equation (6.6) is often referred to as the Greenshields model. It indicates that when density becomes zero, speed approaches free flow speed (i.e., $v \to v_f$ when $k \to 0$).

The relation between flow and density is parabolic in shape as shown in figure 6·4(b) and can be derived as follows: It is known that $q = k.v$, where q is the vehicle flow. Now substituting v with equation (6.6) we get

$$q = v_f.k - (\frac{v_f}{k_j}).k^2 \tag{6.7}$$

To find density at maximum flow, i.e., k_{max} at q_{max} in figure 6·4(b), differentiate equation (6.7) with respect to k and equate it to zero. i.e.,

$$\frac{dq}{dk} = v_f - \frac{v_f}{k_j}2k = 0 \to k_{max} = k_j/2 \tag{6.8}$$

Therefore, density corresponding to maximum flow is half the jam density. Substituting (6.8) in (6.7), one can show that $q_{max} = \frac{v_f.k_j}{4}$. This means that exceeding k_{max}, results in congestion and flow reduction. In other words, when the density is in the range $0 \leq k \leq k_{max}$, the road is uncongested and the flow monotonically increases with k, while for $k_{max} < k \leq k_{jam}$, the road is congested and consequently the flow decreases with k and eventually $q \to 0$ when $k \to k_{jam}$. Thus, as shown in figure 6·4(b), each value of q corresponds to 2 different values of density., k_1 for the uncongested case and k_2 for congested case.

Similarly we can find the relation between speed and flow. For this, put $k = q/v$

in equation (6.6). We get,

$$q = k_j.v - \frac{k_j}{v_f} v^2 \tag{6.9}$$

This relationship is again parabolic and is shown in figure 6·4(c).

Aggregating flows of the segments on each link

Let $\{v_a^j, t_a^j, v_a^{0j}, t_a^{0j}, m_a^j;\ j = 1, \ldots, J_a\}$ denote the available observations (v_a^j, t_a^j), and parameters (v_a^{0j}, t_a^{0j}, m_a^j) of the segments composing link $a \in \mathcal{A}$, where, for each segment j, v_a^j (resp., v_a^{0j}) is the speed (resp., free-flow speed; miles/hour), t_a^j (resp., t_a^{0j}) is the travel time (resp., free-flow travel time; hour), and m_a^j is the flow capacity, equivalent to q_{max} in Fig. 6·4(b), (vehicles/hour). Using the Greenshield's model, we calculate the flow on segment j by

$$x_a^j = \frac{4m_a^j}{v_a^{0j}} v_a^j - \frac{4m_a^j}{(v_a^{0j})^2} (v_a^j)^2. \tag{6.10}$$

In our analysis, we enforce $v_a^j \leq v_a^{0j}$ to make sure that the flow given by (6.10) is nonnegative. In particular, if for some time instance $v_a^j > v_a^{0j}$ (this rarely happens), we set $v_a^j = v_a^{0j}$ in (6.10), thus leading to a zero flow estimation for this time instance.

Aggregating over all segments composing link a we compute:

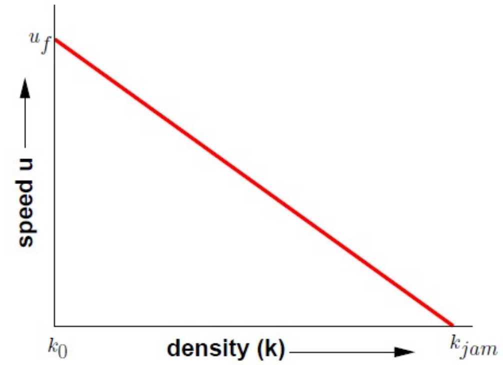
$$x_a = \frac{\sum_{j=1}^{J_a} x_a^j t_a^j}{\sum_{j=1}^{J_a} t_a^j}, \quad t_a^0 = \sum_{j=1}^{J_a} t_a^{0j}, \quad m_a = \frac{\sum_{j=1}^{J_a} m_a^j t_a^{0j}}{\sum_{j=1}^{J_a} t_a^{0j}},$$

where x_a^j is given by (6.10) and $t_a^{0j} = v_a^j t_a^j / v_a^{0j}$, $j = 1, \ldots, J_a$.

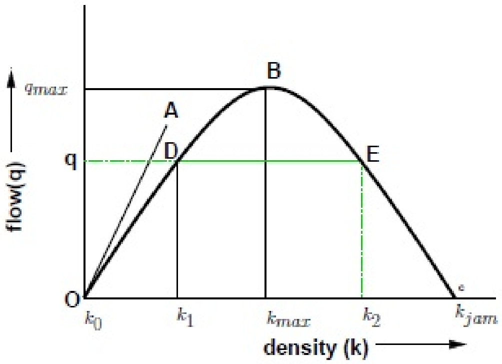
Processing flow data such that the flow conservation law is satisfied

For $a \in \mathcal{A}$, let $\hat{x}_a$ denote the original estimate of the flow on link a, and x_a its adjustment. Solve the following *Least Squares Problem (LSP)*:

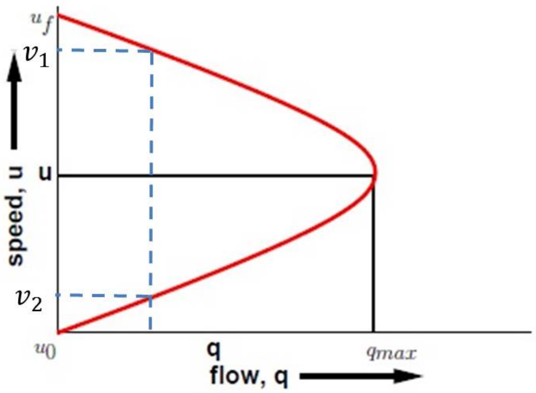
$$\min_{\mathbf{x}} \quad \sum_{a \in \mathcal{A}} (x_a - \hat{x}_a)^2 \tag{6.11}$$



(a) Speed vs. Density



(b) Flow vs. Density



(c) Speed vs. Flow

Figure 6·4: Relationships between speed, density, and flow based on Greenshield's traffic flow model

$$\text{s.t.} \quad \sum_{a\in I(i)} x_a = \sum_{a\in O(i)} x_a, \qquad \forall i \in \mathcal{V},$$

$$x_a \geq 0, \qquad \forall a \in \mathcal{A},$$

where the 1st constraint enforces flow conservation for each node $i \in \mathcal{V}$, where $I(i)$ (resp., $O(i)$) denotes the set of links entering (resp., outgoing) to (resp., from) node i.

6.4.2 Estimating the O-D demand matrix

Note that we need to know the O-D demand information (compiled into a matrix) in both the forward problem formulation (6.2) and the socially optimum formulation (6.4). Based on the parameters and flows of the road network, we borrow the General Least Squares (GLS) method (Hazelton, 2000) to estimate the desired O-D demand matrix, using the following steps:

Obtaining link-route incidence matrix

We assume that each node could be an origin and a destination; for the subnetwork shown in Fig. 6·3(a), there are $8 \times (8-1) = 56$ O-D pairs in total. We then identify feasible routes for each O-D pair, thereby obtaining a 24×314 link-route incidence matrix. (recall the definition of link-route incidence in Sec. 6.2; 314 routes identified in total).

Implementing GLS method

Vectorize the O-D demand matrix as λ. Let $\mathbf{A}$ be the link-route incidence matrix and $\mathbf{P} = [p_{ir}]$ the route choice probability matrix, where p_{ir} is the probability that a traveler between O-D pair i uses route r. Let $\{\mathbf{x}^{(k)}; k = 1, \ldots, K\}$ denote K observations of the flow vector and $\bar{\mathbf{x}}$ the average. Then, the O-D demand matrix estimation problem is equivalent to the following generalized least squares problem (Hazelton, 2000):

$$\begin{aligned}
&\max_{\mathbf{P}\succeq\mathbf{0},\,\lambda\succeq\mathbf{0}} \quad -\frac{1}{2}\sum_{k=1}^{K}\left(\mathbf{x}^{(k)}-\mathbf{A}\mathbf{P}'\boldsymbol{\lambda}\right)'\mathbf{S}^{-1}\left(\mathbf{x}^{(k)}-\mathbf{A}\mathbf{P}'\boldsymbol{\lambda}\right),\\
&\text{s.t.}\quad p_{ir}=0 \quad \forall (i,r)\in\{(i,r): r\notin\mathcal{R}_i\},\\
&\mathbf{P1}=\mathbf{1},
\end{aligned}$$

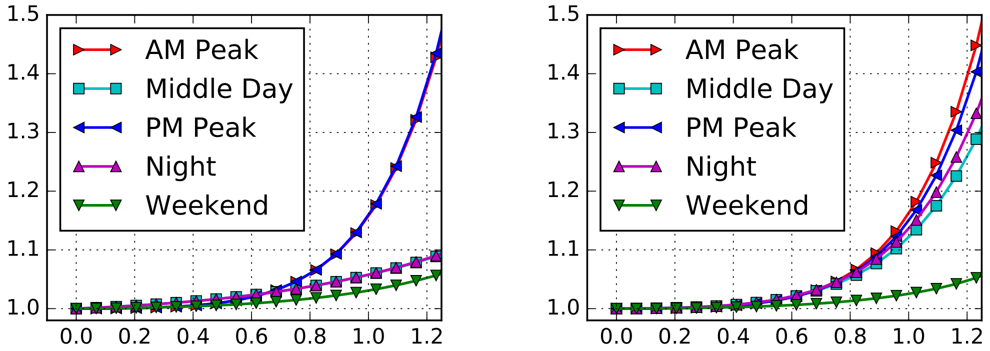
where $\mathbf{S}=(1/(K-1))\sum_{k=1}^{K}\left(\mathbf{x}^{(k)}-\bar{\mathbf{x}}\right)\left(\mathbf{x}^{(k)}-\bar{\mathbf{x}}\right)'$ is the sample covariance matrix. For more details about the implementation of GLS method, please refer to (Zhang et al., 2016). Solving This problem obtains the O-D demand vector minimizing the distance between the resulting flow vector and different flow observations for a specific scenario. Using this formulation, we can estimate the O-D demand vector on a monthly basis or more accurately on a daily basis based on our choice of observations, $\mathbf{x}^{(k)}$s. As an example, if we want to estimate the average O-D demand vector for working days in April - PM period on a monthly basis, our choices for flow vector observations $\mathbf{x}^{(k)},\ \ k=1,....,K$ are among 21×120 different flow vectors corresponding to 21 different working days and 120 different flow vectors according to 120 minutes between 5 PM and 7 PM.

6.4.3 Estimating cost functions

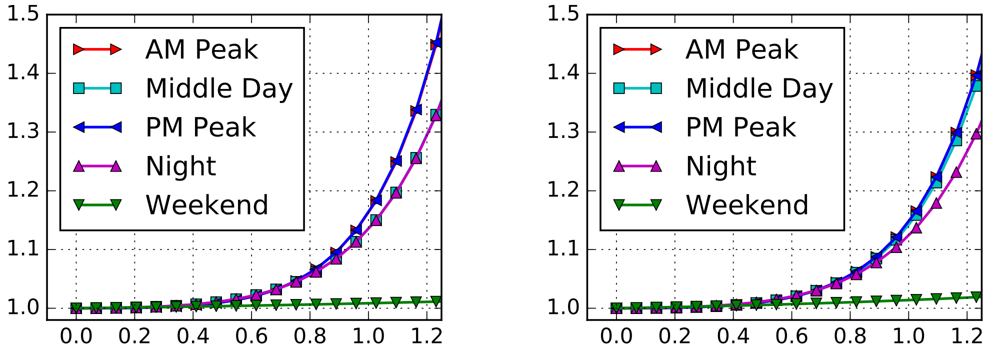
Here, we briefly explain how the cost function (6.1) can be derived from actual traffic data. The whole process is beyond the scope of this thesis and we borrow it from (Zhang et al., 2016). Using the estimated flow data and the O-D demand matrices, (Zhang et al., 2016) estimates the cost functions for 20 different scenarios. To do so, (Zhang et al., 2016) formulates appropriate inverse Variational Inequality (VI) problems to recover the per-road cost (congestion) functions determining user route selection for each month and time-of-day period. Applying polynomial kernels in the corresponding Quadratic Programming (QP) problem, (Zhang et al., 2016) estimates cost functions $g(.)$ in (6.1) as polynomial functions. It estimates the cost functions for different scenarios: AM (7 am – 9 am), MD (11 am – 1 pm), PM (5 pm – 7 pm), and NT (9pm – 11 pm) for each day of January, April, July, and October, all in 2012. The estimated $g(.)$ functions corresponding to five different time periods for mentioned months are shown in Fig. 6·5 We observe that the costs for the AM/PM peaks are much more sensitive to traffic flows than for the other three time periods (MD, NT, and weekend). This can be explained by taking into account the traffic condition during a day: from a congested road network in the AM/PM period to an uncongested road network during MD, NT, or weekend periods.

6.5 Numerical Results

In this section, we investigate the POA for the network shown in Fig. 6·3(a). First, we consider a specific time period in a day during the year 2012 and use the corresponding cost function estimated from data. As discussed in Section 6.4.1, adopting Greenshield's traffic flow model, the Wardrop flow on each link has been calculated using speed and capacity datasets for each minute of a time period. Thus, in order to quantify the POA we calculate socially optimal flows by solving problem (6.4). As an example we calculate the POA for the PM (5 pm – 7 pm) period of Wednesday, Oct. 10, 2012. The corresponding O-D demand matrix includes 42 *active* O-D pairs with nonzero flow demands. Fig. 6·6 shows the socially optimum vs. average user-optimum flows on each link on Oct. 10 during the PM period. We can observe that for some links (e.g., links 5 and 11), there exist significant differences in the link flow values between selfish behavior and system-centric behavior suggesting several potential opportunities to improve the system performance. We then look at the POA

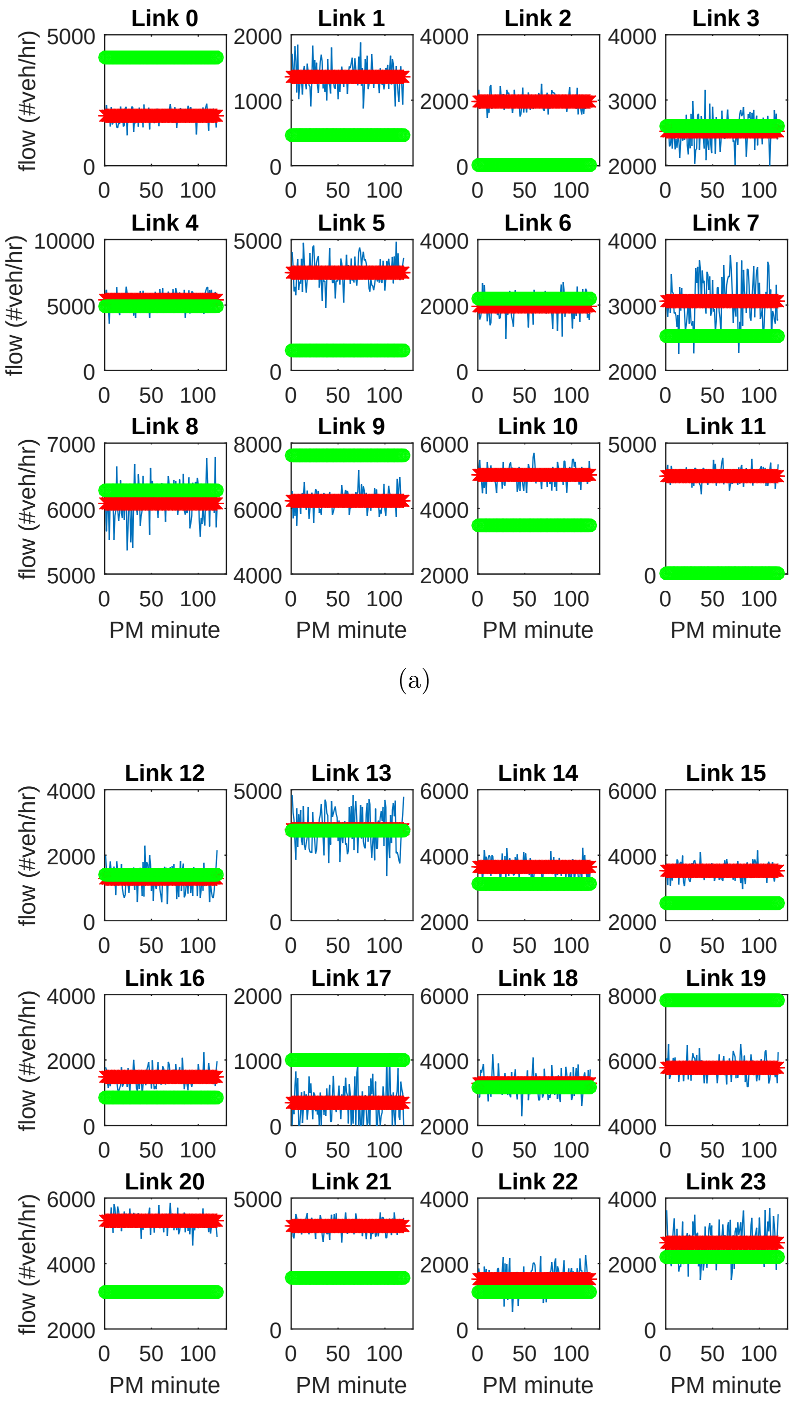


(a) Jan. (b) Apr.



(c) Jul. (d) Oct.

Figure 6·5: Comparison of the estimated cost functions corresponding to different time periods.



(a)

(b)

Figure 6·6: (a) Social-opt. (green) and user-opt. (red) flows on links 0 to 11; (b) Social-opt. (green) and user-opt. (red) flows on links 12 to 23.

for a specific time period in a whole month. Fig. 6·7 shows the POA for the PM period during April 2012. It is observed that POA > 1 for all days in April during the PM period. In the worst case, on April 12 and April 22, POA $\simeq 2$, which means that the system is considerably inefficient under selfish driving. On the other hand, POA $= 1.23$ in the best case showing that we can reduce the total latency in the network by at least 23% if drivers follow socially optimal paths.

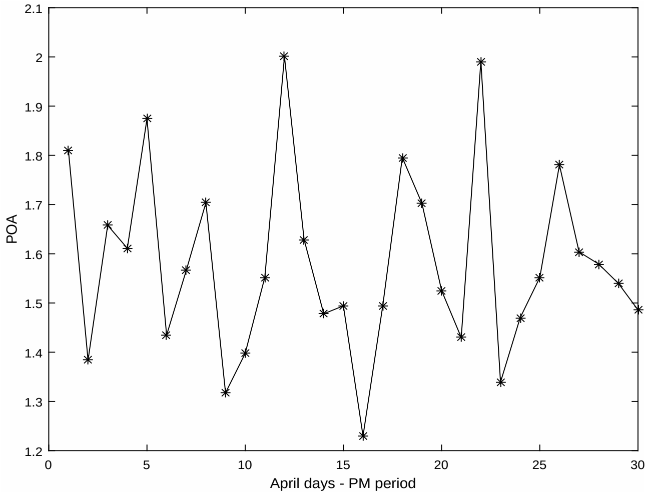


Figure 6·7: POA for PM period (5–7 pm) in April based on avg. flow on each link.

As an alternative way of quantifying the POA, we also assume a scenario in which all drivers use socially optimum routing and a single driver acts selfishly. In this scenario, we assume the user has prior knowledge about the total delay on each path of the desired O-D pair. The selfish driver deviates from the social optimum flow by traveling through the *shortest path*, thus, enjoying less traveling time. In our example, under the social optimum policy, the latency a driver experiences over each link is shown in Tab. 6.1. Now, a driver joining the flow of O-D pair 1-6, can choose one of the four paths from node 1 to node 6 (refer to Fig. 6·3(b)): $\text{Path}_1 : 1 \to 3 \to 6$; $\text{Path}_2 : 1 \to 2 \to 3 \to 6$; $\text{Path}_3 : 1 \to 3 \to 5 \to 6$ and $\text{Path}_4 : 1 \to 2 \to 3 \to 5 \to 6$ with traveling times 0.654, 0.651, 0.872, and 0.869 hr respectively. If the driver acts selfishly and joins the shortest path, path$_2$, instead of her assigned path, path$_4$, she will gain 25.08% in traveling time.

Table 6.1: Traveling time (hr) on each link under system-optimal flows.

edge No.	0	1	2	3	4	5
t_a^*	0.310	0.329	0.455	0.465	0.143	0.169
edge No.	6	7	8	9	10	11
t_a^*	0.533	0.551	0.242	0.252	0.199	0.205
edge No.	12	13	14	15	16	17
t_a^*	0.239	0.247	0.175	0.173	0.343	0.344
edge No.	18	19	20	21	22	23
t_a^*	0.19	0.19	0.26	0.26	0.194	0.201

6.6 Summary

In this chapter, we study a large-scale transportation network (Eastern Massachusetts) using actual traffic data obtained from the Boston Region Metropolitan Planning Organization (MPO) for the year 2012. Estimating equilibrium flows and O-D demand matrices, we quantify the Price of Anarchy (POA) for a interstate highway subnetwork for Eastern Massachusetts. Our findings measure the price the society is paying due to non-cooperative behavior of its members and could provide useful suggestions to the efforts of building a smarter city.

Chapter 7

Conclusions and Future Directions

In this dissertation we focused on applying several control and optimization methodologies to different classes of energy-aware battery-powered systems.

First, we revisited the lifetime maximization problem for wireless sensor networks with fixed topology and incorporated non-ideal battery dynamics for nodes in order to take into account non-ideal phenomena in batteries, i.e., the rate capacity effect and the recovery effect. For the static networks the network lifetime is defined as the earliest time that the first node runs out of energy. We started our analysis by adopting Kinetic Battery Model (KBM) and generalized our results by utilizing a more elaborate battery model, the diffusion based model, of which KBM is a special case. Formulating the problem in the optimal control framework, we have shown that there exist a time-invariant optimal routing policy which maximizes the network lifetime, even with non-linear battery dynamics. The computational complexity of the original OCP problem has been reduced by showing that the associated fixed routing probabilities can be obtained by solving a set of relatively simple Non-Linear Programming (NLP) problems. In addition, under very mild conditions, this optimal policy is independent of the battery parameter model. This robustness property leads to the fact that, one can resort to the case of ideal batteries where the optimal routing problem is much simpler to solve and can be reduced to a Linear Programming (LP) problem. Then, considering a non-ideal, more realistic battery model, and applying the optimal routing policy, we reach a more precise estimate for the network lifetime.

We have also considered a joint routing and initial energy allocation problem over the network nodes with the same network lifetime maximization objective. In this case, the solution to this problem is given by a policy that depletes all node energies at the same time and the corresponding energy allocation and routing probabilities are obtained by solving a NLP problem. We have also investigated the security benefits of probabilistic routing compared to other routing policies under an energy-depletion attack.

Second, we have considered the case when the source node is mobile while the relay nodes are static. We assumed that any such mobile node travels along a trajectory that it determines and which may or may not be known in advance. While on its trajectory, the source node continuously performs sensing tasks and generates data. Adding mobility to the source node, we have redefined the lifetime to be the time until the source node depletes its energy. When the mobile node's trajectory is unknown in advance, we have introduced three versions of an optimal control problem aiming at this lifetime maximization. We have shown that in all cases, the solution can be reduced to a sequence of Non-Linear Programming (NLP) problems solved on line as the source node trajectory evolves. For the more challenging (from a computational perspective) problem, where the source node's trajectory is known in advance, this information can be incorporated into an optimal lifetime maximization policy. We have formulated the problem as an optimal control problem and solved the problem using a numerical solver. Based on our numerical results, It has been observed that the prior knowledge of the source node's motion dynamics considerably increases the network lifetime.

Third, motivated by the significant role of recharging in battery-powered vehicles, we have studied the problem of minimizing the total elapsed time for energy-constrained vehicles to reach their destinations, including recharging when there is no adequate energy for the entire journey. We have investigated two versions of this problem: user-optimal scenario vs. system-optimal scenario. For the user-centric problem, referred as the single-vehicle routing problem, the problem has been formulated as a Mixed-Integer Non-Linear Programming (MINLP) optimization problem which is the exact formulation. Adopting some properties of the optimal solution and applying a locally optimal charging policy, we have reduced the complexity of the problem by decomposing it into two simpler linear programming problems. This problem decomposition yields near-optimal solutions for networks with inhomogeneous charging nodes and optimal solutions for networks with homogeneous nodes. We have also proposed a Dynamic Programming algorithm which finds optimal solutions for both homogeneous and inhomogeneous charging nodes with less computational complexity. For the system-centric problem, referred as the multi-vehicle problem, where traffic congestion effects are incorporated, we used a similar approach by aggregating vehicles into subflows and seeking optimal routing decisions for each such subflow. Again the exact formulation for this problem is a MINLP and its computational complexity directly depends on the network size and the number of subflows. We have found that a low number of subflows is adequate to obtain convergence to near-optimal solutions. As an alternative approach, we also introduced a flow-based formulation which yields approximate solutions with a computational cost reduction of several orders of magnitude, so they can be used in problems of large dimensionality. Since the problem size increase with the number of subflows, its appropriate selection is crucial to render the problem computationally manageable. To do so, we have defined a critical number of subflows which guarantees near-optimality. In particular, we have shown that by selecting the number of subflows to be equal to or larger than a critical number, the average deviation from the optimal solution never exceeds a predefined upper bound. Therefore, by selecting a desired accuracy one can choose

between proximity to optimality and computational effort needed to solve the problem. Finally, we have proposed a DP algorithm for multi-vehicle routing problem which works well for a small number of subflows but as the number of subflows increases, it loses its efficiency.
we have studied a more general version of the problem in which the vehicle flow consists of both Electric Vehicles (EVs) and Non-Electric Vehicles (NEVs). We solved the problem by aggregating vehicles into subflows and seeking optimal routing decisions for each such subflow.

Finally, we have studied the system performance in transportation networks using actual traffic data. Considering the total traveling time that all drivers experience to travel through different Origin-Destination (OD) pairs in the network as the metric, we compared the system performance under two different strategies: user-centric vs. system-centric. In particular we have investigated the system performance of an interstate highway subnetwork of the Eastern Massachusetts road network. To do so, by use of a traffic flow model, we first inferred Wardrop equilibrium flows from the spatial average speed and per-road flow capacity datasets. We then estimated the OD demand matrices accordingly. Incorporating the OD demand matrices and data-driven latency functions, we then calculated social optimum flows by solving a system-centric optimization problem for different months and time-of-day periods. Finally, we have quantified the POA as a ratio of system performance under the user-optimal policy to that under the system-optimal policy.

7.1 Future Directions

7.1.1 Extensions for lifetime maximization problem for static wireless sensor networks

So far, we have developed a centralized scheme which requires global location information, to find optimal routing policy for static WSN. Thus, an obvious direction to pursue is developing distributed versions of the same optimal routing and energy allocation problem approaches. Decentralized algorithms for the problem with ideal battery models have been already proposed (e.g., (Madan and Lall, 2006)). Thanks to the robustness property of the optimal routing, one may apply similar approach by assuming ideal batteries and the solution would be optimal for the nodes with non-ideal battery model too.

7.1.2 Extensions for lifetime maximization problem for wireless sensor networks with mobile source nodes

In our network model we have considered a single mobile source node. The work can be extended by considering multiple mobile source nodes. The network model as well as problem formulation for this case depend on the exact setting, e.g. various sensor types for mobile nodes, different priorities for delivering data packets, number of base stations, and motion dynamics of mobile nodes.

For the case of a single mobile source node, we have assumed two extreme cases: having full knowledge vs. no knowledge about the source node trajectory in advance. Our numerical examples show that the prior knowledge of the source node's motion dynamics considerably increases the network lifetime. An interesting extension of this work is to explore properties of the OCP solution in this case as well as to explore the case when we have partial knowledge about the source node trajectory in advance. For this case, a receding horizon approach seems to be a natural framework.

7.1.3 Extensions to Optimal Routing of Energy-limited Vehicles

In our network model, we have considered unlimited capacities for charging stations and vehicles begin the charging process as soon as they reach a charging station. However in practice, it may happen that there is no free station available and the EV must wait until a station becomes free. Thus, the work can be extended by introducing more realistic characteristics, such as queuing capacities, for the charging stations. This extension can be done by adding capacity constraints for all charging nodes imposing the total flow that can enter a charging node can not exceed a certain upper bound. One can then split such node into two connected nodes with the link capacity equal to the node capacity and back to the case when we only have link capacities in the network (Bertsimas and Tsitsiklis, 1997).

The problem can also be extended by taking into account constraints imposed by the grid capacity into the optimization problem framework to illustrate the significance of vehicle-grid integration. These constraints may limit the ability of vehicles to immediately be charged upon reaching a charging node too.

In the multi-vehicle routing problem, we have formulated the problem in the subflow-level and assume all vehicles in a subflow follow the same optimal policy. In fact, we have assumed that the vehicles assigned to a subflow are from a homogeneous type, i.e., size or residual energy. As a matter of fact, the arrival rate of different types of vehicles is random. Thus, the work can be extended by considering random vehicle types in each subflow and also to provide recharging decision for each individual vehicle in a subflow.

The system-centric routing problem can also be extended by considering stochastic vehicle flows where the objective is to minimize average vehicle travel times or to periodically re-solve the routing problem based on new traffic flow data.

Another important extension of this work is that of implementing an optimal routing and recharging policy for multi-vehicle routing problem. This is a challenging problem for two reasons. First, individual drivers need to be provided explicit guidance by the central controller who determines policy. Second, a driver needs to have the proper incentives to follow this policy. While the first difficulty may be addressed through Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication capabilities which are increasingly being made available to vehicles, the second one is more fundamental, since it concerns the behavior of drivers who are generally "selfish" and concerned with their own individual optimal policy. However, the emerging trend towards Connected Autonomous Vehicles (CAVs) is likely to facilitate a centrally derived system-centric optimal routing policy which could be implemented through CAVs, a research topic of growing interest.

For future research, it might also be interesting to investigate the potential of using reinforcement learning algorithms that would aim the vehicles to learn online how to minimize the total elapsed time to reach their destinations. In this context, each vehicle through its daily interaction with other vehicles and exploration of different feasible routes could eventually learn the optimal one for a given commute.

7.1.4 Extensions to data-driven estimation of Origin-Destination demand matrices

So far, we have used a GLS method to estimate the OD demand matrices for a relatively small network. Solving this problem, heavily depends on the network size and the total number of paths connecting different O-D pairs in the network. In fact, we are unable to use the same GLS method for large networks. Thus, the work can be extended by developing new algorithms to estimate O-D demand matrices for large networks. One direction to explore is to modify the existing algorithm by leveraging a bi-level optimization problem formulation. This method will allow to estimate OD demand matrices for a given larger network based on the OD demands of its representative (landmark) subnetworks. Another direction to investigate is to decompose the network into multiple subnetworks, then to estimate the O-D demand matrices for the larger network by aggregating the results obtained for the smaller subnetworks. Finally, one may adjust the obtained initial demand matrix to minimize the difference between the solution of the traffic assignment problem and the average observed flow vector.

References

Akkaya, K. and Younis, M. (2005). A survey of routing protocols in wireless sensor networks. *Elsevier Ad Hoc Network Journal*, 3(3):325–349.

Al-Karaki, J. N. and Kamal, A. E. (2004). Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11(6):6 – 28.

Artmeier, A., Haselmayr, J., Leucker, M., and Sachenbacher, M. (2010). The optimal routing problem in the context of battery-powered electric vehicles. *2nd International Workshop on Constraint Reasoning and Optimization for Computational Sustainability*, Bologna, Italy, pages 1–13.

Bai, S., Du, Y., and Lukic, S. (2010). Optimum design of an ev/phev charging station with dc bus and storage system. In *2010 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 1178–1184.

Barbarisi, O., Vasca, F., and Glielmo, L. (2006). State of charge kalman filter estimator for automotive batteries. *Control Engineering Practice 14*, pages 267–275.

Bertsekas, D. P. (2012). *Dynamic Programming and Optimal Control*, volume I. Athena Scientific.

Bertsimas, D., Gupta, V., and Paschalidis, I. C. (2014). Data-driven estimation in equilibrium using inverse optimization. *Mathematical Programming*, pages 1–39.

Bertsimas, D. and Tsitsiklis, J. H. (1997). *Introduction to Linear Optimization.* Athena Scientific and Dynamic Ideas, Massachusetts.

Bhardwaj, M. and Chandrakasan, A. P. (2002). Bounding the lifetime of sensor networks via optimal role assignments. In *Proceeding of IEEE International Conference on Computer Communications 2002*, pages 1587– 1596.

Bhardwaj, M., Garnett, T., and Chandrakasan, A. P. (2001). Upper bounds on the lifetime of sensor networks. In *Proceeding of International Conference on communications*, pages 785–790.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization.* Cambridge University Press.

Branston, D. (1976). Link capacity functions: A review. *Transportation Research*, 10(4):223–236.

Bryson, A. E. and Ho, Y. (1975). *Applied Optimal Control*. Hemisphere Publishing Corporation, Washington D.C.

Chang, J. H. and Tassiulas, L. (2004). Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions On Networking*, 12(4):609–619.

Chen, M. and Rincon-Mora, G. (2006). Accurate electrical battery model capable of predicting runtime and I-V performance. *IEEE Transactions on Energy Conversion*, 21(2):504–511.

Chiasserini, C. and Rao, R. (1999a). A model for battery pulsed discharge with recovery effect. In *Proceedings of Wireless Communications and Networking Conference*, pages 636–639.

Chiasserini, C. and Rao, R. (1999b). Pulsed battery discharge in communication devices. In *Proceedings of the 5th International Conference on Mobile Computing and Networking*, pages 88–95.

Chiasserini, C. and Rao, R. (2001). Energy efficient battery management. *IEEE Journal on Selected Areas in Communications*, 19(7):1235–1245.

Dafermos, S. C. and Sparrow, F. T. (1969). The traffic assignment problem for a general network. *Journal of Research of the National Bureau of Standards B*, 73(2):91–118.

Daler Rakhmatov, Sarma Vrudhula, D. A. W. (2003). A model for battery lifetime analysis for organizing applications on a pocket computer. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 11(6):1019–1030.

Dantu, K., Rahimi, M., Shah, H., Babe, S., and ansG. Sukhatme, A. D. (2005). Robomote: enabling mobility in sensor networks. In *Proceeding of 4th International Symposium on Information Processing in Sensor Networks*, pages 404– 409.

Di Francesco, M., DAS, S. K., and Anastasi, G. (2011). Data collection in wireless sensor networks with mobile elements: A survey. *ACM Transactions on Sensor Networks*, 8(1): Article No. 7.

Doya, K., Ishii, S., Pouget, A. and Rao, R.P.N. *Bayesian Brain: Probabilistic Approaches to Neural Coding*, The MIT Press, pages 269–298.

Doyle, M. and Newman, J. S. (1997). Analysis of capacity-rate data for lithium batteries using simplified models of the discharge process. *Journal of Applied Electrochemistry*, 27(7):846–856.

Eisner, J., Funke, S., and Storandt, S. (2011). Optimal route planning for electric vehicles in large networks. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, San Francisco, US.

Fuller, T. F., Doyle, M., and Newman, J. S. (1993). Modeling of galvanostatic charge and discharge of the lithium polymer insertion cell. *Journal of the Electrochemical Society*, 140(6):1526–1533.

Gallager, R. (1977). A minimum delay routing algorithm using distributed computation. *IEEE Transactions on Communications*, 25(1):73–85.

Ganesan, D., Govindan, R., Shenker, S., and Estrin, D. (2001). Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):11–25.

Gill, P. E., Murray, W., and Saunders, M. A. (2002). Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006.

Greenshields, B. D., Bibbins, J. R., Channing, W. S., and Miller, H. H. (1935). A study of traffic capacity. *Highway Research Board Proceedings*, 14:448–477.

Haefner, L. E. and Li, M.-S. (1998). Traffic flow simulation for an urban freeway corridor. In *1998 Transportation Conference Proceedings*, pages 1–6.

Hageman, S. C. (1993). Simple pspice models let you simulate common battery types. *Electronic Design News*, 38:117–129.

Hazelton, M. L. (2000). Estimation of origin–destination matrices from link flows on uncongested networks. *Transportation Research Part B: Methodological*, 34(7):549–566.

He, F., Yin, Y., and Lawphongpanich, S. (2014). Network equilibrium models with battery electric vehicles. *Transportation Research Part B: Methodological*, 67: 306–319.

He, T., Krishnamurthy, S., Stankovic, J. A., Abdelzaher, T., Luo, L., Stoleru, R., Yan, T., and Gu, L. (2004). Energy-efficient surveillance system using wireless sensor networks. In *2nd international conference on Mobile systems, applications, and services*, pages 270–283 .

Hillier, F. S. and Lieberman, G. J. (2005). *Introduction to operations research eighth edition*. McGraw-Hill.

Ho, F. and Ioannou, P. (1996). Traffic flow modeling and control using artificial neural networks. *IEEE Control Systems*, 16(5):16 – 26.

International Energy Agency (2011). Technology Roadmap: Electric and plug-in hybrid electric vehicles.
http://www.iea.org/publications/freepublications/publication/technology-roadmap-electric-and-plug-in-hybrid-electric-vehicles-evphev.html

Jongerden, M. and Haverkort, B. (2009). Which battery model to use? *Institution of Engineering and Technology (IET) Software*, 3(6):445 – 457.

Jongerden, M. R. and Haverkort, B. R. (2008). Battery modeling. Technical report, Centre for Telematics and Information Technology, University of Twente, Enschede. http://doc.utwente.nl/64556/1/BatteryRep4.pdf.

Joos, G., de Freige, M., and Dubois, M. (2010). Design and simulation of a fast charging station for phev/ev batteries. In *2010 IEEE Electric Power and Energy Conference (EPEC)*, pages 1–5.

Khuller, S., Malekian, A., and Mestre, J. (2011). To fill or not to fill: The gas station problem. *ACM Transactions on Algorithms*, 7(3): 36:1–36:16.

Krontiris, I., Giannetsos, T., and Dimitriou, T. (2008). Launching a sinkhole attack in wireless sensor networks; the intruder side. In *Proceeding of IEEE International Conference on Wireless , Mobile Computing, Networking and Communication*, pages 526 – 531.

Kuhne, R. and Rodiger, M. (1991). Macroscopic simulation model for freeway traffic with jams and stop-start waves. In *Proceedings of the Winter Simulation Conference*, pages 762 – 770.

Laibowitz, M. and Paradiso, J. (2005). Parasitic mobility for pervasive sensor networks. In *3rd International Conference on Pervasive Computing*, pages 255–278.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3): 345–358.

Lawson, B. (2005) Battery and energy technologies - state of charge (soc) determination. In *Electropaedia*,
http://www.mpoweruk.com/soc.htm.

Madan, R. and Lall, S. (2006). Distributed algorithms for maximum lifetime routing in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 5(8):2185–2193.

Manwell, J. and McGowan, J. (1993). Lead acid battery storage model for hybrid energy systems. *Solar Energy*, 50:399–405.

Manwell, J. F. and McGowan, J. G. (1994). Extension of the kinetic battery model for wind/hybrid power systems. In *Proceeding of European Wind Energy Conference*, pages 294–289.

Martin, T. L. (1999). *Balancing Batteries, Power, and Performance: System Issues in CPU Speed-Setting for Mobile Computing.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA.

Megerian, S. and Potkonjak, M. (2003). *Wireless Sensor Networks.* Wiley Encyclopedia of Telecommunications. Wiley-Interscience, New York, NY.

Newman, J. (1998). Fortran programs for the simulation of electrochemical systems.

Ning, X. and Cassandras, C. G. (2009). On maximum lifetime routing in wireless sensor networks. In *48th IEEE Conference on Decision and Control*, pages 3757 – 3762.

Panigrahi, D., Chiasserini, C., Dey, S., Rao, R., Raghunathan, A., and Lahiri, K. (2001). Battery life estimation of mobile embedded systems. In *Proceeding of International Conference on VLSI Design*, pages 57–63.

Park, V. D. and Corson, M. S. (1997). A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceeding of IEEE International Conference on Computer Communications*, pages 1405–1413.

Paschalidis, I. C. and Wu, R. (2012). Robust maximum lifetime routing and energy allocation in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2012(Article ID 523787).

Patriksson, P. (2015). *The traffic assignment problem: models and methods.* Dover Dover Publications.

Patterson, M. A. and Rao, A. V. (2014). Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Transactions on Mathematical Software (TOMS)*, 41(1):1:1-1:37.

Perkins, C. E. and Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector (dsdv) routing for mobile computers. In *Proceedings of the conference on Communications architectures, protocols and applications*, pages 234–244.

Pourazarm, S. and Cassandras, C. G. (2015). System-centric minimum-time paths for battery-powered vehicles in networks with charging nodes. In *Proceeding of 5th IFAC Conference on Analysis and Design of Hybrid Systems*, pages 111–116.

Pourazarm, S. and Cassandras, C. G. (2016). Optimal routing and charging of energy-limited vehicles in traffic networks. *International Journal of Robust and Nonlinear Control*, 26:1325–1350.

Raja, A. and Su, X. (2009). Mobility handling in mac for wireless ad hoc networks, In *Wireless Communications and Mobile Computing*, 9(3):303–311.

Rakhmatov, D. and Vrudhula, S. (2001). An analytical high-level battery model for use in energy management of portable electronic systems. In *Proceedings of the International Conference on Computer Aided Design (ICCAD'01)*, pages 488–493.

Rakhmatov, D., Vrudhula, S., and Wallach, D. A. (2002). Battery lifetime predictions for energy-aware computing. In *Proceedings of the 2002 International Symposium on Low Power Electronics and Design (ISLPED'02)*, pages 154–159.

Rao, A. V. (2009). A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135(1):497–528.

Rao, R., Vrudhula, S., and Rakhmatov, D. N. (2003). Battery modeling for energy-aware system design. *Computer*, 36(12):77–87.

Rao, V., Singhal, G., Kumar, A., and Navet, N. (2005). Battery model for embedded systems. In *Proceeding of 18th International Conference on VLSI Design*, pages 105–110.

Rezazadeh, J., Moradi, M., and Ismail, A. S. (2012). Mobile wireless sensor networks overview. *International Journal of Computer Communications and Networks*, 2(1):17–22.

Roughgarden, T. (2005). *Selfish Routing and the Price of Anarchy.* The MIT Press.

Sachenbacher, M., Leucker, M., Artmeier, A., and Haselmayr, J. (2011). Efficient energy-optimal routing for electric vehicles. In *Proceeding of the 25th AAAI Conference on Artificial Intelligence*, pages 1402–1407.

Schneider, M., Stenger, A., and Goeke, D. (2014). The electric vehicle routing problem with time windows and recharging stations. *Journal of Transportation Science*, 48(4):500–520.

Shah, R. and Rabaey, J. (2002). Energy aware routing for low energy ad hoc sensor networks. *Proceeding of IEEE Wireless Communications and Networking Conference*, pages 350–355.

Shah, R. C., Roy, S., Jain, S., and Brunette, W. (2003). Data mules: Modeling a three-tier architecture for sparse sensor networks. In *2nd ACM International Workshop on Wireless Sensor Networks and Applications*, pages 30–41.

Shnayder, V., Hempstead, M., Chen, B., Allen, G. W., and Welsh, M. (2004). Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 188–200.

Siddiqi, U. F., Shiraishi, Y., and Sait, S. M. (2011). Multi-constrained route optimization for electric vehicles (evs) using particle swarm optimization. In *11th International Conference on Intelligent Systems Design and Application (ISDA)*, pages 391–396.

Singh, S. and Raghavendra, C. S. (1998). PAMAS-power aware multi-access protocol with signaling for ad hoc networks. *ACM Computer Communication Review*, 28(3):5–26.

Society of Automotive Engineers (SAE), Standard J1772 (2012). Electric Vehicle and Plug in Hybrid Electric Vehicle Conductive Charge Coupler.

Srinivasan, W. and Chua, K.-C. (2007). Trade-offs between mobility and density for coverage in wireless sensor networks. In *13th ACM International Conference on Mobile Computing and Networking*, pages 39–50.

Sunder, K. and Rathinam, S. (2012). Route planning algorithms for unmanned aerial vehicles with refueling constraints. In *2012 American Control Conference (ACC)*, pages 3266–3271.

Sweda, T. M. and Klabjan., D. (2012). Finding minimum-cost paths for electric vehicles. In *Proceeding of 2012 IEEE International Electric Vehicle Conference (IEVC)*, pages 1–4.

Touati-Moungla, N. and Jost, V. (2012). Combinatorial optimization for electric vehicles management. In *Journal of Energy and Power Engineering*, 6(5):738–743.

Tseng, Y. C., Wang, Y. C., Cheng, K. Y., and Hsieh, Y. Y. (2007). imouse: An integrated mobile surveillance and wireless sensor system. *IEEE Computer*, 40:60 – 66.

Vrudhula, S. and Rakhmatov, D. (2003). Energy management for battery powered embedded systems. *ACM Transactions on Embedded Computing Systems*, 2(3):277–324.

Wang, T. and Cassandras, C. G. (2013). Optimal control of batteries with fully and partially available rechargebility. *Automatica*, 48(8):1658–1666.

Wang, T. and Cassandras, C. G. (2013). Optimal control of multi-battery energy-aware systems. *IEEE Transactions On Control Systems Technology*, 21(5):1874–1888.

Wang, X., Xiao, N., Xie, L., Frazzoli, E., and Rus, D. (2015). Analysis of price of anarchy in traffic networks with heterogeneous price-sensitivity populations. *IEEE Transactions on Control Systems Technology*, 23(6):2227 – 2237.

Wang, Z. M., Basagni, S., Melachrinoudis, E., and Petrioli, C. (2005). Exploiting sink mobility for maximizing sensor networks lifetime. In *38th Hawaii International Conference on System Sciences*, pages 1–9.

Wood, A. D. and Stankovic, J. A. (2002). Denial of service in sensor networks. *IEEE Computer*, 35(10):54–62.

Worley, O., Klabjan, D., and Sweda, T. (2012). Simultaneous vehicle routing and charging station siting for commercial electric vehicles. In *Proceeding of IEEE International Electric Vehicle Conference (IEVC)*, pages 1–3.

Wu, X. and Cassandras, C. G. (2005). A maximum time optimal control approach to routing in sensor networks. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, pages 1137–1142.

Youn, H. and Jeong, H. (2008). The price of anarchy in transportation networks: Efficiency and optimality control. *Physical Review Letters*, 101(128701):1–4.

Zhang, F. and Shi, Z. (2009). Optimal and adaptive battery discharge strategies for cyber-physical systems. In *48th IEEE Conference on Decision and Control*, pages 6232–6237.

Zhang, J., Pourazarm, S., Cassandras, C. G., and Paschalidis, I. C. (2016). The price of anarchy in transportation networks by estimating user cost functions from actual traffic data. *To appear in Proceeding of 55th IEEE Conference on Decision and Control; arXiv:1606.02194*.

Zhao, M., Li, J., and Yang, Y. (2014). A framework of joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks. *IEEE Transactions on Mobile Computing*, 13:2689 – 2705.

CURRICULUM VITAE

Sepideh Pourazarm

Education

PhD - Systems Engineering
Boston University, College of Engineering, Boston, MA, January 2017
Thesis: "Control and Optimization Approaches for Energy-limited Systems"

M.Sc. - Electrical Engineering / Control System
K.N.Toosi University of Technology, Tehran, Iran, June 2007

B.Sc. - Electrical Engineering
K.N.Toosi University of Technology, Tehran, Iran, September 2004

Experience

CODES Lab - Boston University
Graduate Research Assistant, May 2013 - January 2017
– Investigated performance of transportation network in Eastern Massachusetts using actual data.
– Developed algorithm to find optimal routing scheme maximizing lifetime of wireless sensor networks.
– Developed algorithm to determine optimal routing for Electric vehicles

Boston University
Graduate Teaching Fellow, Engineering Computation Course, September 2012 - 2013

Nargan Engineering and Constructors
Instrumentation and Control Engineer, Tehran, Iran, June 2008 - August 2011
– Prepared data sheets and technical bid evaluation for field instruments.
– Prepared control and instrument technical room layouts and cable layout.

Awards

- ACM's Student Research Competition TravelAward, October 2016
- CISE honorable mention, BU Graduate Research Symposium March 2015

- Best Student Paper Finalist Award
  IEEE International Conference on Networking, Sensing and Control, 2014
- Systems Engineering PhD Student Travel Award
  Grace Hopper Conference, October 2014
  IFAC Conference on Analysis and Design of Hybrid Systems, October 2015
- NSF Travel Award, NSF Data Science Workshop, August 2015

Selected Publications

Journal Papers

- S. Pourazarm, C.G. Cassandras, Optimal Routing of Energy-aware Vehicles in Transportation Networks with Inhomogeneous Charging Nodes, IEEE Transactions on Intelligent Transportation Systems (under review)
- S. Pourazarm, C.G. Cassandras, Optimal Routing for Lifetime Maximization of Wireless Sensor Networks with a Mobile Source Node, IEEE Transactions on Control of Network Systems, 2016
- S. Pourazarm, C.G. Cassandras, Energy-Based Lifetime Maximization and Security of Wireless Sensor Networks with General Non-ideal Battery Models, IEEE Transactions on Control of Network Systems, 2016
- S. Pourazarm, C.G. Cassandras, T. Wang, Optimal Routing and Charging of Energy-limited Vehicles in Traffic Networks, International Journal of Robust and Nonlinear Control, 2015
- T. Wang, C.G. Cassandras, S. Pourazarm, Optimal Motion Control for Energy-aware Electric Vehicles, Control Engineering Practice Journal, 2015
- C.G. Cassandras, T. Wang, S. Pourazarm, Optimal Routing and Energy Allocation for Lifetime Maximization of Wireless Sensor Networks with Non-ideal Batteries, IEEE Transactions on Control of Network Systems, 2014

Peer-reviewed Conference Proceedings

- J. Zhang, S. Pourazarm, I. Paschalidis, C.G. Cassandras, Data-driven Estimation of Origin-Destination Demand and User Cost Functions for the Optimization of Transportation Networks, IFAC World Congress 2017, (Submitted)
- J. Zhang, S. Pourazarm, I. Paschalidis, C.G. Cassandras, The Price of Anarchy in Transportation Networks by Estimating User Cost Functions from Actual Traffic Data, in the Proceeding of 55th IEEE Conference on Decision and Control (CDC2016), December 2016

- S. Pourazarm, C.G. Cassandras, Lifetime Maximization of Wireless Sensor Networks with a Mobile Source Node, in the Proceeding of 54th EEE Conference on Decision and Control (CDC2015), December 2015
- S. Pourazarm, C.G. Cassandras, System-Centric Minimum-Time Paths for Battery -Powered Vehicles in Networks with Charging Nodes, IFAC Conference on Analysis and Design of Hybrid Systems, August 2015
- S. Pourazarm, C.G. Cassandras, A. Malikopoulos, Optimal Routing of Electric Vehicles in Networks with Charging Nodes: A Dynamic Programming Approach, in Proceeding of IEEE International Electric Vehicle Conference, December 2014
- S. Pourazarm, C.G. Cassandras, Optimal Routing of Energy-aware Vehicles in Networks with In-homogeneous Charging Nodes, in the Proceeding of 22nd IEEE MED Conference on Control and Automation, June 2014

Professional activities and services

Peer Reviewing

- Automatica
- Control Engineering Practice
- IEEE Transactions on Control of Network Systems
- IEEE Transactions on Intelligent Transportation Systems
- Journal of Network and Computer Applications
- IEEE Transactions on Information Theory
- International Journal of Robust and Nonlinear Control

Organizational Activities

- CISE Graduate Student Workshop, Organizer and Student Chair, Boston University, January 2016
- Center for Information and Systems Engineering (CISE) Seminars, Student Host, 2013 - 2016