Learning in Complex Dynamic Systems:

with Applications to Perpetual Flight, Energy Management, Distributed Decision Making, and Social Networks

Saghar Hosseini Sianaki

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington, Seattle

2016

Reading Committee:
Mehran Mesbahi, Chair
Kristi Morgansen
Anshu Narang-Siddarth

Program Authorized to Offer Degree: Aeronautics & Astronautics Engineering

University of Washington

Abstract

Learning in Complex Systems:
with Applications to Perpetual Flight, Energy Management, Distributed Decision Making, and Social Networks

Saghar Hosseini

Chair of the Supervisory Committee:

Professor Mehran Mesbahi, Aeronautics & Astronautics

This dissertation addresses learning in complex dynamic systems with applications to perpetual flight, energy management, collaborative decision making, and social networks. By increasing the size and complexity of network systems, decentralized optimization schemes or machine learning algorithms are desired for scaling up the automated learning process, reducing data transmission, and ensuring robustness in the presence of local failures. This work approaches these challenges from two fronts: complex dynamics associated with individual agents in the network; and protocols which are run on individual agents in the network. In this direction, energy management for aerial vehicles and small smart grids have been studied . With the objective to develop smart autonomous distributed systems performing in a highly uncertain environment, online distributed learning algorithms have been proposed. These algorithms allow the network topology to adapt and each agent learns the model based on its local data and the information it receives from its neighboring agents. Central to our analysis of the performance of such online distributed algorithms is the examination of the role of the network structure in the so-called social regret. In addition, this dissertation provides analysis of large scale time-varying and state-dependent networks to develop scalable distributed learning algorithms for online network estimation. In this problem, the state of the nodes are affected by their neighboring nodes, inspired by the opinion dynamics. A sampling approach is then applied to scale up the algorithm for massive networks. Our theoretical results demonstrate a good (sub-linear)

regret bound for the topology estimation problem with limited and online observations of the underlying communication links.

Contents

Acknowledgments

I had the pleasure to spend the four years of my Ph.D. in Robotics, Aerospace, and Information Networks (RAIN) lab at the University of Washington (UW). Foremost, I would like to express my sincere gratitude to my advisor Prof. Mehran Mesbahi for the continuous support of my Ph.D. study and research, for his patience, motivation, enthusiasm, and immense knowledge.

Besides my advisor, I would like to thank the rest of the RAIN lab's members for their supportive spirit, the stimulating discussions, the sleepless nights we were working together before deadlines, and for all the fun we have had in the last four years. In particular, I'm grateful to Dr. Airlie Chapman, Prof. Ran Dai, and Eric Schoof for their friendship and constant support whom I will miss deeply. In addition, much of the work in this thesis would not have been possible without the collaboration of Dr. Airlie Chapman, Prof. Ran Dai, and Eric Schoof.

I have been fortunate to have taken classes taught by many fantastic UW professors and I would especially like to thank my reading and doctoral committee, Profs. Kristi Morgansen, Anshu Narang, Lin Xiao, Behcet Acikmese, and Ali Shojaie. Their enthusiasm and insightful comments opened many new avenues for exciting research.

Last but not the least, I would like to thank my family; my parents, Fatemeh Choupani and Hossein Hosseini Sianaki, for supporting me throughout my life. My brother, Saman, for his willingness to listen to my chatter and being a supportive phone call away. My little brother, Iliya, for bringing joy to my life and his constant effort to cheer me up.

Dedication

To Fatemeh Choupani and Hossein Hosseini Sianaki, my parents who have given me the opportunity of an education from the best institutions and support throughout my life.

To Saman Hosseini Sianaki, my brother and my best friend.

To Iliya Hosseini Sianaki, my little brother and my wonder of wonders.

To my friends, my second family who made so many great memories for me during these past years.

Words cannot express how much I love you all.

Introduction

Complex engineered systems have received a great deal of attention in recent years. The past decade has seen successful applications of such systems in many areas, ranging from long endurance Unmanned Aerial Vehicles (UAVs), smart grids, transportation systems, to community detection in massive social networks. Some of the complexities in these systems come from the non-linear and non-convex dynamics, constraints imposed by the environment, high dimension of states, and the uncertainty and disturbances affecting the systems. The two directions to approach these systems are the building blocks behind the two structured parts of the dissertation. In summary, these two parts, together with related publications are:

Part 1. Energy Management under Uncertainties

The first part of this dissertation addresses two types of complexities in the dynamic systems including the non-linearity, non-convexity, and high dimensional states. This part is focused on renewable energy sources and energy management for aerial vehicles and small smart grids. The application of renewable energy sources on aerial vehicles is a new and emerging technology. Collecting solar energy has the potential to vastly improve fuel consumption and flight endurance in UAVs when paired with intelligent energy management. In pursuit of this goal, several solar powered UAVs have been built and flown since 1974. The latest solar powered UAV, Aquila developed by Facebook, will be able to circle a remote region for up to 90 days, providing internet connectivity to people from an altitude of 60,000 to 90,000 feet. These types of UAVs can replace high-cost and difficult to maintain satellites for surveillance, reconnaissance and communication purposes. One of the benefits of energy management in aerial vehicles is the emergence of long endurance UAVs. Therefore, the vision of perpetual flight is becoming feasible providing a wide range

of commercial and scientific applications such as disaster management, crop and forest fire monitoring, remote sensing, mapping, and search and rescue.

My research has examined optimal path planning, coverage, and power allocation problems for solar powered UAV flight. This problem addresses the delicate balance between optimizing sensor coverage range for better observations and maximizing the total energy stored in the batteries while coordinating the operations of all included electric components (Figure 0.0.2).

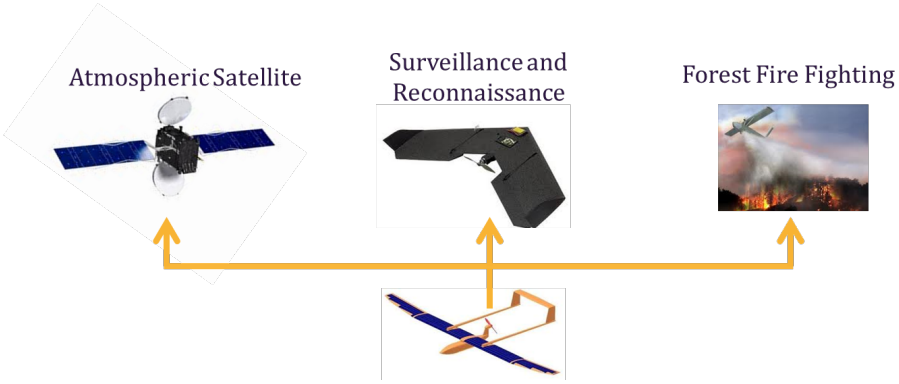


FIGURE 0.0.1. Solar powered UAVs applications.

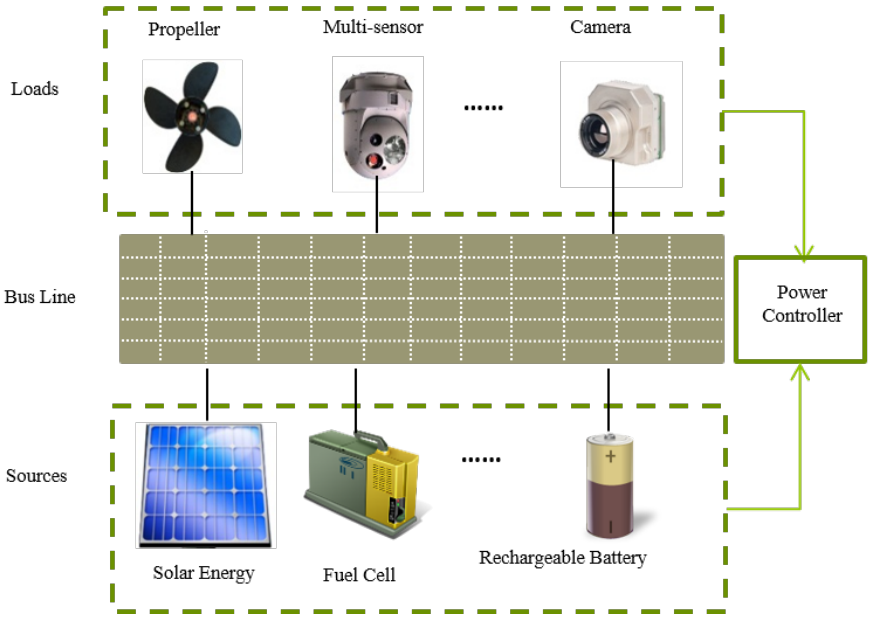


FIGURE 0.0.2. Solar powered UAVs electric components.

Moreover, this dissertation addresses the optimal power management in smart electric cooling systems as one element of smart grids. Power grids have become over-stressed due to uncertainties in schedules and transfers across regions and increasing usage of renewable energy systems. Therefore, smart grids employ telecommunication, control, and optimization to achieve adaptability, efficiency and reliability of power grids.

Future smart grids rely on new levels of transparency and coordination between providers and consumers of electric energy. On one side, the electricity providers will be tracking and estimating the consumer's daily, weekly and seasonal demands to coordinate the energy requests. On the other side, some customers will be enticed to change their consumption habit to save cost and alleviate peak demand by adopting new technologies, e.g., dynamic pricing. Finding an efficient demand schedule is challenging due to the nonlinear thermodynamics and complex heat exchange processes associated with environment and the heating/cooling systems, the fluctuating price of the electricity, and the user preference. My research has proposed an optimal power management strategy for cooling systems in the presence of unknown uncertainties in the environment and dynamic pricing by blending ideas from online learning and Mixed Integer Linear Programming (MILP).

Publications per chapter:

Chapter 1. Optimal Path Planning and Power Allocation for a Long Endurance Solar-Powered UAV

[1] Saghar Hosseini, Ran Dai, Mehran Mesbahi, "Optimal Path Planning and Power Allocation for a Long Endurance Solar-Powered UAV," In Proc. of the American Control Conference, 2013

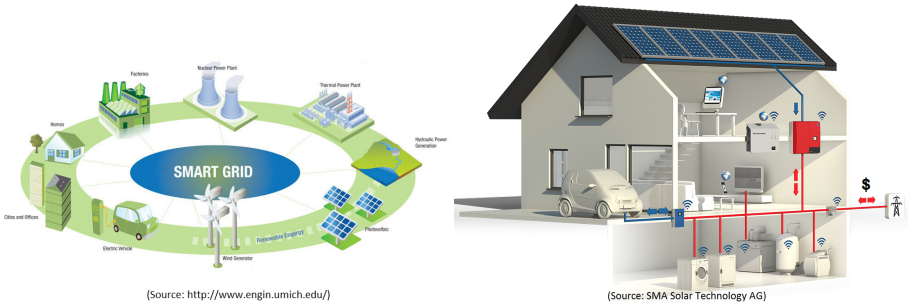


FIGURE 0.0.3. Smart grids and smart homes.

[2] Ran Dai, Unsik Lee, Saghar Hosseini, Mehran Mesbahi, “Optimal Path Planning for Solar-Powered UAVs based on Unit Quaternions,” In Proc. of IEEE Conference on Decision and Control, 2012

Chapter 2. Energy Aware Aerial Surveillance for a Long Endurance Solar-Powered UAV

[3] Saghar Hosseini and Mehran Mesbahi, "Energy Aware Aerial Surveillance for a Long Endurance Solar-Powered UAV,". In Proc. of the AIAA Guidance, Navigation and Control Conference, 2013

[4] Saghar Hosseini and Mehran Mesbahi, "Energy Aware Aerial Surveillance for a Long Endurance Solar-Powered UAV,". AIAA Journal of Guidance, Control, and Dynamics, 2015 (submitted)

Chapter 3. Power Management of Cooling Systems with Dynamic Pricing

[5] Saghar Hosseini, Ran Dai, and Mehran Mesbahi, " Power Management of Cooling Systems with Dynamic Pricing". In Proc. of the American Control Conference, 2014

Part 2. Large Scale Systems

The second part of this dissertation addresses two types of complexities in dynamic systems including uncertainties in the objective and the high dimensional states. More specifically, this part focuses on the online estimation and cooperative decision making in multiple-agent systems where the complexities arise from the large number of agents, unknown uncertainties in the model, and time-varying communication links. Networked systems have received a great deal of attention in areas ranging from multi agents coordination, distributed estimation in sensor networks, decentralized tracking, environmental monitoring, robotics, target recognition, air traffic control, to industrial and manufacturing automation. In the face of local failures, employing multiple cooperative agents leads to a more reliable and faster mission execution. On the other hand, by increasing the size and complexity of network systems, decentralized optimization schemes are desired for reducing data transmission and ensuring robustness in the presence of packet drops. With this consideration, we have developed an analogous approach to the first part of the dissertation with investigation of cooperative decision making in UAV swarm for forest firefighting. In this problem, the UAVs have to reach an agreement on the location of the

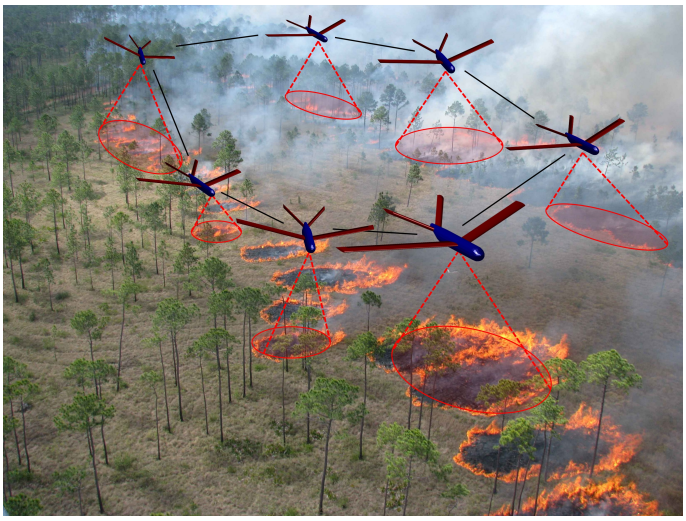


FIGURE 0.0.4. Distributed target tracking and formation acquisition in UAVs swarm.

fire front and the swarm must preserve a formation for communication maintenance. This application fits into a class of distributed optimization problems due to a limited range of communication among UAVs. Moreover, the swarm is facing uncertainties both in the changing location of the wildfire, and the underlying communication topology.

To address this scenario, we have proposed an online distributed optimization algorithm which allows UAVs to adapt the network information flow and make a decision based on the information they receive from their neighboring agents.

In addition, we have addressed optimization and the collaborative decision making over state-dependent networks. Generally, the underlying communication topology of network systems is time-varying and dependent on the state of the nodes. In most cases, the topology of mobile or social networks are not known ahead of time and thus can be imposed as an online learning problem. Therefore, we have proposed a fast convex optimization algorithm that estimates the network topology based on the history of communication links where the nodes are approximately following the opinion dynamics. Due to the large size of the networks of interest, a sampling approach is applied to select a subset of the network and scale the estimation algorithm.

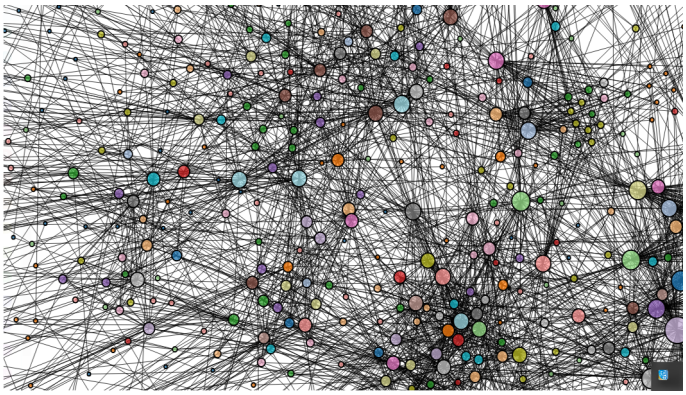


FIGURE 0.0.5. Massive time-varying networks.

Publications per chapter:

Chapter 4. Online Distributed Optimization on Dynamic Networks

[6] Saghar Hosseini, Airlie Chapman, and Mehran Mesbahi, "Online Distributed Optimization via Dual Averaging," In Proc. of IEEE Conference on Decision and Control, 2013

[7] Saghar Hosseini, Airlie Chapman, and Mehran Mesbahi, "Online Distributed Optimization on Dynamic Networks," IEEE Transactions on Automatic Control, 2017 (to appear)

Chapter 5. Online Distributed Optimization with Local Constraints

[8] Saghar Hosseini, Airlie Chapman, and Mehran Mesbahi, "Online Distributed ADMM via Dual Averaging," In Proc. of IEEE Conference on Decision and Control, 2014

[9] Saghar Hosseini, Airlie Chapman, and Mehran Mesbahi, "Online Distributed ADMM on Networks: Social Regret, Network Effect, and Condition Measures," Automatica, 2016 (submitted)

Chapter 6. Online Optimization over Time-varying, State-Dependent Networks

[10] Saghar Hosseini and Mehran Mesbahi, "Online Estimation of Large State-Dependent and Time-Varying Social Networks, " In Proc. of IEEE Conference on Decision and Control, 2016 (in preparation)

[11] Saghar Hosseini and Mehran Mesbahi, "Online Estimation of Large State-Dependent and Time-Varying Networks with Applications to Social Graphs," IEEE Transactions on Automatic Control, 2016 (in preparation)

Part 3. Final Remarks and Road Ahead

Chapter 7 concludes the dissertation with some remarks and directions for future work.

Part 1

Energy Management under Uncertainties

I. Background and Preliminaries

Nomenclature

ϕ	Bank angle, rad
ψ	Heading angle, rad
γ	Flight path angle, rad
a	Azimuth of the sun, rad
e	Elevation of the sun, rad
i	Incidence angle of sun rays, rad
θ	Sensor field of view angle, rad
c	Sensor coverage range, m
x	x coordinate in a flat earth-fixed reference frame, m
y	y coordinate in a flat earth-fixed reference frame, m
z	z coordinate in a flat earth-fixed reference frame, m
V	Flight speed, m/s
n_v	Vertical load factor
n_h	Horizontal load factor
η_{prop}	Efficiency of the propeller
η_{sol}	Efficiency of the solar cells
$z_{\max}$	Maximum altitude gain, m
z_c	Rate of climb, m/s
λ	Regularization constant
L	Lift force, N
C_L	Coefficient of lift
D	Drag force, N
C_D	Coefficient of drag
C_{D0}	Parasitic drag coefficient
T	Thrust force, N
ϵ	Oswald efficiency factor
g	Gravitational acceleration, m/s^2
ρ	Air density, kg/m^3

P_{batt}	Battery pack output power, $watt$
Q	Battery pack charge capacity, Ah
V_{oc}	Battery pack open circuit voltage, $Volt$
R	Battery pack internal resistance, ohm
SOC	Battery pack state of charge, %
P_{Tot}	Total power, $watt$
P_{sd}	Solar spectral density, $watt/m^2$
C	Coverage cost, m
z_g	Rate of descent, m/s
N	Number of iterations

This section presents a brief background on the models for the aircraft kinematics, electric components, sensor coverage range, and regret definition used in Chapters 1 and 2. The main power systems considered in the solar UAVs are the photovoltaic cells, electric motor, and rechargeable batteries. There are other power units on-board of a solar powered UAV with negligible power consumption compared to the three components mentioned above.

I.1. Components of the Power System. In this section we discuss the models for the three power systems on-board.

I.1.1. *Electric Motor.* We assume the solar powered UAV is equipped with an electric engine attached to a propeller. The power consumed by the propeller engines can be expressed as

$$P_{Eng} = \frac{TV}{\eta_{prop}}, \tag{I.1}$$

where T is the engine thrust, V is the speed of the UAV, and η_{prop} is the efficiency of the propeller. Moreover, the maximum available thrust is denoted by $T_{\max}$, that is predetermined by the characteristics of the aircraft engine.

I.1.2. *Photovoltaic Cells.* Another major power system in solar powered UAVs is photovoltaic cells. The power harvested from the solar radiation is modeled based on the

incidence angle of solar rays over the photovoltaic cells denoted by i, and solar cell parameters [**16**]:

$$P_{Sun} = \eta_{sol} P_{sd} S \cos i, \tag{I.2}$$

where η_{sol} is the efficiency of the photovoltaic cells, S is the photovoltaic cells area, and P_{sd} is the solar spectral density. The incidence angle i is a function of the sun's position in the sky and the UAV attitude. The sun's position can be described by its azimuth and elevation angles [**29**].Thus, the vector $\widehat{s_E}$ representing the sun's position vector in the earth-fixed frame can be expressed as,

$$\widehat{s_E}(t) = [\cos e(t) \cos a(t) \;\; \cos e(t) \sin a(t) \;\; \sin e(t)]^T ,$$

where $a(t)$ is the azimuth angle and $e(t)$ is the elevation angle of the sun at time t. Note that the solar angles a and e are changing over time and are measured at all time. For example, the elevation and azimuth angles of the sun on a typical spring day in San Diego, California, is depicted in Fig. I.1.

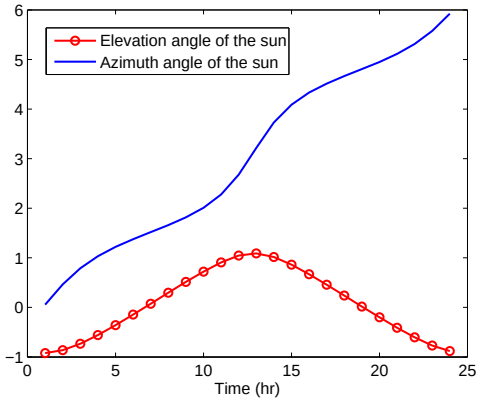


FIGURE I.1. The azimuth and elevation angles of the sun on April 1st, 2012, in San Diego, CA. The simulation time starts at 1am and ends at midnight.

From the aircraft perspective, the cosine function for the incidence angle i can be obtained from

$$\cos i = \widehat{s_A}^T . z_A, \tag{I.3}$$

where $\widehat{s_A}$ is the unit vector to the sun in the aircraft-fixed frame and z_A is the aircraft-fixed vertical axis defined as $z_A = [0\ 0\ 1]^T$. Moreover, the unit vector to the sun in aircraft-fixed frame can be obtained based on the orientation of the UAV :

$$\widehat{s_A}(t) = R_1(\phi)R_2(\gamma)R_3(\psi)\widehat{s_E}(t), \tag{I.4}$$

where R_1, R_2, and R_3 are rotation matrices about the first, second, and third axis, respectively.

I.1.3. *Rechargeable Battery Pack.* The rechargeable battery pack stores solar power harvested by the photovoltaic cells and delivers it to the electric motor. The battery pack output power is modeled based on the circuit representation of lithium-sulfur cells (Li-S) **[30]**:

$$P_{Batt} = -V_{OC} Q S\dot{O}C - R(Q S\dot{O}C)^2, \tag{I.5}$$

where V_{OC} is the open circuit voltage, R is the internal resistance, Q is the capacity, and SOC is the state of charge defined as the ratio of current charge capacity to maximum charge capacity. The battery pack specification **[31]** implies that the open circuit voltage of the battery pack is a function of the cell temperature and the value of SOC. However, the impact of SOC is negligible under normal temperature and V_{OC} is assumed to be constant in our subsequent analysis.

In order to optimize the limited life cycle of rechargeable batteries **[30]**, the SOC should stay in the range of

$$0.25 \leq SOC \leq 0.9. \tag{I.6}$$

In addition, the capability of battery pack determines the charge and discharge performance and their limitation can be described as

$$P_{Batt_0,Charge} \leq P_{Batt} \leq P_{Batt_0,Discharge}. \tag{I.7}$$

Note that the input power to the rechargeable battery pack is negative and thus $P_{Batt_0,Charge} < 0$, while the discharge power is positive implying that $P_{Batt_0,Discharge} > 0$. Based on Eq. (I.5), the rate of charge/discharge of battery pack can be represented as

$$\frac{d}{dt}SOC = -\frac{V_{OC} - \sqrt{V_{OC}^2 - 4P_{Batt}R}}{2QR}. \tag{I.8}$$

I.2. Aircraft Kinematics. In this chapter, we assume that the UAV flies in still air, moves along its thrust vector, and its point mass kinematics in 3D space is as follows **[32]**:

$$\begin{gathered} \dot{x} = V\cos\psi\cos\gamma, \quad \dot{y} = V\sin\psi\cos\gamma, \quad \dot{z} = V\sin\gamma \\ \dot{\psi} = \frac{g}{V}\left(\frac{n_h}{\cos\gamma}\right), \quad \dot{\gamma} = \frac{g}{V}(n_v - \cos\gamma) \\ \dot{V} = \left(\frac{T-D}{mg} - \sin\gamma\right) g. \end{gathered} \tag{I.9}$$

The states x, y, and z are the aircraft position coordinates in the earth-fixed reference frame, V is the UAV velocity, ψ is the heading angle, and γ is the flight path angle. The drag and thrust forces are denoted by D and T, respectively. The parameters n_v and n_h are the vertical and horizontal load factors defined as

$$n_h = \frac{L\sin\phi}{W}, \quad n_v = \frac{L\cos\phi}{W}, \tag{I.10}$$

where W represents the aircraft weight and L represents the lift force. Thus, the bank angle ϕ can be expressed as $\phi = \tan^{-1}\left(\frac{n_h}{n_v}\right)$ based on Eq. (I.10). Moreover, the standard lift and drag equations gives the drag force D in Eq. (I.9) as

$$\begin{gathered} C_L = \frac{nW}{\frac{1}{2}\rho V^2 S}, \quad C_D = C_{D0} + KC_L^2 \\ D = \frac{1}{2}\rho V^2 S C_D, \end{gathered} \tag{I.11}$$

where C_L is the lift coefficient, C_D is the drag coefficient, C_{D0} is the coefficient of parasitic drag, and K is the aerodynamic coefficient. The parameter n is the magnitude of the load factor, which can be determined by n_h and n_v such that $n = \sqrt{n_h^2 + n_v^2}$. Note that the air density ρ in Eq. (I.11) is estimated from the U.S. Standard Atmospheric database **[33]**.

I.3. Sensor Coverage Range. Regarding the surveillance problem, we need to define the sensor coverage range and resolution in this section. The sensor coverage is modeled as a linear function of altitude **[20]**. More specifically, the sensor has a constant field of view and its coverage range increases as the UAV gain altitude. Nevertheless, the sensor resolution diminishes at high altitudes which is modeled as

$$c(z) = \begin{cases} z \tan\theta & if\ 0 \le z \le h_{\max} \\ h_{\max} \tan\theta - \frac{h_{\max} \tan\theta}{h_0 - h_{\max}}(z - h_{\max}) & if\ h_{\max} \le z \le h_0 \end{cases}, \tag{I.12}$$

where c and z are the coverage range and altitude, respectively. The sensor field of view is denoted by θ; $h_{\max}$ is the altitude at which the sensor has maximum coverage range. Also, h_0 represents the altitude at which the sensor resolution is not appropriate for the aerial surveillance and without loss of generality, we can assume the coverage range is zero at $z \ge h_0$. The coverage range of a sensor with a given field of view, θ, is also shown in Fig. I.2.

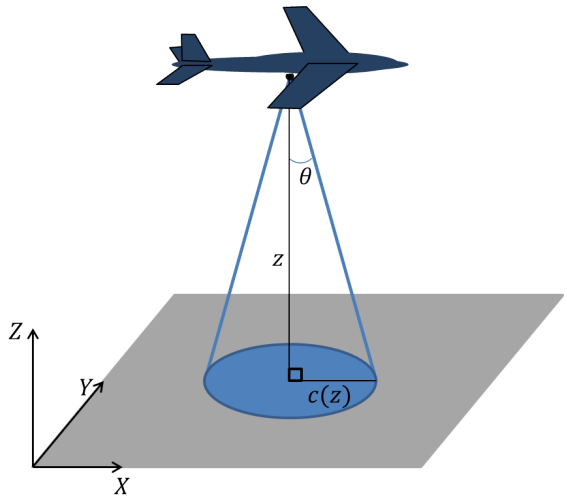


FIGURE I.2. The sensor coverage range on the $X - Y$ plane.

I.4. Regret. In online optimization, an online algorithm is used to generate a sequence of decisions $\{\xi(k)\}_{k=1}^{N}$ to solve the following problem:

$$\min_{\xi} f_k(\xi) \text{ subject to } \xi \in \chi. \tag{I.13}$$

The number of iterations is denoted by N and at iteration k, after committing to decision $\xi(k)$, a previously unknown convex cost function f_k is revealed, and a loss $f_k(\xi(k))$ is incurred. The feedback available to the system are the loss $f_k(\xi(k))$ and its gradient.

The goal of the online algorithm is to ensure that the difference between the total cost and the cost of the best fixed decision is small; the best fixed decision ξ^* is chosen with the benefit of hindsight. Formally, the difference between these two cost over $k = 1, 2, ..., N$ iterations is called the regret of the online algorithm and is expressed as

$$R_N = \sum_{k=1}^{N} \left(f_k(\xi(k)) - f_k(\xi^*) \right). \tag{I.14}$$

Regret is a standard measure for the performance of learning algorithms. An algorithm performs well if its regret is sub-linear as a function of N, i.e., $\lim_{N\to\infty} R_N/N = 0$. This implies that on average, the algorithm performs as well as the best fixed strategy in hindsight independent of the adversary's actions. Further discussion on online algorithms and their regret can be found in **[138, 139, 140]**.

CHAPTER 1

Optimal Path Planning and Power Allocation for a Long Endurance Solar-Powered UAV

1. Summary

In this chapter the problem of optimal path planning and power allocation for an Unmanned Aerial Vehicle (UAV) is explored. The UAV is equipped with photovoltaic cells on top of its wings and its energy sources are solar power and rechargeable batteries. The Sun incidence angle on the photovoltaic cells, which subsequently affects energy harvesting, is determined by the attitude of the UAV and the Sun position. The desired optimal path between two given boundary points, is aimed at increasing the amount of energy storage at the final point. Meanwhile, the charging state of the battery, resulting from the power allocation, needs to be determined along with the path planning procedure. Two approaches, nonlinear programming and model reduction, are proposed and their corresponding simulation results are presented and compared.

2. Introduction

The application of renewable energy sources on aerial vehicles is an emerging technology which provides substantial benefits for Unmanned Aerial Vehicles (UAVs) fuel consumption by collecting the solar radiation energy. As sun is the major energy source for solar powered UAVs, it is necessary to employ optimal strategies to maximize energy harvesting during the flight. The energy optimized path planning problem has been studied by Klesh and Kabamba **[15, 16]** for level flight where the only source of energy is solar. In these works, the UAV flies between two given locations where the sun angles are constant during the mission, under the assumption that the flight simulation time is short. In **[17]**, periodic optimization has been used to obtain a perpetual flight, where the total energy is maximized

such that the aircraft is constrained to move on the surface of a vertical cylinder. Sachs *et al.* [**18**] have also used periodic optimization to design a trajectory that allows the UAV to fly with minimum or or even no storage equipment. In [**18**], the mission is decomposed into three phases. In the first phase, the UAV climbs to the altitude of 8.5km when the solar energy is available and in the second phase it descends during night. In the last phase, it stays at low altitude performing level flight.

Solving this type of optimal control problems is complicated due to the nonlinear kinematics of aircraft and its feasibility constraints on maneuvers. The direct and indirect optimization methods are the two general approaches proposed in solving optimal path planning problems [**1**]. The indirect method, which is based on Hamiltonian and Lagrangian equations, has low rate of convergence and is sensitive to the initial guess. Since the kinematics of the system is complex due to non-linearity, it is very difficult to obtain a proper initial guess. Another approach is direct optimization, such as collocation and nonlinear programming, which discretizes the trajectory into multiple segments and then solves the parametrized optimization problem. In addition to these approaches, the aircraft kinematics can be simplified based on the characteristics of different flight maneuvers, which is referred to as the model reduction approach in this chapter. The subsequent generated trajectories are composed of several modes, for example, Kamgarpour *et al.* in [**2**] have assumed the operating state of the aircraft in each mode to solve a hybrid optimization problem.

In this chapter, we examine both the optimal path planning and power allocation problems for a duration of one day and night of a solar powered UAV. The objective is to maximize the remaining energy stored in batteries at the end of the duration while the satisfying the boundary conditions and feasibility constraints on the states of the aircraft and batteries. Since the UAV is equipped with electronic components, i.e., photovoltaic cells, batteries and electrical engine, the power allocation problem is to efficiently allocate the amount of power to or from each component during the flight. In our work the power allocation problem is handled as part of the optimal path planning problem, where the states and controls of the flight kinematics are coupled with the amounts of allocated power.

The direct optimization method is described first to find an optimal trajectory as well as time history of power allocation for a 24-hour flying period UAV. In the second approach, the desired trajectory is divided into three phases with reduced models representing each phase and a combined analytical and numerical method is used to solve the corresponding optimization problem. The results are then compared with those obtained from the direct optimization method.

The chapter is organized as follows. The optimal control problem formulation is presented in §3 and a solution strategy is discussed. In §4 numerical results from nonlinear programming and their interpretation for a sample UAV path planning are presented. The model reduction method, including the corresponding simulation results are presented in §5 followed by concluding remarks in §6.

3. Problem Statement

The problem of optimal energy path planning and power allocation between two given locations can now be formulated as an optimization problem, whereby the final battery state of charge, $SOC(t_f)$, is used as the cost function. In addition, the state variables $[x, y, z, \psi, \gamma, V, SOC]^T$ and control variables $[n_v, n_h, T, P_{Batt}]^T$are constrained to satisfy the aircraft kinematics (I.9), battery discharge rate (I.8), and boundary conditions. The mission is to fly between specified initial and final positions during the time interval $[t_0, t_f]$. Due to limited UAV maneuverability, solar sources, and battery capacity, constraints on state and control variables are introduced. Equations (I.7) and (I.6) represent the constraints associated with the battery performance.

Based on the above description, the optimal control problem can be formulated as:

$$\max_{(n_v, n_h, T, P_{Batt})} SOC(t_f) \tag{3.1}$$

subject to

$$
\begin{aligned}
\dot{x} = V\cos\psi\cos\gamma,\ \dot{y} = V\sin\psi\cos\gamma,\ \dot{z} = V\sin\gamma \\
\dot{\psi} = \frac{g}{V}(\frac{n_h}{\cos\gamma}),\ \dot{\gamma} = \frac{g}{V}(n_v - \cos\gamma) \\
\dot{V} = \left(\frac{T-D}{mg} - \sin\gamma\right) g, \\
S\dot{O}C = -\frac{V_{OC} - \sqrt{V_{OC}^2 - 4P_{Batt}R}}{2QR}
\end{aligned} \tag{3.2}
$$

and

$$
\begin{aligned}
|\psi| \le 2\pi,\ |\dot{\psi}| \le \dot{\Psi},\ |\gamma| \le \frac{\pi}{2},\ V_{\text{stall}} \le V \le V_{\max} \\
0.25 \le SOC \le 0.9,\ 0 \le T \le T_{\max} \\
|n_v| \le n_{v_{\max}}, \quad |n_h| \le n_{h_{\max}} \\
P_{Batt_0,Charge} \le P_{Batt} \le P_{Batt_0,Discharge} \\
P_{Tot} = P_{Sun} + P_{Batt} - P_{Eng} \ge 0,
\end{aligned} \tag{3.3}
$$

with boundary conditions

$$
\begin{aligned}
[x(t_0), y(t_0), z(t_0), \psi(t_0), \gamma(t_0), V(t_0), SOC(t_0)]^T = \\
[x_0, y_0, z_0, \psi_0, \gamma_0, V_0, SOC_0]^T \\
[x(t_f), y(t_f), z(t_f)]^T = [x_f, y_f, z_f]^T,
\end{aligned} \tag{3.4}
$$

where V_{stall} and $V_{\max}$ are the stall and maximum feasible velocity of the UAV, respectively and the heading angle has to satisfy $|\psi| \le 2\pi$ based on its physical definition. Furthermore, $|\dot{\psi}| \le \dot{\Psi}$ is required to generate a smooth trajectory where $\dot{\Psi}$ is the maximum rate of change of the heading angle. Powered-flight requires the total power, P_{Tot}, to be positive, which means the power consumption is less than or equal to power generated by solar cells and/or the battery pack. This constraint is expressed in the last equation of (3.3).

4. Nonlinear Programming Approach

One simulation example, assumed to be performed on April 1st, 2012, in San Diego, CA is illustrated in Figures. 5.1 to 5.7 by solid lines and its corresponding boundary conditions are

$$[x_0, y_0, z_0] = [0, 0, 0]\,(\text{km}),\ [x_f, y_f, z_f] = [0, 0, 0](\text{km})$$
$$[\psi_0, \gamma_0] = [0, 0](\text{rad}),\ V_0 = V_{\min},\ SOC_0 = 0.9. \tag{4.1}$$

SNOPT [**34**] can be used to solve this problem. Since the simulation time of 24-hour is quite long (86400 seconds), some of the state and control variables need to be scaled to make the solution converge faster. This also avoids the singularities that may occur in estimating the underlying Hessian matrices. The path to be optimized is discretized into 24 nodes representing each hour and the time span is from 1am to midnight.

The time history of altitude in Figure 5.2 shows that the UAV flies at low level flight ($z \geq$ 100m) during the early hours of the day where the minimum power required to maintain the powered flight is optimum. The aircraft stays at low altitude until 9 a.m., then gradually climbs to the summit when the sun is rising, and then descends to its initial altitude. A feature of the optimal output is that part of the solar energy is saved as potential energy during the day which is transformed to kinematic energy as the UAV glides down during night.

The projection of the 3D trajectory on the $\{x - y\}$ plane in Figure 5.2 is relatively smooth based on the fact that the time step in optimization algorithm is 1 hour. Figure 5.3 shows the UAV velocity during the flight implying that the aircraft flies at two specific velocity magnitudes at most of the time with small scale fluctuation. Figure 5.4 shows the smooth transitions of attitude angles between discrete points. The flight path angle γ, shown in Figure 5.5, is approximately zero during the early hours of the mission, implying the low altitude level flight during those hours. The flight path angle increases while the aircraft is ascending during the day and eventually reaches a constant negative value during the night, indicating a gradual descend.

In addition to the optimal trajectory and UAV attitude presented above, the optimal power allocation strategy for the periodic mission is shown in Figure 5.7. The amount of power at the early hours of the day is approximately equivalent to the minimum power required at the low altitude flight. Around noon, the input power from the sun is large enough to compensate for both recharging the battery and providing sufficient power to ascend to higher altitude. Around 7pm, the engine is shut down ($T = 0$N) and the battery is fully charged. It is important to note that the required engine power P_{Eng} is zero at night and the total power P_{Tot} in (3.3) is non-negative at during the flying period.

Figure 5.8 indicates the time history of SOC status from the optimal power allocation output. Consistent with our presupposition, the UAV uses the battery power as a sole source of energy during night and recharge it in the day. The output indicates that the rechargeable battery pack reaches its maximum allowed capacity at the end of the mission, implying that the battery returns to its initial state of charge. Therefore, this is a global optimum solution for the problem and the result of SOC allows the UAV to resume its operation in the next day.

5. Model Reduction

Based on the numerical results, the one-day mission can be divided into three phases starting from 1 a.m.: low altitude level flight, climb, followed by a descend flight. As discussed in section 4, the UAV flies at low altitude level to minimize the engine power consumption and start climbing to gain potential energy when sufficient amount of solar power is available to both charge the battery and maintain the engine operation. The UAV uses the stored potential energy to glide to the final location. In this section, the flight kinematics during each phase is represented by a first order approximation model and the duration of each phase is specified by the availability of the solar source.

5.1. Phase 1 (Low Altitude Level Flight). The first phase starts at the initial time (1 a.m.) and lasts until sun rise. A common property of all phases is the bank angle profile. Since the time span is large and the aircraft altitude change is small, we can assume the steady flight condition for all three phases.

PROPOSITION 1. *The minimum power consumption for steady flight with a given time span is when* $\phi = 0$,$\gamma = 0$, *and* $V_{\min} = \left(\frac{4KW^2}{3\rho^2 S^2 C D_0}\right)^{1/4}$.

PROOF. At steady flight the altitude change is constant or even zero in level flight, therefore the vertical acceleration of the aircraft is zero. According to Newton's law ($L\cos(\phi)\cos(\gamma) - W = 0$) and from (I.11) the power consumption can be expressed as

$$P_{Eng} = \frac{1}{2}\frac{\rho S V^3}{\eta_{prop}}\left(C_{D0} + K\frac{W^2}{\left(\frac{1}{2}\rho V^2 S\right)^2 cos^2\phi cos^2\gamma}\right). \tag{5.1}$$

Therefore, P_{Eng} is always positive and if we minimize it over ϕ and γ the optimal solution is obtained when $\cos^2\phi^* = 1$ and $\cos^2\gamma^* = 1$. Since $\mid \gamma \mid \leq \frac{\pi}{2}$ and $\mid \phi \mid \leq \frac{\pi}{2}$, the optimal solution is at $\phi^* = 0$, $\gamma^* = 0$ and the power consumption can be rewritten as

$$P_{Eng} = \frac{1}{2}\frac{\rho S V^3}{\eta_{prop}}\left(C_{D0} + K\frac{W^2}{\left(\frac{1}{2}\rho V^2 S\right)^2}\right). \tag{5.2}$$

If the UAV flies at this constant velocity, the minimum required power provided from the battery to maintain the mission is constant during this phase and equal to the minimum engine power consumption. Therefore, with the information on the low altitude air density we can determine the velocity at which (5.2) is minimized, referred as $V_{\min}$ which is expressed as

$$V_{\min} = \left(\frac{4KW^2}{3\rho^2 S^2 C D_0}\right)^{1/4}. \tag{5.3}$$

□

This proposition applies to all three phases. Thus the desired bank angle and flight path angle during the entire mission is zero. This property simplifies the flight kinematics and solar power harvesting equation. Based on (I.8) and specified time span for this phase, $SOC(t_1)$ can be computed. The time instant t_1 refers to sunrise and the final time of low altitude level flight phase. Using the aforementioned assumptions, the aircraft kinematics in this phase is simplified as $\dot{x} = V_{\min}\cos\psi$, $\dot{y} = V_{\min}\sin\psi$and the the thrust from the engine is determined by $T = \frac{P_{Batt}\eta_{prop}}{V_{\min}}$, where P_{Batt} is equal the minimum required power from the engine flying at the given altitude.

5.2. Phase 2 (Climb). The second phase starts right after the first phase and ends at sunset. In order to maximize the aircraft mechanical energy, it is desirable for the UAV to reach its ceiling which is assumed to be 8km in this chapter. Rapid changes in altitude are not desirable in order to maintain a smooth trajectory. Therefore, if we discretize the time horizon into intervals with one hour step size, the vertical acceleration of the aircraft and flight path angle are very small. In the simplified model, we assume that the vertical displacements at all intervals are the same and equal to Δz which specifies the rate of climb ($z_r = \frac{\Delta z}{\Delta t} \approx \dot{z}$). From (I.9), the aircraft kinematics is reduced to

$$\begin{aligned} \dot{x} = V\cos\psi,\ \dot{y} = V\sin\psi,\ \dot{z} = V\gamma \\ \dot{V} = \frac{T-D}{m} - \gamma g. \end{aligned} \tag{5.4}$$

At the beginning of this phase, the battery is almost discharged. Our objective here is to minimize the value of P_{Batt} during the second phase to achieve the maximum recharging rate from the solar source. The P_{Batt} is assumed to be negative when being recharged.

Proposition 2. *The maximum available power to recharge the battery while maintaining powered flight is the solution of the following optimal control problem with the constraint* $\cos(a-\psi) = -1$*:*

$$\begin{aligned} &\min_{V,T} \int_{t_1}^{t_2} \left(\frac{TV}{\eta_{prop}} - \eta_{sol} P_{sd} S \cos e \frac{z_r}{V} \right) \\ &\text{s.t.} \quad \dot{V} = \frac{T-D}{m} - \frac{z_r}{V} g, \quad V_{stall} \le V \le V_{max}, \end{aligned} \tag{5.5}$$

where

$$\begin{aligned} T - D = \frac{VW}{z_r} - \frac{\rho S V^3}{4K z_r^2} (V - \\ \sqrt{V^2 - 4KC_{D0}z_r^2 - \frac{8KSz_r}{\rho SV^2}(WV - z_r T)}). \end{aligned} \tag{5.6}$$

PROOF. From the last equation of (3.3), the minimum power provided from the battery to maintain powered flight occurs when $P_{Tot} = 0$, P_{Eng} is minimum, and P_{Sun} is maximum. Based on the fact that the bank angle is zero and flight path angle is very small, P_{Sun} is reduced to

$$P_{Sun} = \eta_{sol} P_{sd} S(\sin e - \gamma \cos e \cos(a - \psi)). \tag{5.7}$$

In the climb flight, the path angle γ is always positive, hence the heading angle ψ is specified to satisfy $\cos(a-\psi) = -1$ during the climbing phase to maximize the solar power harvesting. Hence (5.7) is expressed as $P_{Sun} = \eta_{sol} P_{sd} S(\sin e + \gamma \cos e)$. The sun elevation angle e is specified and thus we can ignore the term $\sin e$ in the cost function. Meanwhile the flight path angle is determined by $\gamma = \frac{\dot{z}_r}{V}$, assuming that z_r is predetermined. Therefore, the minimization problem of P_{Batt} will reduce to (5.5). The vertical acceleration of the aircraft is zero, thus based on Newton's law, the lift force is expressed as $L = mg - (T - D)\frac{\dot{z}_r}{V}$. Therefore, the $T - D$ term can be derived from (I.11). □

By substituting $T - D$ in (5.5), we can solve the minimization problem and find the optimal V and T during the climb. Then one can proceed to calculate the flight path angle and subsequently the altitude change.

5.3. Phase 3 (Glide). In the third phase, the solar power is not available, however, the UAV can use its potential energy to glide through the night. The ideal scenario is that the electric engine is turned off during the entire phase and the value of SOC is constant. To avoid rapid changes in altitude, the flight path angle is assumed to be zero and the bank angle is zero for this phase as well. The aircraft kinematics at this phase is expressed as in (5.4), where $T = 0$ and $P_{Batt} = 0$. To further simplify the aircraft model without loss of generality, the flight path angle is assumed to be a constant negative number, which is constrained by the highest altitude of the UAV.

5.4. Solution Strategy. In order to compare the the results from the reduced model with the direct collocation method, the time span is discretized into 24 nodes with one hour interval for two adjacent nodes. A feasible solution with maximum $SOC(t_f)$ is guaranteed

using the simplified model and its corresponding trajectory properties are discussed in the above sections. Therefore, the solution strategy will concentrate on satisfying the path constraints.

5.4.1. *Path Planning Problem.* A path satisfying periodical boundary conditions can guarantee the perpetual flight. We find the UAV trajectory by integrating backward the states of kinematics starting from the final point which is also the initial point for the periodical flight. Since the UAV fly at low altitude level flight after the third phase, we change the z component of periodic boundary conditions to a low altitude of 100m. Flying at this altitude reduce the risk of collision with trees, buildings, and earth terrains. Without lack of generality, an arbitrary function is considered for heading angle which satisfies $\psi(t_2) = a(t_2) - \pi$ and $\psi(t_f) = \psi_0$ in the third phase. The results from phase 3 will specify the state of UAV at time t_2. Given the state of UAV at time t_2 and trajectory characteristics introduced in subsection 5.2, the aircraft kinematics (5.4) is integrated backward using Euler's method. Based on the result from second phase, the path planning problem in phase 1 is expressed as a two point boundary value problem where the boundary conditions are specified as

$$\begin{aligned} [x(t_0), y(t_0), \psi(t_0)]^T &= [x_0, y_0, \psi_0]^T \\ [x(t_1), y(t_1), \psi(t_1)]^T &= [x_1, y_1, a(t_1) - \pi]^T. \end{aligned} \tag{5.8}$$

This problem can be solved using a numerical approach such as shooting method. In this part, the algorithm runs feasibility evaluation to make sure that the UAV can reach the boundary condition with $V_{\min}$ during the specified time interval.

5.4.2. *Power Allocation Problem.* In the power allocation problem, (I.8) is integrated forward. In the first phase P_{Batt} and engine thrust are specified and there is no solar power available. Therefore, the SOC at time t_1 can be specified. The T and V in second phase are calculated by solving the optimal control problem (5.5). Given the constraint in phase 2, the solar power is expressed as $P_{Sun} = \eta_{sol} P_{sd} S(\sin e + \gamma \cos e)$, and thus P_{Batt} can be written as $P_{Batt} = \frac{TV}{\eta_{prop}} - \eta_{sol} P_{sd} S(\sin e + \gamma \cos e)$. From (I.8), the $SOC(t_2)$ is specified.

Since the electric engine is turned off, the SOC does not change during phase 3 and it will maintain its level for the rest of the mission which is equivalent to $SOC(t_2)$.

5.5. Numerical Results using the Simplified Model. In this section, the numerical results obtained from model reduction method are presented by marked lines and compared with the results obtained from NLP method in Figures. 5.1 to 5.8. The the UAV trajectory in the $\{x, y\}$ plane, as illustrated in Figure 5.2, is completely different from original results due to the arbitrary selection of heading angle in the last phase as reflected in Figure 5.4. Unlike the velocity profile obtained using original model, the velocity profile of UAV in the simplified model varies smoothly. However, the aircraft altitude and flight path angle in the first and last phases are almost identical as shown in Figures 5.1 and 5.4.

This analysis indicates that a more complicated maneuver other than the steady climb is required for both models to match on their altitude change and flight path angle profile in the second phase. However, according to the time history of SOC and power output in Figures 5.7 and 5.8, the $SOC(t_f)$ is identical in both methods. Therefore, we can make the conclusion that there are many optimal paths leading to the maximum power storage for a battery pack with a favorable storage capacity.

6. Remarks

The main contribution of this chapter is the investigation of optimal path planning and energy allocation problem for solar-powered UAVs in three dimensional space. The solutions using the direct optimization approach provides complete information of the trajectory, attitude, and battery status of the UAV whose kinematics and electrical models are integrated when solving the optimization problem. In addition, it was revealed from the simulation results that the UAV mission can be divided into three phases, level flight, climb, and glide. The corresponding maneuvers are then represented by simplified aircraft kinematics to improve the efficiency of the numerical and analytical solutions. The reduced model is shown to yield similar power allocation strategy and optimal cost as the one obtained from the original model. We used a desktop computer with quad core processor (3.4 GHz) and the computation time for the direct optimization method and simplified

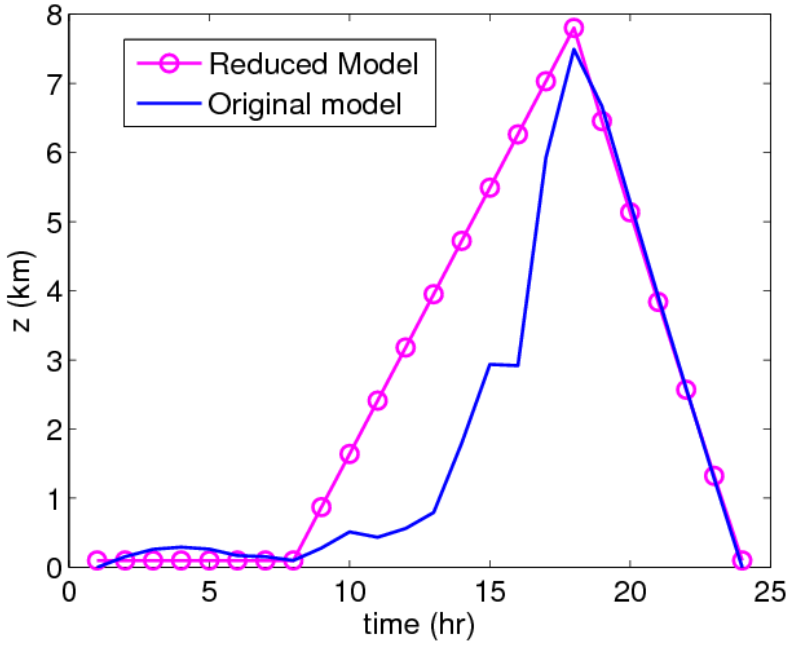


FIGURE 5.1. The altitude of UAV as a function of time.

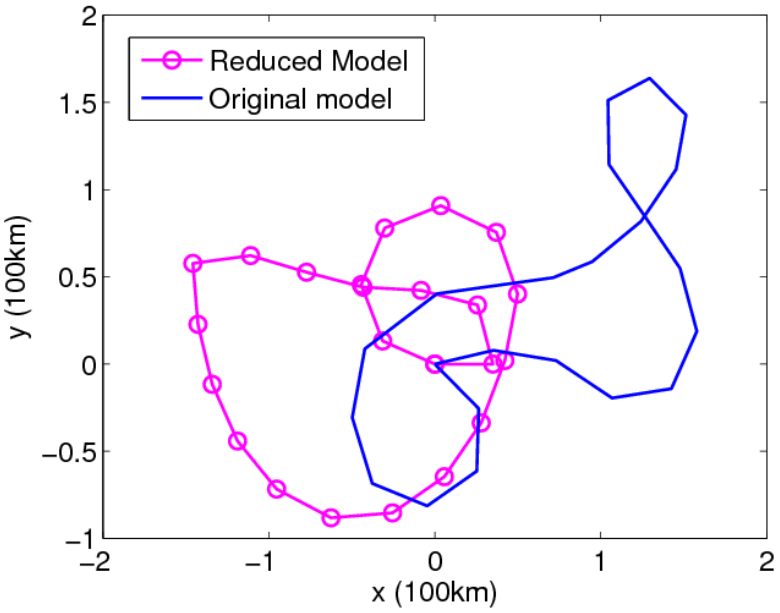


FIGURE 5.2. The optimal trajectory is shown in the $\{x - y\}$ plane.

model are 393.417*sec* and 1.758*sec*, respectively. This indicates that both methods are executable on a real time operating system, and that the computation speed using reduced model is much faster.

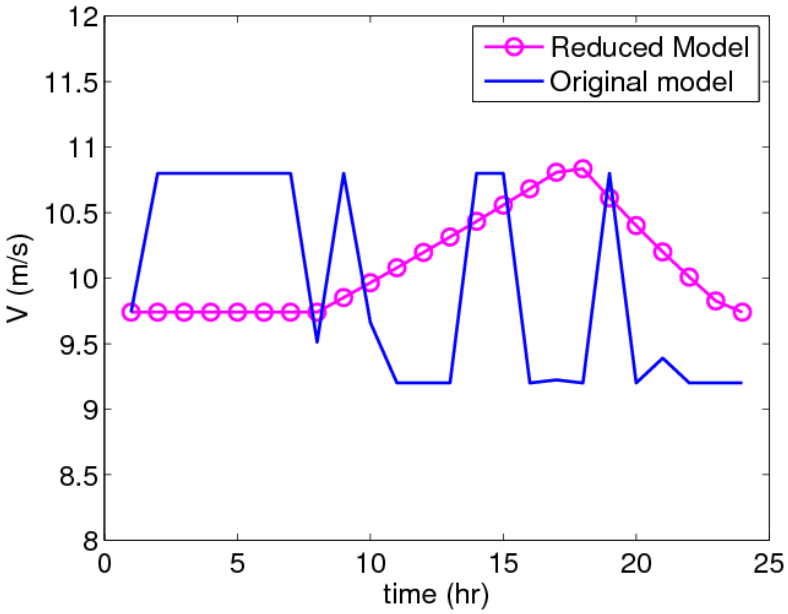


FIGURE 5.3. The UAV velocity during the trajectory.

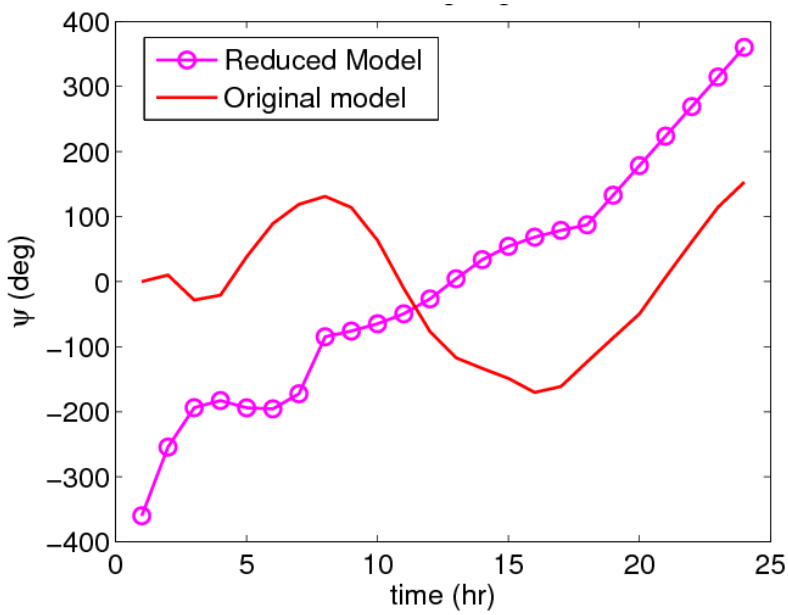


FIGURE 5.4. The UAV heading angle ψ.

The future research will focus on different cost functions such as the weight of the battery pack and the flight range. In addition, other types of missions will be considered, imposing additional constraints on UAV path planing. For example, surveillance and reconnaissance missions require the UAV to meet specific way points along its trajectory.

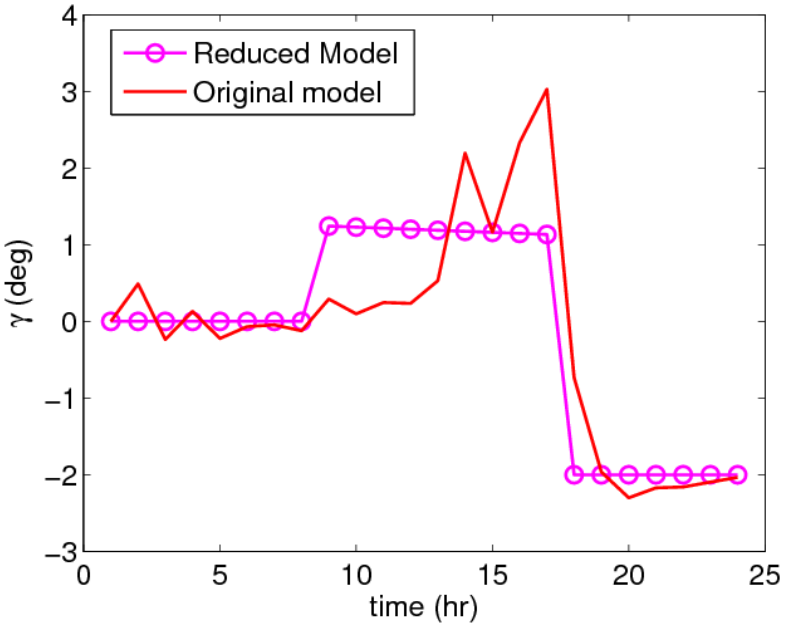


FIGURE 5.5. The UAV flight path angle γ.

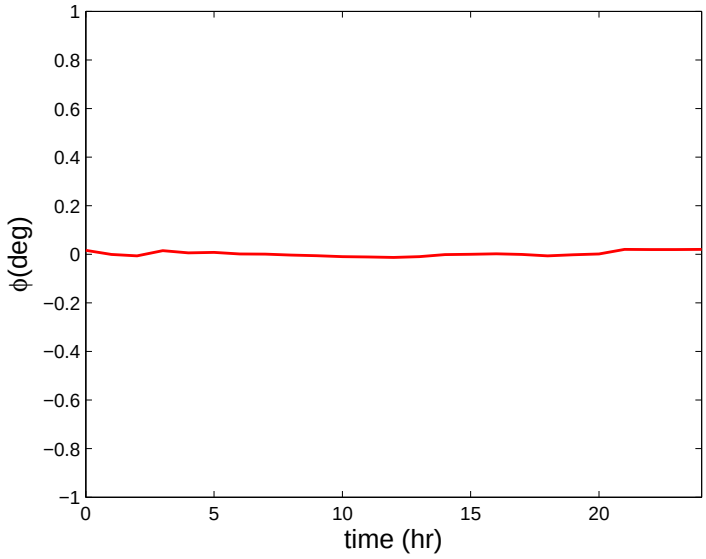


FIGURE 5.6. The UAV flight bank angle ϕ.

7. Appendix

The aircraft and battery pack parameters for a sample UAV which is used in numerical simulation are:

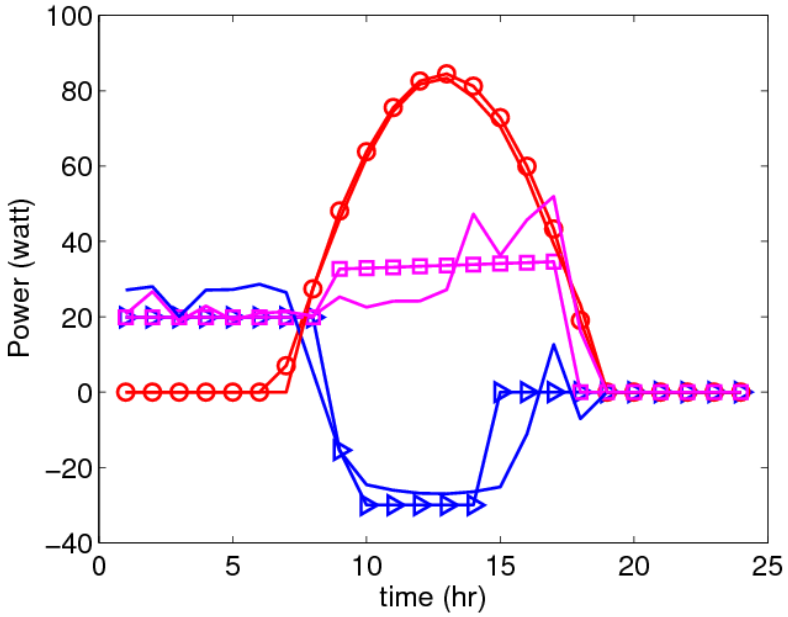


FIGURE 5.7. The optimal power allocation during the optimal trajectory.

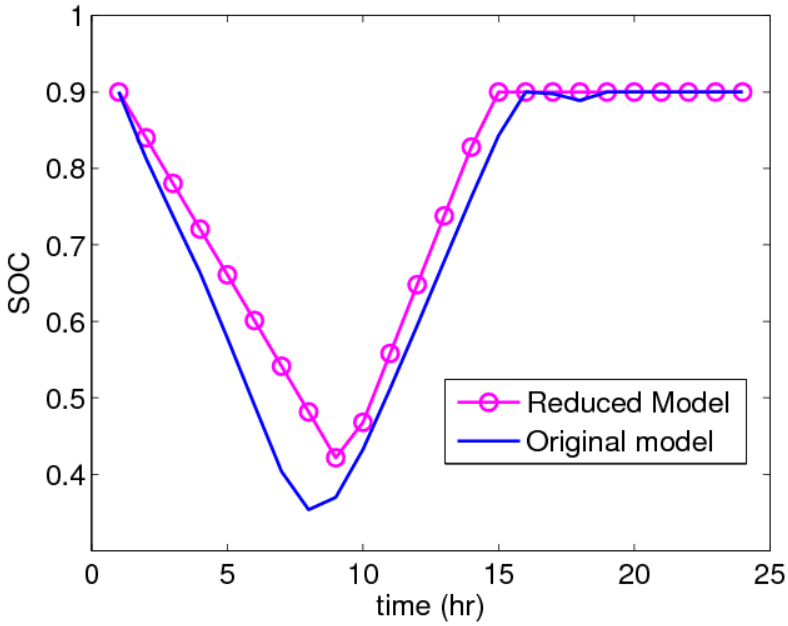


FIGURE 5.8. The *SOC* of the battery pack.

TABLE 1. Sample UAV parameters

η_{sol}	$P_{sd}(\frac{W}{m^2})$	$b(m)$	$S(m^2)$	$m(kg)$	η_e	η_{prop}
0.19	886	3.1	0.56	4.201	0.992	0.7

TABLE 2. Battery pack (Lithium-Sulfur) parameters

Q	V_{OC}	R	m_{Batt}	$P_{Batt_0,Discharge/Charge}$
26.4 Ah	12.6 V	0.0125 Ω	1.280 kg	$\pm$30 watt

TABLE 3. Sensor parameters

θ	$h_{\max}$	h_0
30 deg	500 m	750 m

CHAPTER 2

Energy Aware Aerial Surveillance for a Long Endurance Solar-Powered UAV

1. Summary

In this chapter, energy optimal surveillance trajectories for Unmanned Aerial Vehicles (UAV) are explored. The main objective is to have maximum sensor coverage range while maintaining a perpetual flight in the presence of uncertainties. A solar powered UAV is equipped with photovoltaic cells mounted on its wings and rechargeable batteries. The photovoltaic cells generate solar energy based on the position of the sun, attitude of the UAV, and sky clarity. The vehicle aims to optimize the energy storage in the batteries and coverage during the day while the availability of solar radiation is uncertain and the sensor resolution diminishes due to altitude gain. A model for optimal coverage, path planning, and power allocation in a solar powered UAV is proposed and the corresponding simulation results are presented. In addition, the effect of maximum altitude gain on the energy storage is studied based on a reduced hybrid model. An online setting is proposed to represent the solar radiation uncertainties. This approach demonstrates convergence to the best fixed strategy in both theory and simulation results.

2. Introduction

Long endurance Unmanned Aerial Vehicles (UAVs) have a wide range of applications including crop and forestry fire monitoring, search and rescue, remote sensing, mapping, and atmospheric sensing missions. One way to extend the duration of flight in UAVs is to change the configuration of the aircraft, i.e., its wings **[3]** or propulsion system. In this context, hybrid propulsion systems have been proposed **[4, 5, 6]** which is composed of several power sources on-board such as solar cells, fuel cells, and batteries. Taking

advantage of the environmental energy sources in the form of wind gusts and thermals is another approach to increase the flight duration. There is an extensive literature on path planning and trajectory optimization for soaring flight [**7, 8, 9, 10, 11, 12, 13**]. In addition, photovoltaic cells are a favorable candidate to collect energy from the environment and power the UAVs [**14**]. In pursuit of this goal, several solar powered UAVs have been built and flown since 1974. The latest solar powered UAV, Aquila developed by Facebook, will be able to circle a remote region for up to 90 days, providing internet connectivity to people from an altitude of 60,000 to 90,000 feet.

The major energy source for solar powered UAVs is the sun and it is thus necessary to employ optimal strategies to maximize energy harvesting that leads to higher flight endurance. Energy optimal path planning problem in 2D space has been studied for solar powered UAVs where the only source of energy is the solar radiation. In [**15, 16**], the solar powered UAV flies between two given locations in a level flight under the assumption that the flight simulation time is short and the sun incident angle over the wings is constant. Moreover, in [**17**], Spangelo *et al.* have proposed a periodic optimization method to obtain perpetual flight, where the total energy is maximized, constraining the aircraft to move on the surface of a vertical cylinder. Exploiting the potential energy and on-board electrical storage units, Sachs *et al.* [**18**] have proposed a trajectory that allows the solar powered UAV to fly with minimum or no solar energy stored in the batteries. The aforementioned trajectory is composed of three phases. The UAV climbs to the altitude of 8.5 km in the first phase, when the solar energy is available. In the second phase, it descends during night and finally, the UAV performs low altitude level flight during the third phase. Moreover, in our previous work [**19**], we have investigated the optimal path planning and energy allocation problem for a single UAV in 3D space using model reduction techniques. The objective in [**19**] was to maximize the final state of charge of the battery pack, representing the energy stored on-board.

Enabled with long endurance UAVs, the surveillance problem in aerial vehicles has become feasible and is now an active research topic [**20, 21, 22, 23, 24**]. However, the energy efficiency aspect of this problem has not been addressed at large by the scientific

community. Mei *et al.* [**25**] proposed an efficient algorithm for the deployment of multiple mobile robots that take into account energy and timing constraints, and explores the surveillance problem for ground robots. Nigam *et al.* have also considered an endurance constraint in the control of multiple quad-rotor vehicles in order to achieve persistent surveillance [**26**]. Moreover, Sujit and Ghose [**27**] presented a search algorithm for UAVs with limited fuel. These works do not deal with energy consumption constraints directly and use a fixed flight time to represent these constraints. Moreover, in [**28**], the problem of energy collection using multiple solar powered UAVs in 2D space has been explored. In this problem setup, the UAVs are required to carry on continuous coverage of a target area in which harvesting solar energy would potentially eliminate the need for a refueling policy.

Our first contribution is posing the energy optimal surveillance as an optimal control problem where the solar powered UAV employs optimal strategies to maximize both energy harvesting and sensor coverage range during the flight. In this chapter, we examine the optimal energy and coverage surveillance for a duration of 24 hours of a single solar powered UAV in 3D space. In pursuit of perpetual flight, the secondary objective of the vehicle is to maximize the total energy stored in batteries at the end of the duration. During the mission, the aircraft must also comply with maneuverability constraints presented as boundary conditions and feasibility constraints on its states. A direct optimization method is used to solve the optimal energy surveillance problem and the results are compared with the optimal coverage surveillance problem. We present an approach for assessing the effect of altitude gain on energy storage based on a reduced hybrid model proposed in our previous work [**19**].

Both nonlinear programming-based approaches and the reduced problem solutions provide complete information on the trajectory and power allocation strategy of the UAV under the assumption that the solar radiation is predictable. However, the actual amount of solar radiation that reaches a particular point in the earth's atmosphere is extremely variable. In addition to the regular daily and yearly variation due to the apparent motion of the sun, irregular variations are caused by local atmospheric conditions, such as clouds.

Therefore, the available solar energy is unknown a priori without detailed probabilistic characteristics. As our second contribution, we model the energy aware surveillance problem in an online framework where the uncertainty in the system is demonstrated by an arbitrarily time varying cost function. In other words, since the solar radiation is unpredictable, the cost function associated with this optimization problem is unknown and lacking probabilistic assumption when the relevant strategy is implemented. A method to approach this type of problems is online learning and a standard metric to measure the performance of online algorithms is called *regret* **[64, 86, 141]**. Regret measures the difference between the incurred cost and the cost of the best fixed strategy in hindsight. Consequently, a good algorithm is one where the average regret approaches zero. In this chapter, we exploit Online Dual-Averaging (ODA) algorithm to estimate the optimal rate of climb during the steady climb phase while the solar spectral density is unknown ahead of time.

The contribution of this chapter is two folds. First we propose a model for the energy aware surveillance problem. Second, we extend the aforementioned problem to the online setting to account for the intermittent solar radiation. The chapter is organized as follows. In §3, the optimal control problem formulations for two different objective functions are presented and a solution strategy based on a nonlinear programming is discussed. Numerical results for a sample UAV path planning are investigated and interpreted in §4. In §5, a reduced hybrid model and heuristic analysis of maximum altitude gain are discussed. The intermittent solar radiation and its effect on altitude gain is explored in §6. Finally, §7 presents concluding remarks followed by future research directions.

3. Problem Statement

The solar powered UAV's mission is to fly between specified initial and final positions during the time interval $[t_0, t_f]$. The corresponding trajectories should demonstrate optimal coverage and energy properties. Therefore, the desired trajectories are modeled as solutions to an optimal control problem with a set of constraints. The cost function associated with

the surveillance problem is presented as the cumulative sum of sensor coverage over time:

$$C(t_f) = \int_{t_0}^{t_f} c(z)dt, \tag{3.1}$$

where $c(z)$ is the coverage range given in Eq. (I.12). Furthermore, the cost function associated with energy optimal path planning is characterized as the final state of charge $SOC(t_f)$ in the battery pack **[19]**. The state variables denoted by $[x, y, z, \psi, \gamma, V, SOC, C]^T$ and control variables $[n_v, n_h, T, P_{Batt}]^T$are constrained to satisfy the aircraft kinematics in Eq. (I.9), battery charge/discharge rate in Eq. (I.8), sensor coverage range in Eq. (I.12), and the boundary conditions.

Several constraints on state and control variables are also introduced based on limited UAV maneuverability, structural capability, photovoltaic cells characteristics, and battery capacity. More specifically, the stall and maximum feasible velocity of the UAV are denoted by V_{stall} and $V_{\max}$, respectively, which implies $V_{stall} < V \leq V_{\max}$. The rate of change of heading angle ψ has to satisfy $|\dot{\psi}| \leq \dot{\Psi}_{\max}$ in order to generate a smooth trajectory where $\dot{\Psi}_{\max}$ is the maximum changing rate of the heading angle.

Moreover, powered flight requires the total power P_{Tot} remains positive, which means the power consumption must be less than or equal to power generated by solar cells and/or the battery pack.

Based on the above problem description, the optimal energy surveillance problem can be formulated as:

$$\max_{n_v, n_h, T, P_{Batt}} C(t_f) + SOC(t_f) \tag{3.2}$$

subject to

$$\dot{x} = V\cos\psi\cos\gamma, \quad \dot{y} = V\sin\psi\cos\gamma, \quad \dot{z} = V\sin\gamma \tag{3.3a}$$

$$\dot{\psi} = \frac{g}{V}(\frac{n_h}{\cos\gamma}), \quad \dot{\gamma} = \frac{g}{V}(n_v - \cos\gamma) \tag{3.3b}$$

$$\dot{V} = \left(\frac{T-D}{mg} - \sin\gamma\right) g \tag{3.3c}$$

$$S\dot{O}C = -\frac{V_{OC} - \sqrt{V_{OC}^2 - 4P_{Batt}R}}{2QR} \tag{3.3d}$$

$$\dot{C} = c(z) \tag{3.3e}$$

and

$$||\dot{\psi}| \leq \dot{\Psi}_{\max}, \ |\gamma| \leq \frac{\pi}{2}, \tag{3.4a}$$

$$V_{stall} < V \leq V_{\max}, \tag{3.4b}$$

$$0 \leq T \leq T_{max}, \tag{3.4c}$$

$$0.25 \leq SOC \leq 0.9, \tag{3.4d}$$

$$|n_v| \leq n_{v_{\max}}, \quad |n_h| \leq n_{h_{\max}}, \tag{3.4e}$$

$$P_{Batt_0,Charge} \leq P_{Batt} \leq P_{Batt_0,Discharge} \tag{3.4f}$$

$$P_{Tot=}P_{Sun} + P_{Batt} - P_{Eng} \geq 0, \tag{3.4g}$$

with boundary conditions:

$$[x(t_0), y(t_0), z(t_0), \psi(t_0), \gamma(t_0), V(t_0), SOC(t_0)]^T = [x_0, y_0, z_0, \psi_0, \gamma_0, V_0, SOC_0]^T \tag{3.5a}$$

$$[x(t_f), y(t_f), z(t_f)]^T = [x_f, y_f, z_f]^T. \tag{3.5b}$$

The optimal control problem presented in Eq. (3.2) through Eq. (3.5) is a non-convex nonlinear optimization problem. Using a nonlinear programming approach, an approximate solution to this problem can be achieved and further discussion and simulation results are provided in the next section.

4. Simulation Results using Nonlinear Programming Approach

4.1. Nonlinear Programming Method. The general approach to solve a nonlinear optimization problem, presented in Eq. (3.2) through Eq. (3.5), is the direct collocation (DC) method. DC is an approximate method that discretizes a continuous solution and uses linear interpolation to satisfy the differential equations imposed on the states. This method transforms the optimal control problem into a nonlinear programming problem (NLPP) and the solution is in terms of infinitely many values of state and control variables.

The nonlinear programming (NLP) solver used to solve our optimization problem is based on a sequential quadratic programming (SQP) algorithm called NPSOL **[34, 35]**. NPSOL can be used to solve problems such as minimizing a performance index subject to constraints on individual state and/or control variables. Since the mission duration of 24-hour is long (86400 seconds), some of the state and control variables need to be scaled accordingly to make the solution converge faster and also avoid the singularities that may occur in estimating the underlying Hessian matrices.

The simulation results for two optimal control problems are illustrated in this section and their corresponding sensor coverage and energy storage are compared. The boundary conditions are provided in Table 1. The UAV starts from the origin with zero heading and flight path angles and returns to its initial location after 24 hours. The mission endurance is discretized into 24 nodes representing each hour during a day. This large time step simplifies the optimization problem as well as providing fast convergence and smooth trajectories.

TABLE 1. Boundary conditions

States	Values
$[x_0, y_0, z_0]$ (km)	$[0, 0, 0]$
$[\psi_0, \gamma_0]$ (rad)	$[0, 0]$
V_0 (m/s)	V_{min}
SOC_0	0.9
$[x_f, y_f, z_f]$ (km)	$[0, 0, 0]$

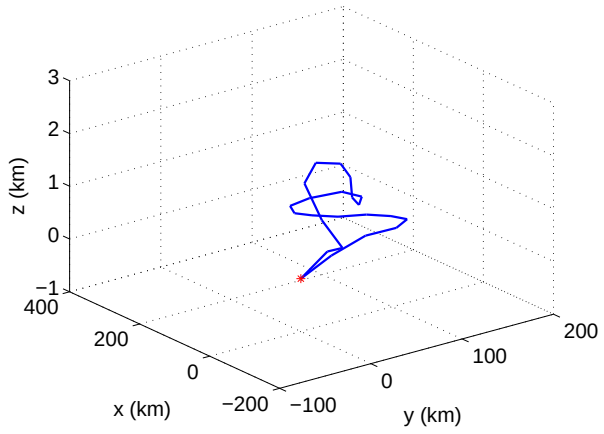
The values of UAV, battery pack, and sensor parameters, used in numerical simulations are provided in the appendix.

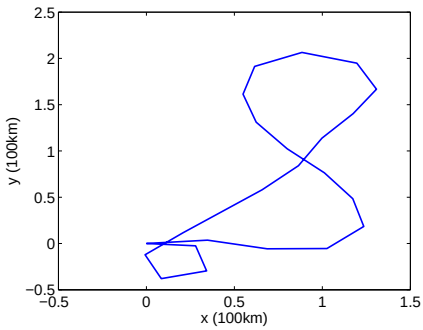
4.2. Optimal Coverage Surveillance Problem. The first optimal control problem is the optimal coverage surveillance where the objective is to

$$\max_{n_v, n_h, T, P_{Batt}} C(t_f) \tag{4.1}$$

subject to Eq. (3.3)-(3.5). Note that although the energy storage is not explicitly optimized, the UAV has to maintain enough energy for a powered flight according to the inequality (3.4g). Figure 4.1(a) and Fig. 4.1(b) present the optimal trajectory in 3D space as well as its projection on the $X - Y$ plane. The optimal trajectory is relatively smooth as the time step chosen in the algorithm is 1 hour and the heading rate of change is constrained in the optimization problem.

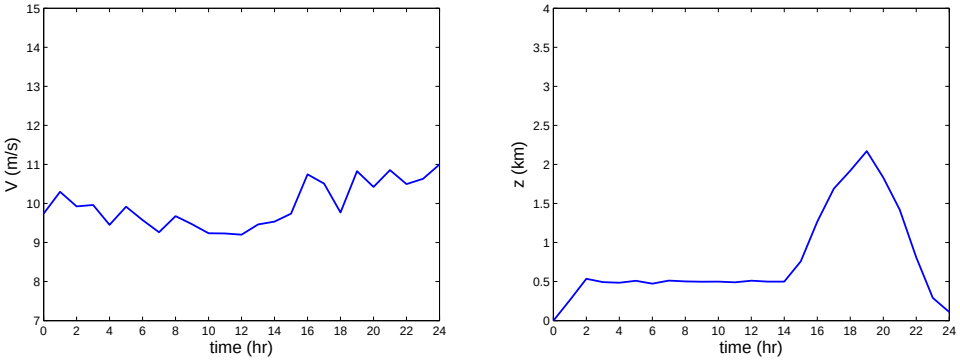
Figure 4.2(a) demonstrates small scale fluctuation around $10\, m/s$ in the he UAV velocity during the flight. Th altitude profile in Fig. 4.2(b) shows that the UAV starts at zero altitude and reaches $z \approx 500\, m$ altitude and stays at that altitude for a while. Note that $h_{\max} = 500\, m$, implying that the sensor coverage range is at its maximum value when $z = h_{\max}$. Although, the goal of the UAV is to achieve maximum sensor coverage range, it has to maintain the powered flight. Therefore, the UAV climbs up to higher altitudes ($z \approx 2\,\text{km}$) to gain potential energy and spend this energy by soaring during night. Figure 4.3 depicts the smooth transitions of attitude angles of the UAV during the solution trajectory. Fig. 4.3(a) implies that the solar energy harvesting, that is proportional to $\cos(i)$, reaches its peak at noon and is zero between sunset and sunrise. It is noted in Fig. 4.3(b), the bank angle ϕ is approximately zero during the mission which refers to the fact that the aerodynamic drag is at its local minimum when $\phi = 0$ [**19**]. In addition, the flight path angle γ shown in Fig. 4.3(c) is very small during the flight. The power allocation strategy is illustrated in Fig. 4.4 for further investigation. Figure 4.4(a) depicts the time history of the SOC and Fig. 4.4(b) represents the power allocation during the trajectory for three major power sources presented in §I.1. It is shown that the battery pack is charged during the day while the UAV uses solar radiation to supply the required power. Finally, the coverage range of the UAV sensor is presented in Fig. 4.5, which demonstrates the UAV tendency to stay at $h_{\max}$.





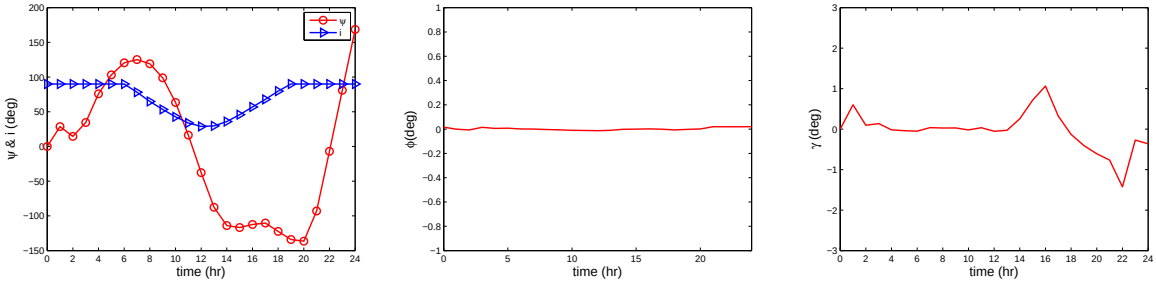
(a) (b)

FIGURE 4.1. The optimal trajectory in $(X-Y-Z)$ space (a) and the $X-Y$ plane (b) are depicted in these figures for the optimal coverage surveillance problem in Eq. (4.1).



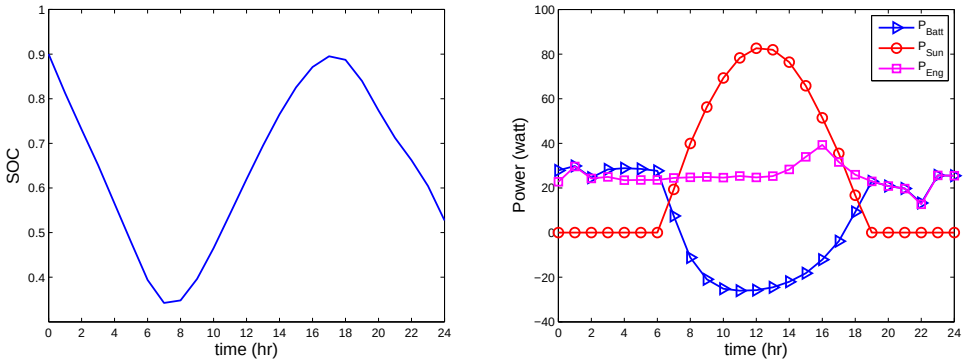
(a) (b)

FIGURE 4.2. (a) The UAV velocity during the solution trajectory, and (b) the altitude of UAV as a function of time are shown for the optimal coverage surveillance problem in Eq. (4.1).



(a) (b) (c)

FIGURE 4.3. The aircraft attitude for the optimal coverage surveillance problem in Eq. (4.1) is represented by (a) the heading angle, ψ, sun incidence angle, i, (b) bank angle, ϕ, (c) and flight path angle, γ. The time span is from 1am to midnight of April 1, 2012.



(a) (b)

FIGURE 4.4. (a) Presents the *SOC* of the battery pack and (c) shows the optimal power allocation during the optimal trajectory for the optimal coverage surveillance problem in Eq. (4.1).

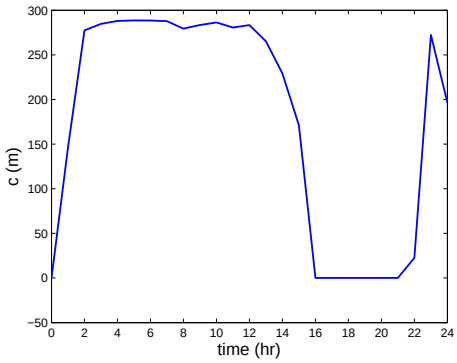


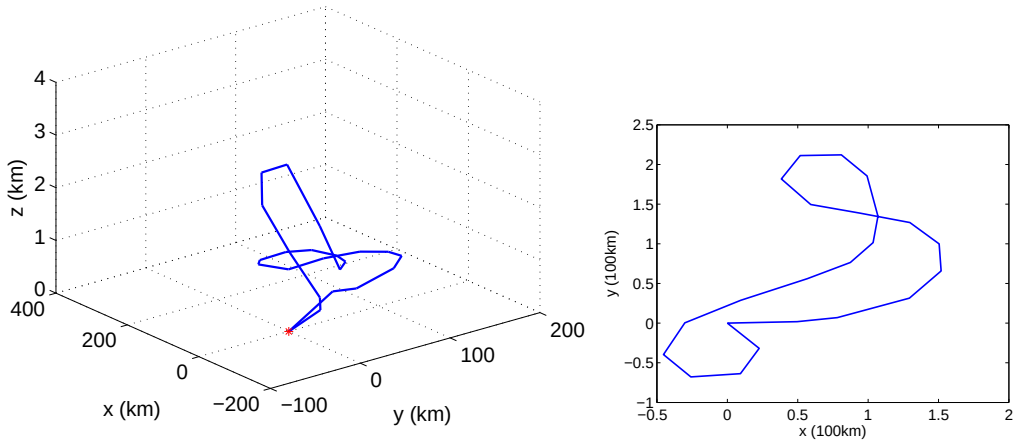
FIGURE 4.5. Sensor coverage range during the optimal trajectory for the optimal coverage surveillance problem in Eq. (4.1).

The optimal coverage cost and final SOC in this scenario are

$$C(t_f) = 16226\,\text{km}, \quad SOC(t_f) = 0.383. \tag{4.2}$$

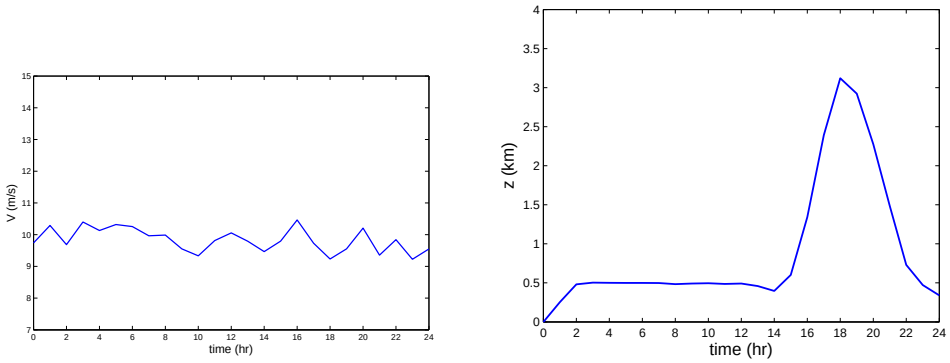
4.3. Energy Aware Surveillance Problem. In the second optimal control problem, the objective is to maximize both the sensor coverage range and the final state of the charge for the battery pack which represents the energy stored in the system. The objective function for this optimal control problem is expressed in Eq. (3.2)-(3.5). The optimal trajectory and its velocity profile shown in Figs. 4.6 and 4.7 have slightly different properties than the optimal coverage problem results presented in §4.2. However, the energy storage in the system has an important contribution to the cost function in the energy aware surveillance problem. Therefore, the UAV climbs to a higher altitude ($z \approx 3$ km) in order to harvest more potential energy. The attitude of the UAV is shown by its Euler angles in Fig. 4.8. The Euler angles have also similar profiles as in the optimal coverage surveillance problem solution presented in Fig. 4.3.

The optimal power allocation strategy is shown in Fig. 4.9. Moreover, Fig. 4.9(a) indicates the time history of the SOC suggesting that the rechargeable battery pack recovers most of its initial state of charge which allows the UAV to resume its operation the next day. The engine input power, as well as output power form the battery pack and photovoltaic cells are depicted in Fig. 4.9(b). Moreover, Fig. 4.9(b) demonstrates that the



(a) (b)

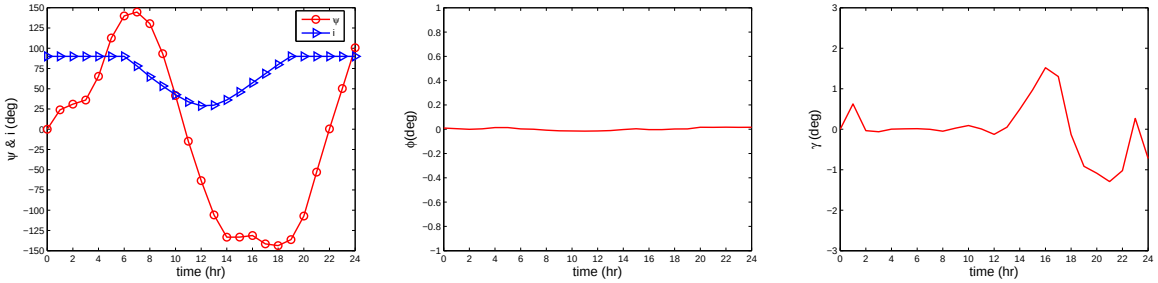
FIGURE 4.6. The optimal trajectory in $X-Y-Z$ space (a), and the $X-Y$ plane (b) are depicted in these figures for the optimal energy surveillance problem in Eq. (3.2).



(a) (b)

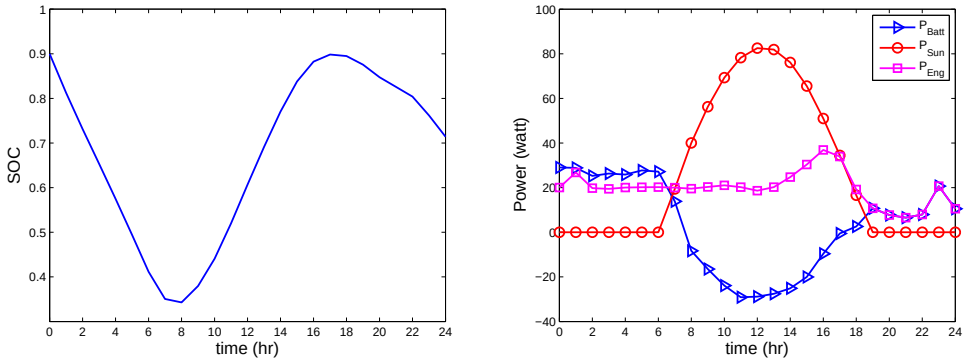
FIGURE 4.7. The UAV velocity during the solution trajectory (a), and the altitude of UAV as a function of time (b) are shown in this figure for the optimal energy surveillance problem in Eq. (3.2).

required power for electric engine is approximately at its minimum while UAV operates at low altitude level flight during the early hours of the mission. In addition, the solar power from the sun compensates for both recharging the battery and supplying enough power for maintaining powered flight during the day. Lastly, the coverage range of UAV sensor is presented in Fig. 4.10 which has approximately the same profile as in the optimal coverage surveillance result presented in Fig. 4.5.



(a) (b) (c)

FIGURE 4.8. The aircraft attitude for the optimal energy surveillance problem in Eq. (3.2) is represented by (a) heading angle, ψ, sun incidence angle, i, (b) bank angle, ϕ, (c) and flight path angle, γ. The time span is from 1am to midnight of April 1, 2012.



(a) (b)

FIGURE 4.9. Figure (a) presents the *SOC* of the battery pack and figure (b) shows the optimal power allocation during the optimal trajectory for the optimal energy surveillance problem in Eq. (3.2).

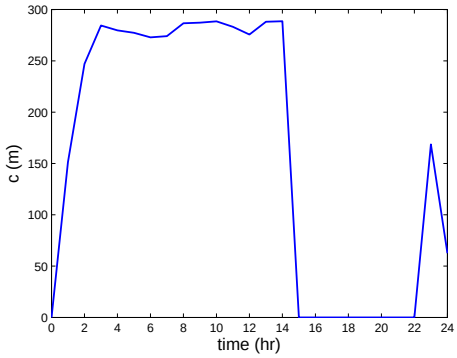


FIGURE 4.10. Sensor coverage range during the optimal trajectory for the optimal energy surveillance problem in Eq. (3.2).

The optimal coverage cost and final SOC in this scenario are

$$C(t_f) = 16372\,\text{km}, \quad SOC(t_f) = 0.706. \tag{4.3}$$

From Eqs. (4.2) and (4.3) it is noted that the coverage cost function $C(t_f)$ is slightly less than its value for the energy optimal surveillance problem. Moreover, the final SOC of the battery pack is improved by 84% which implies that the UAV can achieve a local optimal coverage while optimizing its energy storage. This property can also be used to design perpetual surveillance mission. Figures 4.2 and 4.7 show that in both scenarios the solar powered UAV maintains level flight at an altitude of $500\,m$ for roughly 15 hours. Based on the coverage function expressed in Eq. I.12, the sensor coverage is maximum at this altitude. Therefore, as it is shown in Figs. 4.5 and 4.10, the duration of this level flight plays an important role in maximizing the surveillance coverage.

In addition, the effect of potential energy on optimal energy storage and coverage can be deduced from Figs. 4.2(b) and 4.7(b) and will be further studied next.

5. Analysis of Maximum Altitude Gain

Based on the numerical results, the effect of potential energy on optimal energy storage and coverage is studied in this section. The SOC profile for $t \in [1\,\text{am}, 3\,\text{pm}]$ is similar in Figs. 4.4 and 4.9. Thus the amount of energy stored in the battery pack depends on the trajectory of the solar powered UAV after $t = 3\,\text{pm}$ given that the rate of climb and descend are different in the results presented in Figs. 4.2 and 4.7. In other words, the maximum altitude gain or potential energy is proportional to the amount of energy stored in the battery pack. In order to study the effect of potential energy on optimal energy storage, a reduced hybrid model introduced in our previous work **[19]** is considered. In reduce hybrid model, the optimal trajectory of UAV is presented with three simplified modes. These modes are level flight, steady climb, and steady glide, consistent with the trajectories presented in Figs. 4.1-4.9. A detailed explanation of the reduced hybrid model is provided in the next subsection.

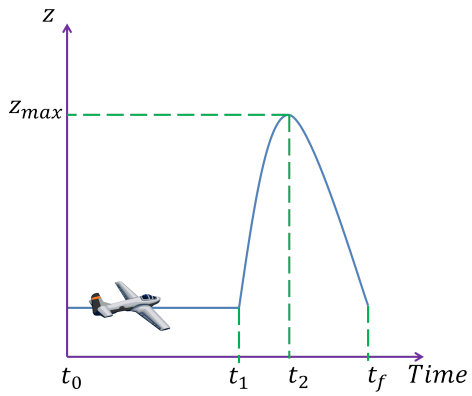


FIGURE 5.1. Generic altitude profile for a reduced hybrid model.

5.1. Reduced Hybrid Model. In this subsection, the assumptions and simplified dynamics are presented for each mode. A generic altitude profile for the reduced hybrid model is depicted in Fig. 5.1 where $z_{\max}$ denotes the maximum altitude gain during the trajectory.

5.1.1. *Mode 1 (Level Flight).* The first phase is initiated at time, $t_0 = 1\,\text{am}$, and lasts until $t_1 = 3\,\text{pm}$. Proposition 1 in [**19**], has demonstrated that the minimum power consumption for steady level flight within a given time span occurs when $\phi = 0, \gamma = 0$, and

$$V_{min} = \left(\frac{4KW^2}{3\rho^2 S^2 C D_0}\right)^{1/4}. \tag{5.1}$$

These properties simplify the flight kinematics and solar energy harvesting equation. The aircraft kinematics for this phase is thus reduced to

$$\dot{x} = V_{min}\cos\psi \tag{5.2a}$$

$$\dot{y} = V_{min}\sin\psi. \tag{5.2b}$$

In addition, given the time span for this phase, $SOC(t_1)$ can be computed based on Eq. (I.8), where time instant t_1 refers to the final time of low altitude level flight phase. In order to preserve the SOC of battery, P_{Batt} is equal to the minimum required power from

the engine flying at the given altitude:

$$P_{Eng} = \frac{1}{2}\frac{\rho S V^3}{\eta_{prop}}\left(C_{D0} + K\frac{W^2}{\left(\frac{1}{2}\rho V^2 S\right)^2}\right), \tag{5.3}$$

and based on Eq. I.1, the engine thrust is determined by

$$T = \frac{P_{Batt}\eta_{prop}}{V_{min}}. \tag{5.4}$$

5.1.2. *Mode 2 (Steady Climb).* The second phase starts right after the first phase and as long as the solar radiation energy is available to maintain powered flight, the UAV climbs to gain potential energy. For simplicity and obtaining a smooth trajectory, the time horizon is descretized into k intervals of length Δt and the rate of climb is kept constant. In other words, the vertical displacements during all time intervals Δt are the same and equal to Δz. Therefore, the rate of climb can be defined as

$$z_c = \frac{\Delta z}{\Delta t} > 0, \tag{5.5}$$

and the maximum altitude is given by

$$z_{\max} = z(t_1) + k\Delta t z_c, \tag{5.6}$$

where $z(t_1) = h_{\max}$ as defined in Eq. (I.12). In addition, the flight path angle is small and set as $\phi = 0$. Therefore, based on Eq. (I.9), the aircraft kinematics is reduced to

$$\dot{x} = V\cos\psi,\ \dot{y} = V\sin\psi,\ \dot{z} = V\gamma \tag{5.7a}$$

$$\dot{V} = \frac{T - D}{m} - \gamma g. \tag{5.7b}$$

The objective is to minimize the value of P_{Batt} during the second phase to achieve the maximum recharging rate and solar energy harvest. Proposition 2 in **[19]** states that we need to solve the following optimal control problem to find the maximum available power

to recharge the battery:

$$\min_{V,T} \int_{t_1}^{t_2} \left(\frac{TV}{\eta_{prop}} - \eta_{sol} P_{sd} S \cos e \frac{z_c}{V} \right) dt \tag{5.8}$$

$$\text{s.t.} \quad \dot{V} = \frac{T-D}{m} - \frac{z_c}{V} g, \quad V_{stall} \leq V \leq V_{\max}, \tag{5.9}$$

$$\cos(a - \psi) = -1$$

where

$$z_c = \frac{z_{\max} - z(t_1)}{k \Delta t},$$

$$T - D = \frac{VW}{z_c} - \frac{\rho S V^3}{4 K z_c^2} \left(V + \sqrt{V^2 - 4 K C_{D0} z_c^2 - \frac{8 K S z_c}{\rho S V^2} (WV - z_c T)} \right), \tag{5.10}$$

and time instant t_2 refers to the sunset. By substituting $T - D$ in Eq. (5.8), we can solve the minimization problem and find the optimal velocity V and thrust T during the steady climb. We can then proceed to calculate the $SOC(t_2)$ of the battery for a given $h_{\max}$ and the number of Δt intervals k.

5.1.3. *Mode 3 (Steady Glide).* In the third mode, the solar power is not available ($P_{Sun} = 0$) and the battery is discharging ($P_{Batt} > 0$). Therefore, the UAV exploits its potential energy and battery storage to glide through the night. Based on the numerical results, the flight path angle is assumed to be small and the bank angle is zero. Thus, the aircraft kinematics for this mode is expressed as in Eq. (5.7). Similarly, the time horizon is descretized into k intervals of Δt and the rate of descend is constant:

$$z_g = \frac{\Delta z}{\Delta t} < 0. \tag{5.11}$$

Given that $P_{Batt} = P_{Eng}$ in this mode, the objective is to minimize the discharge rate of the battery or equivalently the power required for the electric motor. Therefore, similar to

mode 2, the optimal control problem can be expressed as

$$\min_{V,T} \int_{t_2}^{t_f} \left(\frac{TV}{\eta_{prop}} \right) dt \tag{5.12}$$

$$\text{s.t.} \quad \dot{V} = \frac{T-D}{m} - \frac{z_g}{V} g, \quad V_{stall} \leq V \leq V_{\max}, \tag{5.13}$$

where

$$z_g = \frac{z(t_f) - z_{\max}}{k\Delta t},$$

$$T - D = \frac{VW}{z_g} - \frac{\rho S V^3}{4K z_g^2} \left(V + \sqrt{V^2 - 4KC_{D0} z_g^2 - \frac{8KSz_r}{\rho S V^2}(WV - z_g T)} \right), \tag{5.14}$$

and time instant t_f refers to the final time of the mission. Moreover, Eq. (3.5b) implies that $z(t_f) = z_f$. Based on the solution of this optimal control problem and Eq. (I.8), we can find $SOC(t_f)$ for a given $z_{\max}$ and number of Δt intervals k.

5.2. Analysis. In [**19**], it has been shown that the proposed reduced hybrid model yields similar power allocation strategy and optimal cost as the one obtained from the original model using the direct optimization approach. However, the reduced hybrid model leads to a superior computational efficiency as compared with the full scale optimization. In this section, we examine the variation of the cost function presented in Eq. (3.2) as a function of $z_{\max}$ based on the proposed reduced hybrid model. Since the initial time and $SOC(t_1)$ of the steady climb are assumed to be known, we can carry on the optimization process for each mode sequentially and find the coverage function and SOC at final time. Figures 5.2 and 5.3 show the numerical simulations results for the final state of charge of the battery and coverage cost, respectively.

It is shown in Fig. 5.2 that the $SOC(t_f)$ is optimum at altitude gain around $5000 \sim 6000\,m$. For higher altitude gain, the UAV has to spend its energy storage in order to reach the desired $z_{\max}$ during the second mode. In other words, the potential energy does not always help to optimize the energy storage. Figure 5.3 shows that the coverage cost $C(t_f)$ decreases as we increase $z_{\max}$. However, a minimum coverage is guaranteed during

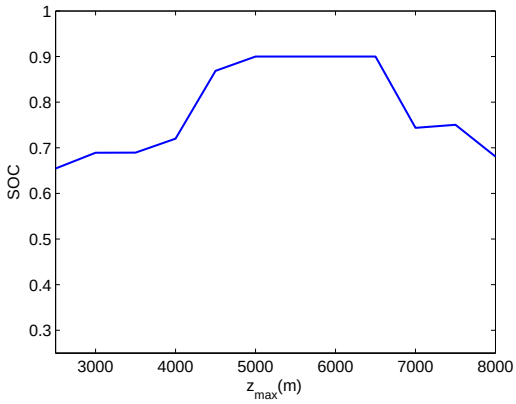


FIGURE 5.2. SOC of battery pack at time instant t_f as a function of maximum altitude gain during 1 day using reduced hybrid model.

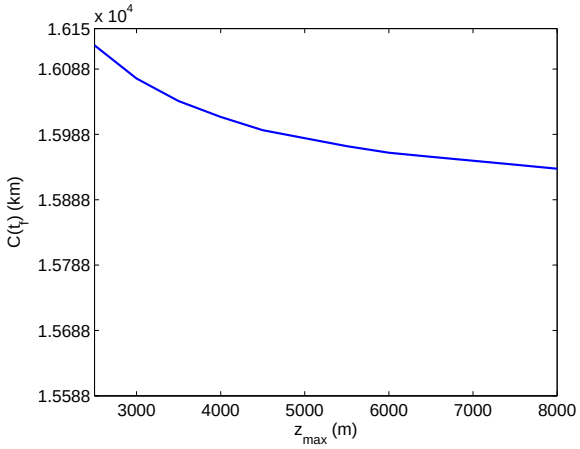


FIGURE 5.3. Coverage cost at time instant t_f as a function of maximum altitude gain during 1 day using reduced hybrid model.

mode 1 namely $C(t_1) = 15588\,\text{km}$. Therefore, these numerical results suggest that the energy aware surveillance problem can be expressed as a tradeoff between energy storage and sensor coverage. In the next section, we extend the reduced hybrid model and study the effect of intermittent solar radiation during mode 2 on the power allocation strategy.

6. Online Estimation of Rate of Climb to Cope with Intermittent Solar Radiation

Both nonlinear programming approach and the solution based on reduced hybrid model provide complete information on the trajectory and power allocation strategy of the UAV

given accurate estimates on the solar radiation intensity. However, the solar radiation is variable due to unpredictable atmospheric conditions such as clouds. In other words, the power spectral density P_{sd} is not known a priori. Therefore, we modify mode 2 in reduced hybrid model to an online optimization problem in order to consider the unpredictable variation of P_{sd}. For simplicity of analysis, we assume that the velocity is given and constant during mode 2. The constant velocity is denoted by V_2 and from Eq. (5.9) we have

$$TV_2 = mgz_c + DV_2. \tag{6.1}$$

Since the flight path angle is very small and $\phi = 0$, the lift force is expressed as

$$L = mg\cos\gamma \simeq mg, \tag{6.2}$$

and the drag force in Eq. (6.1) is computed from Eq. (I.11). In addition, we assume that the time horizon $[t_1, t_2]$ is descretized into N intervals of Δt. The objective is to minimize the value of P_{Batt} during the time horizon $[t_1, t_2]$ to achieve the maximum recharging rate and solar energy harvest. Furthermore, the UAV must reach the altitude $z_{\max}$ at the end of the interval. Therefore, the optimal control problem in Eq. (5.8) can be expressed as

$$\begin{aligned} &\min_{z_c} \sum_{k=1}^{N} \left(\frac{T(z_c)V_2}{\eta_{prop}} - \eta_{sol} P_{sd}(k) S \cos e(k) \frac{z_c}{V_2} + \frac{\lambda}{2} ||z_c N\Delta t + z(t_1) - z_{\max}||^2 \right) \Delta t \\ &\text{subject to } 0 \leq z_c, \end{aligned} \tag{6.3}$$

where $\lambda > 0$ is the regularization constant and can be tuned to balance the trade-off between energy harvesting and reaching the desired maximum altitude. Note that the goal is to find a fixed rate of climb such that the cost function (6.3) is minimized. The problem in Eq. (6.3) can be posed as an online convex optimization problem where the online algorithm minimizes the regret defined in Eq. (I.14). Thus based on Eq. (6.3), the

convex cost function in Eq. (I.13) is expressed as

$$f_k(z_c) = \frac{DV_2}{\eta_{prop}} + z_c\left(mg - \frac{\eta_{sol}}{V_2}P_{sd}(k)S\cos e(k)\right) + \frac{\lambda}{2}\left(z_c N\Delta t + z(t_1) - z_{\max}\right)^2, \tag{6.4}$$

and closed convex set $\chi \subseteq \mathbb{R}$ is given by

$$\chi = \{\xi \in \mathbb{R} \,|\, 0 \leq \xi\}.$$

Note that the gradient of $f_k(z_c)$ is

$$g_k = mg - \frac{\eta_{sol}}{V_2}P_{sd}(k)S\cos e(k) + \lambda N\Delta t(z_c N\Delta t + z(t_1) - z_{\max}), \tag{6.5}$$

that can be determined based on the amount of solar radiation. In addition, $f_k(z_c)$ is G-Lipschitz continuous, i.e., for all $\xi, \varrho \in \chi$ we have

$$|f_k(\xi) - f_k(\varrho)| \leq G||\xi - \varrho||,$$

where $G = mg$. Moreover, we note that Eq. (6.4) implies the direct proportionality of the rate of climb and the energy harvested from solar radiation. In other words, the UAV climbs higher as the available solar radiation increases. The algorithm used to learn the rate of climb is presented in the next section.

Online Dual Averaging Algorithm. We use Nesterov's dual averaging algorithm **[64, 110]** to solve the online optimization problem presented in Eq. (I.13). The algorithm sequentially updates the state variable $z_c(k)$ and the dual gradient $q(k)$ at each iteration. The update is based on a provided gradient of a convex cost function $f_k(z_c(k))$ denoted as g_k followed by a projection step onto the constraint set χ, specifically,

$$q(k+1) = q(k) + g_k$$
$$z_c(k+1) = \prod_{\chi}^{\tau}\left(q(k+1), \alpha(k)\right).$$

Note that $\prod_{\chi}^{\tau}(\cdot)$ is the projection operator onto χ,

$$\prod_{\chi}^{\tau}(q(k),\alpha(k)) = \arg\min_{\xi\in\chi}\left\{\langle q(k),\xi\rangle + \frac{1}{\alpha(k)}\tau(\xi)\right\}, \tag{6.6}$$

where $\alpha(k)$ is a non-increasing sequence of positive functions and $\tau(\xi) : \chi \to \mathbb{R}$ is a proximal function. The standard dual averaging algorithm uses proximal function $\tau(\xi)$ to avoid oscillations in the projection step. Without loss of generality, τ is assumed to be strongly convex with respect to $\|.\|$, $\tau \geq 0$, and $\tau(0) = 0$. An example of proximal function is the quadratic function $\tau(\xi) = 1/2||\xi||_2^2$ that is strongly convex with respect to the second norm $||.||_2$ for $\xi \in \mathbb{R}$. Moreover, the entropic function $\tau(\xi) = \sum_{i=1}^{n} \xi_i \log \xi_i - \xi_i$ is a proximal function which is strongly convex with respect to the first norm $||.||_1$ for ξ in the probability simplex, $\{\xi \,|\, \xi \geq 0, \sum_{i=1}^{n} \xi_i = 1\}$.

The Online Dual Averaging (ODA) algorithm is presented in Protocol 2. The projection function used in this algorithm is defined in (6.6).

Algorithm 1: Online Dual Averaging (ODA)

```
for k = 1 to N do
    f_k(ẑ_c)(k)) is revealed
    Compute subgradient g(k) ∈ ∂f_k(ẑ_c)(k))
    d(k+1) = d(k) + g(k)  ẑ_c(k+1) = Π_χ^Γ(d(k+1), α(k))
end
```

Xiao [**64**] presented a sub-linear regret of $O(\sqrt{N})$ for ODA which implies that the algorithm cost is converging to the best fixed case in hindsight. In other words, we have

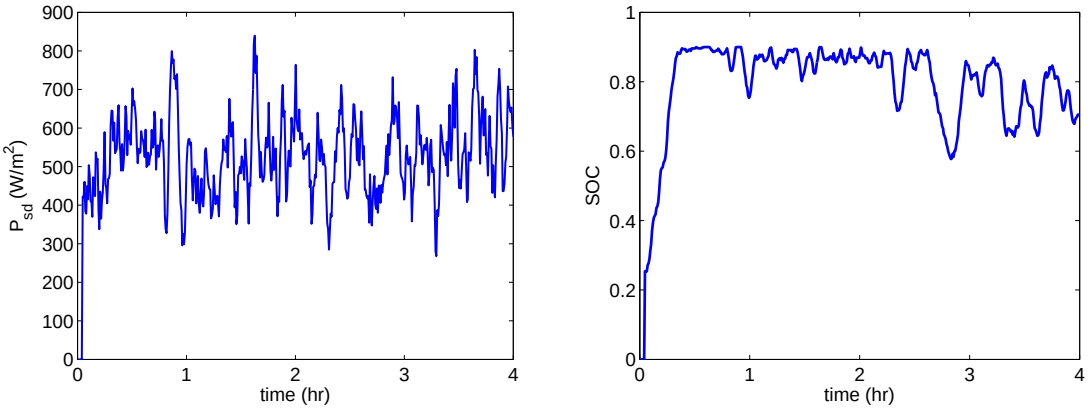
$$\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}(f_k(z_c(k)) - f_k(z_c^*)) = \lim_{N\to\infty}\frac{1}{N}O(\sqrt{N}) = 0, \tag{6.7}$$

where z_c^* is the best fixed rate of climb. We can calculate z_c^* from Eq. (6.3) and based on full knowledge of the hindsight. In addition, we can express the cost function in Eq. (6.3) based on Eq. (6.4) as $\sum_{k=1}^{N}[f_k(z_c(k))\Delta t]$. Therefore, Eq. (6.7) implies

$$\lim_{N\to\infty}\left(\sum_{k=1}^{N}[f_k(z_c(k))\Delta t] - \sum_{k=1}^{N}[f_k(z_c^*)\Delta t]\right) = 0, \tag{6.8}$$

where $\Delta t = (t_2 - t_1)/N$. Consequently, Eq. (6.8) implies that the solution of the online algorithm converges to the best fixed rate of climb as we increase the number of time intervals N.

The ODA algorithm was applied on the described setup for a solar powered UAV. The objective is to estimate a scalar $z_c \geq 0$. The observation at iteration k is defined in Eq. (6.5) for $P_{sd} \in (0, 886)\ W/m^2$, $z_{\max} = 5\,\text{km}$, and $V_2 = 10\,m/s$. The time step Δt is $30\,sec$ implying that the UAV may change the rate of climb every 30 seconds based on the availability of the solar radiation. Figure 6.1(a) presents the average power spectral density P_{sd} during mode 2 which is lower than the nominal value of $886\,W/m^2$ considered in the offline problem. The SOC of the battery pack is also shown in figure 6.1(b).

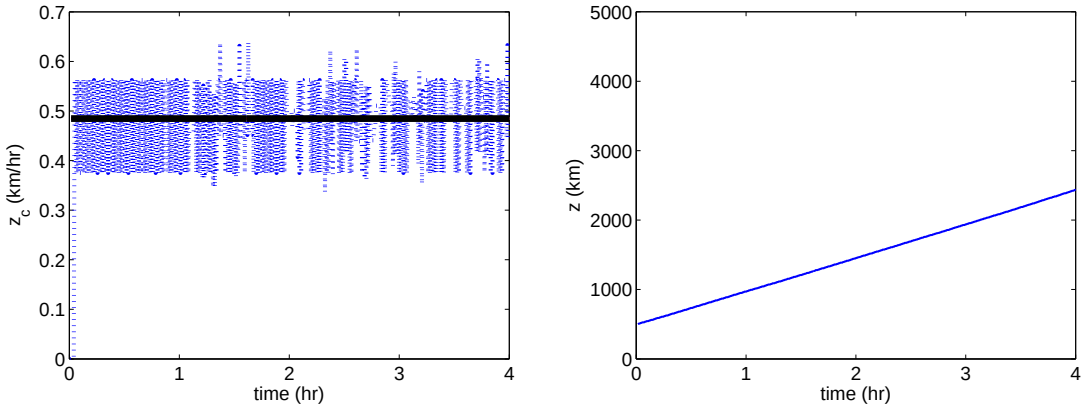


(a) (b)

FIGURE 6.1. Figure (a) presents the average power spectral density P_{sd} and figure (b) presents the SOC of the battery pack during mode 2 for the online estimation of rate of climb problem.

Moreover, based on the available P_{sd}, the best fixed rate of climb z_c^* is $0.4848\,\text{Km/hr}$ and Fig. 6.2(a) depicts the ODA algorithm solution for z_c. Note that the best fixed rate of climb is represented by the horizontal solid line in Fig. 6.2(a) and the ODA solution is varying in the vicinity of z_c^*. Finally, based on the values of $z_c(k)$ and Eq. (5.5) the approximate altitude of the UAV is computed and shown in Fig. 6.2(b). It is shown that due to lack of enough solar radiation, the UAV does not reach the desired maximum

altitude $z_{\max} = 5\,\text{km}$. However, the battery pack achieves a SOC of 70% at the end of mode 2, sufficient for carrying on the mission at the next modes.



(a) (b)

FIGURE 6.2. Figure (a) presents the rate of climb z_c and figure (b) presents the altitude of UAV as a function of time during mode 2 for the online estimation of rate of climb problem.

Moreover, Fig. 6.3 shows the agreement of the theoretical regret bound, presented Eq. (6.7), and simulation results based on ODA algorithm with respect to their asymptotic sublinear behavior. Note that the theoretical regret bound represents the worst case scenarios.

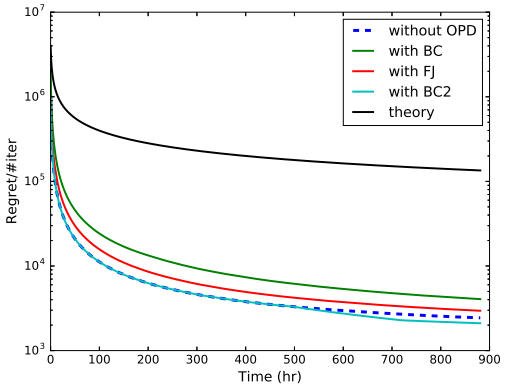


FIGURE 6.3. The theoretical regret bound and the regret for the online estimation of rate of climb problem.

7. Conclusion

A simple model has been proposed for the optimal energy surveillance problem of a solar-powered UAV in 3D space. In this model, the sensor capabilities and electrical models are integrated with the aircraft kinematics when solving the corresponding optimization problem. It was noted from the nonlinear programming approach that the UAV mission is generally divided into three phases: level flight, climb, and descend and the most effective contribution to saving energy is due to the potential energy harvested during the climb phase. Based on a reduced hybrid model and the analysis of the maximum altitude gain, it was shown that the energy aware surveillance problem is a tradeoff between maximum altitude gain and coverage cost. Furthermore, there exists a range of potential energy in which the energy storage is optimum. By exploiting the reduced hybrid model, the energy aware surveillance problem was modeled as an online convex optimization problem where the effect of unpredictable intermittent solar radiation on the maximum altitude gain is considered. The numerical and theoretical results suggest that online learning algorithms can specify the rate of climb based on the available solar radiation intensity on the fly such that the solar powered UAV will be capable of carrying on its mission.

This chapter is a step towards designing trajectories for long endurance solar powered UAVs, a topic that requires further investigation. The future research will focus on thermal soaring and the effect of wind gust on the UAV trajectory. Cooperative aerial surveillance and developing convex models for more efficient on-board implementations will also be explored in our future works.

Appendix

The aircraft parameters for a sample UAV which is used in numerical simulations are presented in Table 2. Table 3 contains the battery pack parameters and the sensor parameters are given in Table 4 .

Table 2. Sample UAV parameters

η_{sol}	$P_{sd}(\frac{W}{m^2})$	$b(m)$	$S(m^2)$	$m(kg)$	η_e	η_{prop}
0.19	886	3.1	0.56	4.201	0.992	0.7

TABLE 3. Battery pack (Lithium-Sulfur) parameters

Q	V_{OC}	R	m_{Batt}	$P_{Batt_0,Discharge/Charge}$
26.4 Ah	12.6 V	0.0125 Ω	1.280 kg	$\pm$30 watt

TABLE 4. Sensor parameters

θ	$h_{\max}$	h_0
30 deg	500 m	750 m

CHAPTER 3

Power Management of Cooling Systems with Dynamic Pricing

1. Summary

This chapter addresses the optimal power management problems in electric cooling systems based on appropriately constructed thermal dynamic models and cost profiles. In this venue, the dynamics and logical constraints for the cooling load are first formulated as mixed-integer linear programming models. We subsequently apply an online learning algorithm to adjust the weighting factor for customers' satisfaction level considering the fluctuating prices and customers' preferences. The proposed approach is expected to save the user's electricity cost by adequately scheduling the operations of the cooling load without an adverse effect on the entire system. Simulation results for the chiller systems are presented using the proposed approach. The effectiveness of the proposed temperature control and trade-off between electricity cost and customers' satisfaction level is demonstrated via a simulation scenario.

2. Introduction

Future smart grids will rely on new levels of transparency and coordination between providers and consumers of electric energy. On one side, the electricity providers will be tracking and estimating the consumer's daily, weekly and seasonal demands to coordinate the energy requests. On the other side, some customers will be enticed to change their consumption habit to save cost and alleviate peak demand by adopting new technologies, e.g., dynamic pricing [**36, 37**]. The success of cost reduction at the consumer's side requires accessing real-time prices and adapting corresponding strategies, e.g., scheduling operation of loads when the electricity is cheaper. In this chapter, we examine an optimization-based

approach for temperature control of cooling systems. The goal is to reduce the cost of energy consumption while satisfying the demands of the consumers. Finding an efficient demand schedule is challenging due to the nonlinear thermodynamics and complex heat exchange processes associated with environment and the heating/cooling systems [**38, 39**]. Introducing simple thermal load models and applying different control strategies have brought significant progress in this field. For example, Perfumo *et al.* [**40**] proposed a model based control that thermostatically schedule the loads. Katipamula and Lu [**41**] evaluated different thermostat setting approaches for energy saving of residential heating, ventilation, and air-conditioning (HVAC) systems. Moreover, Ha *et al.* [**42**] applied a multi-scale optimization mechanism in load management of electric home heaters. In addition, uncertainties in energy price [**43, 44**] and power resources [**45**] affect the resulting optimal control problem. The work presented in [**46, 47, 48**] addresses the uncertainties in the energy sources by controling the energy storage and load operation.

Inspired by these works, we propose a mixed integer linear programming (MILP) model for a thermostatically controlled cooling system which allows for power curtailment strategy to reduce the energy cost. However, our approach can override the thermostat control command when necessary, e.g., temporally pausing the operation of the cooling system during high price intervals without adverse effect on the overall operation. We focus on cooling systems designed specifically for storage of brewery, winery, dairy, and etc., with a preferred temperature range. By efficiently scheduling the cooling systems' operation, the proposed approach is expected to minimize the energy cost, and at the same time provide a degree of consumer satisfaction which is reflected by the difference between the ideal temperature and real maintained temperature.

The optimization problem is modeled as a trade off between energy cost and the consumer satisfaction level. Therefore, the user's preference and the price he/she is willing to pay has a significant impact on the cost function. However, the electricity price is not known a priori and the user's preference is changing with time, making it difficult to set up a tractable optimization problem. Therefore, an online estimation approach is implemented to model the weighting factor of consumer's satisfaction level. This approach is

a special case of our previous work **[95]** regarding online distributed estimation via dual averaging. Online learning algorithms have been proposed **[49, 50]** to address the uncertainties in the systems *without probabilistic assumption* where the stochastic optimization methods **[61, 62, 112]** are inadequate. Such learning algorithms have been widely used to solve the optimization problems with unknown cost function at the time when relevant decision is made **[138]**. The performance of online algorithms is captured by *regret* which is a standard measure in machine learning literature **[139, 141]**. *Regret* represents the non-optimality of the algorithm by not following the best fixed decision in hindsight. Consequently, the average regret of a good algorithm should approach zero over time.

The main contribution of the present chapter is to formulate an MILP model for the thermostatically controlled cooling systems based on estimated weighting factors obtained from online estimation algorithm and electricity companies. The power curtailment strategy is implemented by solving optimal power management problem via a MILP solver **[52]**. In addition, we propose a novel approach to estimate the weighting factor of customer's satisfaction level based on the dual averaging algorithm. Further analysis on the convergence of the proposed approach is provided.

The organization of the chapter is as follows. First, we formulate the power management problem of cooling systems in §3. Subsequently, the (MILP) model of the thermostatically controlled loads is formulated in §4, followed by the dual averaging algorithm for setting the weighting factor in §5. Simulation example demonstrating the applicability of the proposed approach is detailed in §6, which is followed by concluding remarks in §7.

3. Problem Formulation

The objective of power management for cooling systems is to schedule the loads' operation such that the consumed electricity cost is minimized and at the same time a degree of consumer satisfaction is maintained. Assuming the consumers have access to the setting

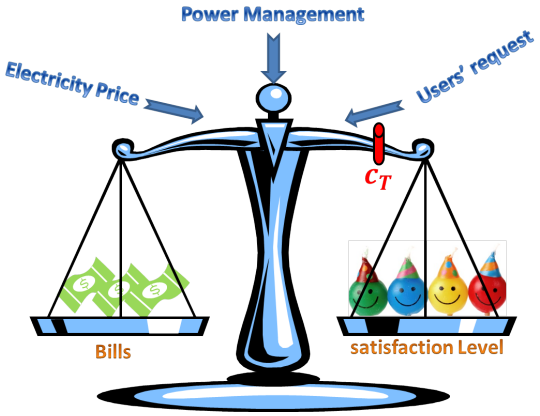


FIGURE 3.1. Trad-off between electricity cost and satisfaction level with weighting factor c_T

the ideal temperature, T_I, for the cooling system, the objective function is defined as

$$\min_{z_1,z_2} \sum_{t=t_0}^{t_f} [c_p(t)\,(g_1 z_1(t) + g_2 z_2(t)) + c_T |T(t) - T_I|]\Delta t \tag{3.1}$$
$$\text{subject to } T(t) \in [T_{min}, T_{max}],$$

where Δt is the step size of each horizon, g_1 and g_2 are the rapid pull down working power (with fast cooling rate) and the normal working power (with normal cooling rate), respectively. The term c_p refers to the day-ahead predicted electricity price published online from the electricity company at time t. The cooling temperature at horizon t is denoted by $T(t)$, while T_{max} and T_{min} are the allowed highest and lowest temperature, respectively. The term c_T is the coefficient related to the degree of consumer satisfaction; c_T acts as a weighting factor that balances the importance between the electricity cost and the consumer's preference as shown in Figure 3.1. However, c_T can be adjusted during the control interval when the electricity price or ideal temperature changes in the power system.

The solution of the optimization problem (3.1) is to find the operation control z_1 and z_2, where z_1, $z_2 \in \{0, 1\}$ and $z_1(t) + z_2(t) \leq 1$, for rapid pull down or normal operation of every load at each horizon to minimize the objective function from initial time t_0 to final time t_f. During the optimization process, the temperature changes due to the operation

and the heat exchange between the system and the environment. The temperature $T(t)$ is constrained by the thermal dynamic equation expressed as

$$\begin{aligned} T(t+1) &= T_e + Q_1 R z_1(t) + Q_2 R z_2(t) - \\ &\quad (T_e + Q_1 R z_1(t) + Q_2 R z_2(t) - T(t)) e^{-\frac{\Delta t}{RC}}. \end{aligned} \tag{3.2}$$

where T_e is the temperature of the environment. The electric cooling capacities for rapid pull down and normal operation are denoted by Q_1 and Q_2, respectively. The terms C and R are the cooling thermal capacitance and resistance, respectively. For the above described power management problem, on one hand, it is desired to maintain the minimum electricity cost based on the predicted tariffs. On the other hand, it is also desired to maintain the degree of consumer satisfaction to keep the cooling temperature as close as possible to the ideal setting. The decision to minimize all the terms in the objective function express a bargaining process that balances between the energy price and the degree of consumer satisfaction. When the predicted tariff is low, we intuitively intend to use the provided electricity to keep the temperature close to the desired setting. Otherwise, the cooling system is scheduled to temporarily shut down to allow the temperature rise above the setting with no power consumption. Then, there is no power consumption and the cost associated to the degree of consumer satisfaction is determined by $c_T|T(t) - T_I|\Delta t$, the third term in (3.1). With the three coupling terms in the objective function (3.1), the goal is to find the best solution to minimize their sum. Therefore, the operation control terms z_1 and z_2 are the key factors in determining the objective value. In addition, considering fluctuating electricity prices and uncertainties of consumers' preference, the weighting factor, c_T, contributes to the objective value as well. Determining the weighting factor is, however, more complicated and requires integration of objective cost from data. In the following, two integrated approaches, MILP and online learning, are introduced to obtain the optimal solution for the power management problem of cooling systems. From the MILP solution, the cooling system finds the controls, z_1 and z_2, under current value of c_T. The online learning approach updates the value of c_T by tracking the time history of the electricity price and ideal temperature settings.

4. Temperature Control via Mixed-Integer Linear Programming

MILP is the optimization problem of minimizing an objective function expressed by a linear combination of integral and real-valued state variables, subject to linear equality and inequality constraints. It can be solved using the branch and bound, branch and cut, or branch and price algorithms. There are numerous applications for MILPs in many areas of operations research, including network flow, path planning, and scheduling. In order to formulate the MILP model for dynamic programming, the nonlinear term $\exp[-\Delta t/(RC)]$ in (3.2) is approximated as a linear expression $1 - \Delta t/(RC)$. Consequently, the thermal dynamic equation in (3.2) is linearized as

$$T(t+1) = T(t) + (Q_1 R z_1(t) + Q_2 R z_2(t))\frac{\Delta t}{C} + (T_e - T(t))\frac{\Delta t}{RC}$$

During each horizon t, only one operation is allowed among the three options. These options include “rapid pull down" with $z_1(t) = 1$, “normal chilling" with $z_2(t) = 1$, and “off" with $z_1(t) + z_2(t) = 0$. Generally, the rapid pull down has higher cooling rate, but requires more operational power than the normal cooling. The described logical constraints can be expressed as

$$z_1(t) + z_2(t) \leq 1,\ z_1(t),\ z_1(t) \in \{0, 1\},\ \forall\, t \in [t_0, t_f].$$

To find the linear expression of the objective function (3.1), we introduce the new variable $y(t)$ at each time horizon to relax the objective value by assigning $y(t)$ as

$$y(t) \geq |T(t) - T_I|, \quad \forall\, t \in [t_0, t_f].$$

In addition, it is not desirable that the control commands switch frequently between working and “off" states, which is against the healthy operation of the electric unit. Therefore, we assign a upper bound temperature T_b, such that once the the rapid pull down or normal cooling operation is turned off, it will not be turned on again until the temperature increases beyond this upper bound. To realize this constrained operation, a binary variable $z_b(t)$, is introduced to indicate whether the temperature increases beyond the assigned

temperature T_b. Logically, z_b satisfies the following constraint

$$z_b(t) - T(t)/T_b \leq 0,\ z_b(t) \in \{0,1\}\ \forall t \in [t_0, t_f]. \tag{4.1}$$

The above inequality guarantees that when z_b equals to one, the temperature has to be more than T_b to make the upper bound of $z_b(t) - T(t)/T_b$ less than zero. In other word, $z_b(t)$ cannot be set as one if $T(t) \leq T_b$. However, $T(t) \geq T_b$ does not imply that $z_b(t)$ must be equal to one. In addition, the following expression,

$$z_1(t) + z_2(t) \leq z_b(t-1) + z_1(t-1) + z_2(t-1)\ \forall t \in [t_0, t_f],\ t \neq t_0, \tag{4.2}$$

ensures that the two types of cooling operation, rapid pull down and normal cooling, cannot be turned on unless z_b equals one in the previous horizon. The fact that z_b is equivalent to one implies that the boundary temperature is reached, or the cooling operation was performing in the previous horizon. Thus, the cooling operation cannot be turned on if the operation at previous step is not “on", $(z_1(t-1)+z_2(t-1) \neq 1)$, and the temperature does not increase beyond the boundary value $(z_b(t-1) \neq 1)$. With the two linear constrains expressed in (4.1) and (4.2), we can imagine that once the lower bound temperature is reached, neither of the cooling operations will be on again until the temperature increases beyond the boundary value, that is when $z_b(t) = 1$. The power output for the cooling systems is simply

$$P(t) = g_1 z_1(t) + g_2 z_2(t).$$

From the above description, the MILP formulation for the power management problem described in §3 is summarized as

$$
\begin{aligned}
(4.3)\qquad & \min_{z_1,z_2,y,z_b} \sum_{t=t_0}^{t_f} [c_p(t)(g_1 z_1(t) + g_2 z_2(t)) + \hat{c}_T y(t)]\Delta t \\
& \text{such that } \ \forall\, t \in [t_0, t_f]; \\
& T(t+1) = T(t) + (Q_1 z_1(t) + Q_2 z_2(t))\Delta t/C + (Te - T(t))\Delta t/(RC), \\
& T(t) \in [T_{min}, T_{max}], \\
& z_1(t) + z_2(t) \le 1, \\
& y(t) \ge T(t) - T_I, \ \ y(t) \ge TI - T(t), \\
& z_b(t) - T(t)/T_b \le 0, \\
& z_1(t) + z_2(t) \le z_b(t-1) + z_1(t-1) + z_2(t-1), \\
& z_1(t),\ z_2(t),\ z_b(t) \in \{0, 1\}.
\end{aligned}
$$

The above optimization problem provides the operation control z_1 and z_2 and the variables y and z_b for the predicted power price $c_p(t)$, ideal temperature T_I, weighting factor $\hat{c}_T$ and other pre-specified parameters. Most parameters in the above equations are specified from external entities and are time invariant, e.g., g_1 and g_2. However, the electricity price, satisfaction level coefficient, and the ideal temperature are dynamic according to the market and user's preference. Therefore, the power management system needs to adapt the weighting factor $\hat{c}_T$ in order to compensate for these uncertainties. The following section specifically describes an online estimation scheme to predict the weighting factor c_T in (3.1) a day ahead and adjust $\hat{c}_T$ for the MILP (4.3).

5. Weighting Factor Design via Online Estimation

The consumer satisfaction level coefficient c_T is generally changing based on the user preference and the cost she/he is willing to pay. Therefore, $c_T(t)$ is unknown a priori and an adaptive scheme is required to update this parameter at each time step. If we solve the optimization (3.1) for N days, the cost at the end of day τ based on the actual electricity

price $c_{r,\tau}$ is

$$C_R(\tau) = \sum_{t=t_0}^{t_f} [c_{r,\tau}(t)P(t) + c_T \Delta T_\tau(t)]\Delta t, \tag{5.1}$$

while the predicted cost at the end of day τ based on the day-ahead predicted electricity price $c_{p,\tau}$ is

$$C_P(\tau, x) = \sum_{t=t_0}^{t_f} [c_{p,\tau}(t)P(t) + \hat{c}_T \Delta T_\tau(t)]\Delta t. \tag{5.2}$$

Note that $\hat{c}_T(\tau)$ represents the estimated value of c_T on the τth day. Moreover, $\Delta T_\tau(t) = |T(t) - T_I(\tau)|$, and $P(t)$ is the power output for the cooling system. Therefore, specifying consumer satisfaction level coefficient can be formulated as an online estimation problem where the objective is to find the argument x that minimizes the cost function

$$f_\tau(\hat{c}_T) = \frac{1}{2}||C_R(\tau) - C_P(\tau, \hat{c}_T)||_2^2 \text{ subject to } \hat{c}_T \in \chi. \tag{5.3}$$

Note that $\chi = \{x | 0 \leq x \leq c_{T,\max}\}$ is a closed convex set, where $c_{T,\max}$ is an upper bound on the coefficient representing the consumer's satisfaction level. The function $f_\tau(\hat{c}_T)$ is allowed to change over time in an unpredictable manner due to modeling errors and uncertainties in the electricity price, user preference, and the environment. Based on (5.1) and (5.2), we can express (5.3) as

$$f_\tau(\hat{c}_T) = \frac{1}{2}||b_\tau - \hat{c}_T a_\tau||_2^2 \quad \text{subject to } \hat{c}_T \in \chi, \tag{5.4}$$

where $b_\tau = \sum_{t=t_0}^{t_f} [(c_{r,\tau}(t) - c_{p,\tau}(t))P(t) + c_T(\tau)\Delta T_\tau(t)]\Delta t$ and $a_\tau = \sum_{t=t_0}^{t_f} \Delta T_\tau(t)\Delta t$. The (sub)gradient of the estimation error (5.4) is

$$\partial f_\tau(\hat{c}_T) = -a_\tau \left(b_\tau - \hat{c}_T a_\tau\right), \tag{5.5}$$

which is assumed to be known to the pricing algorithm. Further, since the function $f_t(x)$ is convex on a compact domain, it is Lipschitz, i.e. there exists a positive constant L for

which

$$|f_\tau(x) - f_\tau(y)| \leq L\|x - y\| \text{ for all } x, y \in \chi. \tag{5.6}$$

The Lipschitz constant L for the cost function in (5.4) can be found by observing that [1]

$$\begin{aligned} &|f_\tau(x) - f_\tau(y)| \\ &\leq \frac{1}{2} \|a_\tau\|_F \left(\|a_\tau\|_F \|x - y\|_2 + \|b_\tau\|_2\right) \|x - y\|_2 , \end{aligned}$$

for all $x, y \in \chi$. Assuming that $c_{r,\tau}, c_{p,\tau} \in (0, c_{\max})$, $c_T(\tau) \in (0, c_{T,\max})$, $P \in (0, P_{\max})$, and $\Delta T_\tau \in (0, T_{\max} - T_{\min})$, we have

$$\begin{aligned} L &= \frac{1}{2} \left(3c_{T,\max}(T_{\max} - T_{\min}) + 2c_{\max}P_{\max}\right) \\ &\quad (T_{\max} - T_{\min})(t_f - t_0)^2. \end{aligned} \tag{5.7}$$

Note that L is a function of temperature range, maximum power output for the cooling systems, maximum satisfaction level coefficient, and maximum electricity price. Since the energy price, ideal temperature and environment are dynamics, the behavior of (5.3) is unpredictable. Therefore, we need an online scheme in which no assumption or knowledge of the statistical properties of the data are available. In the proposed estimation algorithm, at time step τ, the system estimates $\hat{c}_T \in \chi$ and then an "oracle" announces the cost $f_\tau(\hat{c}_T)$ and its (sub)gradient $\partial f_\tau(\hat{c}_T)$.

5.1. Online Estimation via Dual Averaging. In order to solve this estimation problem in an online setting, we employ Nesterov's dual averaging algorithm **[64, 110]**. The Online Dual Averaging (ODA) algorithm is presented in Algorithm 2 in which the state variable $\hat{c}_T(\tau)$ and a dual variable $d(\tau)$ are updated sequentially:

$$d(\tau + 1) = d(\tau) + \gamma(\tau),$$

[1]Note that the Frobenius norm of A is defined as $||A||_F = \sqrt{\text{trace}(A^T A)}$.

where $\gamma(\tau) = \partial f_\tau(\hat{c}_T(\tau))$ is the (sub)gradient of the cost function. Then, $\hat{c}_T(\tau+1) = \prod_\chi^\Gamma (d(\tau+1),,\alpha(\tau+1))$, where $\prod_\chi(\cdot)$ is a projection onto χ and is defined as

$$\Pi_\chi^\Gamma(d(\tau),\alpha(\tau)) = \arg\min_{\hat{c}_T(\tau)\in\chi} \left\{ \langle d(\tau), \hat{c}_T\rangle + \frac{1}{\alpha(\tau)}\Gamma(\hat{c}_T) \right\}, \tag{5.8}$$

where $\alpha(\tau)$ is a non-increasing sequence of positive functions. In addition, $\psi(\hat{c}_T) : \chi \to \mathbb{R}$ is a proximal function and is used to avoid wide oscillation in the projection step. Without loss of generality, Γ is assumed to be strongly convex with respect to $\|.\|$,$\Gamma \geq 0$, and $\Gamma(0) = 0$. Hence, we can select $\Gamma(\hat{c}_T) = \frac{1}{k}\|\hat{c}_T\|_2^2$ for some $k > 0$, which is bounded as $\Gamma(\hat{c}_T) \leq \frac{1}{k} c_{T,\max}^2$ in our problem setting.

Algorithm 2: Online Dual Averaging (ODA)

for $\tau = 1$ *to* N **do**
  $f_\tau(\hat{c}_T(\tau))$ is revealed
  Compute subgradient $\gamma(\tau) \in \partial f_\tau(\hat{c}_T(\tau))$
  $d(\tau+1) = d(\tau) + \gamma(\tau)$ $\hat{c}_T(\tau+1) = \prod_\chi^\Gamma(d(\tau+1),\alpha(\tau))$
end

In addition, the power mangament process is demonstrated in Figure 5.1 where $\partial f_\tau(\hat{c}_T)$ is computed based on previous information, real electricity price for the previous day, and the temperature variation. Then, the MILP solver finds an optimum load schedule based on day-ahead electricity price and estimated weighting factor provided by online estimation algorithm. At the end of the day, the electricity company collects energy demand information from all residential units and aggregates them to provide an estimate of electricity price for the next day as well as the cost of electricity for the current day.

5.2. Convergence Analysis. The convergence analysis for the online algorithms is based on a measure called *regret* and is expressed as

$$R_N(c_T^*, \hat{c}_T) = \sum_{\tau=1}^{N} (f_\tau(\hat{c}_T(\tau)) - f_\tau(c_T^*),) \tag{5.9}$$

over $t = 1, 2, ..., N$, iterations. The goal of online algorithms is to ensure that the cumulative penalty (5.9) that algorithm incurs due to its decisions on the cost sequence $\{f_\tau\}_{\tau=1}^N$ is

small. In other words, if $\lim_{N\to\infty} R_N/N = 0$, the algorithm performs as well as the best fixed strategy c_T^* independent of the uncertainties.

The following Lemma 3 can be found in **[59]**, and is required for our analysis.

LEMMA 3. *For any positive and non-increasing sequence $\alpha(\tau)$ and $c_T^* \in \chi$*
$\sum_{\tau=1}^{N} \langle \gamma(\tau), \hat{c}_T(\tau) - c_T^* \rangle \leq \frac{1}{2} \sum_{\tau=1}^{N} \alpha(\tau-1)\|\gamma(\tau)\|_*^2 + \frac{1}{\alpha(N)}\Gamma(c_T^*)$,
where $||.||_*$ is the dual norm. [2]

THEOREM 4. *The sequence of $\hat{c}_T(\tau)$ and $d(\tau)$ are generated by lines* 5 *and* 8 *in* Algorithm 2, *with $\Gamma(c_T^*) \leq \frac{1}{k} c_{T,\max}^2$ and $\alpha(\tau) = q/\sqrt{\tau}$, where $k, q > 0$. Thus, we have*

$$R_N(c_T^*, \hat{c}_T) \quad \leq \left(qL^2 + \tfrac{1}{qk} c_{T,\max}^2 \right) \sqrt{N} - \tfrac{L^2}{2} q,$$

where L is given in (5.7).

PROOF. The proof is similar to the regret analysis in **[64, 110]** and is presented here for completeness. An arbitrary fixed decision $c_T^* \in \chi$ and sequence $\hat{c}_T(\tau)$ generated by line 8 in Algorithm 2 are given. Since f_t is convex,

$$f_\tau(\hat{c}_T) - f_\tau(c_T^*) \leq \langle \gamma(\tau), \hat{c}_T(\tau) - c_T^* \rangle \tag{5.10}$$

Thus, from the definition of regret in (5.9),

$$R_N(c_T^*, \hat{c}_T) \leq \sum_{\tau=1}^{N} \left(\langle \gamma(\tau), \hat{c}_T(\tau) - c_T^* \rangle \right). \tag{5.11}$$

Therefore, using the bound in Lemma 3, we have

$$R_N(c_T^*, \hat{c}_T) \leq \frac{1}{2} \sum_{\tau=1}^{N} \alpha(\tau-1)\|\gamma(\tau)\|_*^2 + \frac{1}{\alpha(N)}\Gamma(c_T^*). \tag{5.12}$$

Note that convexity of f_τ implies that $\langle \gamma, x - y \rangle \leq f_\tau(x) - f_\tau(y)$ for all $x, y \in \chi$. Therefore, based on L-Lipschitz continuity of f_τ, we have $||\gamma||_* \leq L$. Thus, using (5.12) the regret is

[2] Note that the dual norm of a vector x is defined as $||x||_* = \sup_{||y||=1} \langle x, y \rangle$.

further bounded as

$$R_N(c_T^*, \hat{c}_T) \leq \frac{L^2}{2} \sum_{\tau=1}^{N} \alpha(\tau - 1) + \frac{1}{\alpha(N)} \Gamma(c_T^*), \tag{5.13}$$

and the theorem follows by applying the integral test on the first term in (5.13). [3] □

The "good" performance of the ODA algorithm is demonstrated by sub-linear regret. In addition, the result shows the importance of the underlying system properties through parameter L. Further, we can improve the regret bound by selecting appropriate values for q and k in (4.3).

Next we exhibit a similar dependence on the parameters of the system for the regret analysis of the (temporal) running average estimates.

COROLLARY 5. *The sequence of* $\bar{c}_T(\tau)$ *is defined as* $\bar{c}_T(\tau) = \frac{1}{\tau} \sum_{s=1}^{\tau} \hat{c}_T(s)$ *where the sequence of* $\hat{c}_T(\tau)$ *is generated by line* 8 *in* Algorithm 2. *Let* $\Gamma(c_T^*) \leq \frac{1}{k} c_{T,\max}^2$ *and* $\alpha(\tau) = q/\sqrt{\tau}$, *where* $k, q > 0$. *Thus, we have*

$$R_N(c_T^*, \bar{c}_T) \leq 2 \left(qL^2 + \frac{1}{qk} c_{T,\max}^2 \right) \sqrt{N}.$$

PROOF. Since the cost function $f_\tau(\hat{c}_T(\tau))$ is convex, $f_\tau(\bar{c}_T(\tau)) \leq \frac{1}{\tau} \sum_{s=1}^{\tau} f_\tau(\hat{c}_T(s))$. Therefore, we have

$$f_\tau(\bar{c}_T(\tau)) - f_\tau(c_T^*) \leq \frac{1}{\tau} \sum_{s=1}^{\tau} \left(f_\tau(\hat{c}_T(s)) - f_\tau(c_T^*) \right), \tag{5.14}$$

and given the definitions of regret in (5.9) we have

$$R_N(c_T^*, \bar{c}_T) \leq \sum_{\tau=1}^{N} \left(\frac{1}{\tau} R_\tau(c_T^*, \hat{c}_T) \right), \tag{5.15}$$

and the corollary follows from (5.15). □

6. Simulation Example

In order to illustrate the feasibility of the proposed approach and the MILP model, a simulation example for a small scale cooling system is examined in this section. We

[3] Note that $\sum_{\tau=1}^{N} \frac{q}{\sqrt{\tau}} \leq 2q\sqrt{N} - q$.

aim at validating the applicability and efficiency of the proposed approach for scheduling the operation of the cooling load. At the beginning of each day, we obtain the day-ahead predicted prices online from [**43**] and calculate the optimal schedules using the MILP algorithm with current ideal temperature setting and the estimated weighting factor. At the end of that day, the real-time price is published online and we adjust the weighting factor for the next day. An example of the predicted and real-time price in Zone I of Illinois state on October 7^{th} of 2012 is demonstrated in Figure 6.1.

In addition, the average electericity price per day over 30 days is presented in Figure 6.2 which implies the day-ahead price is an over estimation most of the times.

The power management process in Figure 5.1 is repeated as long as the operation schedule is required. Additional loads can also be easily included in the current framework. The corresponding parameters of the cooling load used in the simulation are listed in Table 1 [**38**].

TABLE 1. Parameters used in the cooling loads

g_1 (kw)	g_2 (kw)	T_e (F)	T_{min} (F)	T_{max} (F)
75	50	72	40	72
Q_1/C (F/s)	Q_2/C (F/s)	$1/RC$ (F/s)	Δt(minutes)	T_b (F)
-3	-1.5	0.02	2	$T_I + 5$

In addition, the ODA algorithm was applied on the described system to estimate a scalar $c_T \in (0, 0.1)$ with $q = 0.004$, $k = 1$, and $(T_{\max}, T_{\min}) = (40, 72)$. Thus, $\chi = (0, 0.1)$, $c_{T,\max} = 0.1$, and $L = 4953.6$. During one day and night, the MILP algorithm generates optimal schedules for every hour with time step of two minutes. This demand scheduling is based on the predicted price, the current ideal temperature setting, and the designed weighting factor. The performance of the proposed approach is compared with thermostat strategy based on temperature tracking error [4] and electricity cost. We assume that the

[4]Note that the tracking error is a measure of how closely the true temperature follows T_I and is represented by $\sqrt{\frac{1}{K}\sum_{t=1}^{K}|T(t) - T_I(t)|^2}$ over K measurements.

thermostat strategy is to switch on when the temperature is above 5^oF over T_I and switched off when it drops 5^oF below T_I.

The temperature tracking error over a month is illustrated in Figure 6.3 for both online MILP and the thermostat strategy. This figure shows that the temperature profile with the online MILP approach traces the ideal setting better than the one with the thermostat schedule. In addition, Figure 6.3 depicts a large error on the first day which was improved showing the fast learning rate of the online algorithm.

Moreover, a comparison between the online MILP and thermostat schedule costs is presented in Figure 6.4.

Note that the performance of online MILP improves over time as the algorithm starts learning. Moreover, the simulation results confirm that the cost for the online MILP scheduling scheme is lower than the the offline MILP schedule for the whole month. However, for a cooling system with ten or hundred of similar loads, the saving of electricity cost by adopting the proposed approach is significant. Since the ideal temperature is changing randomly every hour and the real-time price is not known a priori, the user can chose the weighting factor c_T randomly from a uniform distribution. However, $\hat{c}_T$ does not track the user's specified c_T as shown in Fig6.5.

Despite ignoring user's choice, Figs 6.4, 6.3 and 6.5 imply that designing the weighting factor using online DA algorithm significantly improves the overall performance of the system. In addition, Figure 6.6 demostrates the good performance of online DA algorithm given that the regret is much smaller than the theoretical regret bound.

7. Remarks

This chapter presents an optimal power management strategy in cooling systems in the presence of unknown uncertainties. Blending ideas from online learning and MILP is essential to control the cooling system operation. In this venue, the chapter describes an optimization-based modeling technique for thermostatically controlled cooling load via MILP. Moreover, the weighting factor is adjusted as the real cost is revealed. Simulation results show that the online MILP significantly improves the system performance

by accommodating for unknown parameters through designing weight factor. Therefore, integrating optimal power curtailment strategies and the proposed algorithm in future smart grid systems can potentially lead to improved operation of the grid in presence of uncertainties.

Future research will consider more complex systems including the operation of multiple loads in the cooling systems and distributed energy management in a residential neighborhood with dynamic pricing and demands.

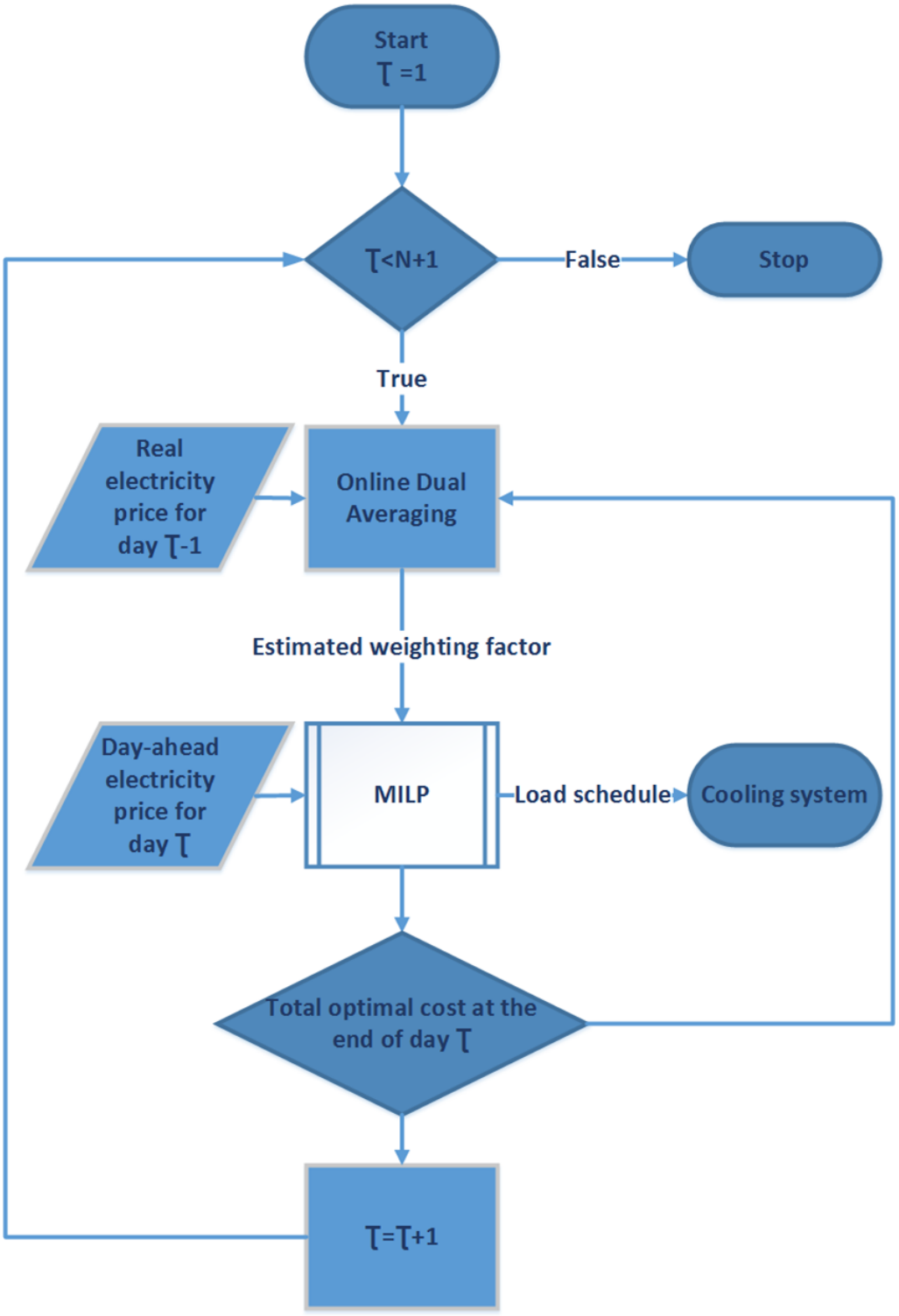


FIGURE 5.1. Power mangament process

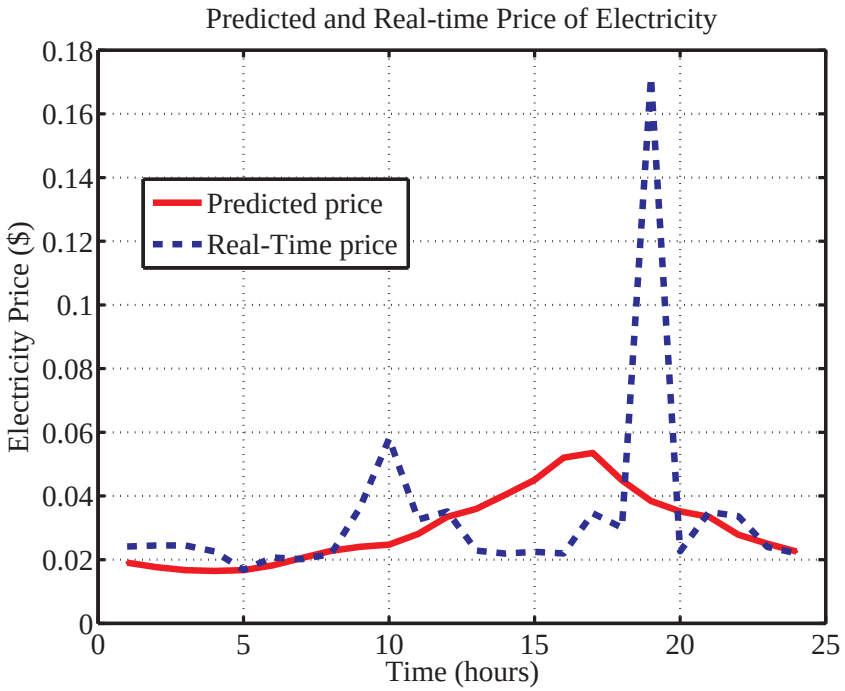


FIGURE 6.1. The predicted and real-time price in Zone I of Illinois state on October 7^{th} of 2012 [**43**].

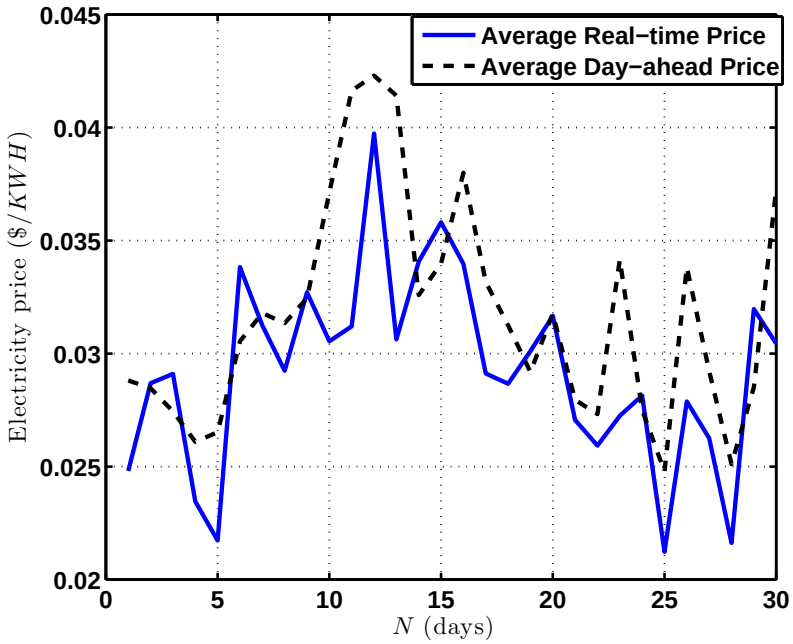


FIGURE 6.2. The average predicted and real-time price per day in Zone I of Illinois state [**43**].

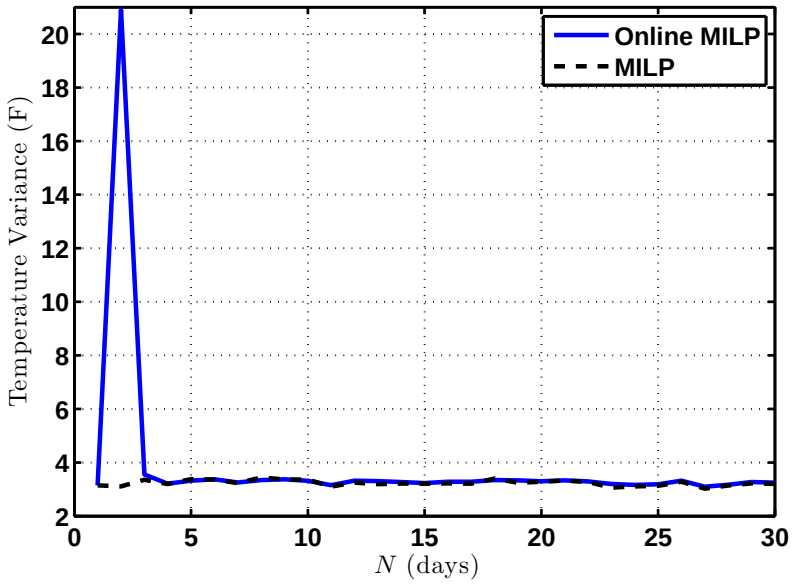


FIGURE 6.3. Temperature tracking error of chilling load with online MILP and thermostat schedule for a month.

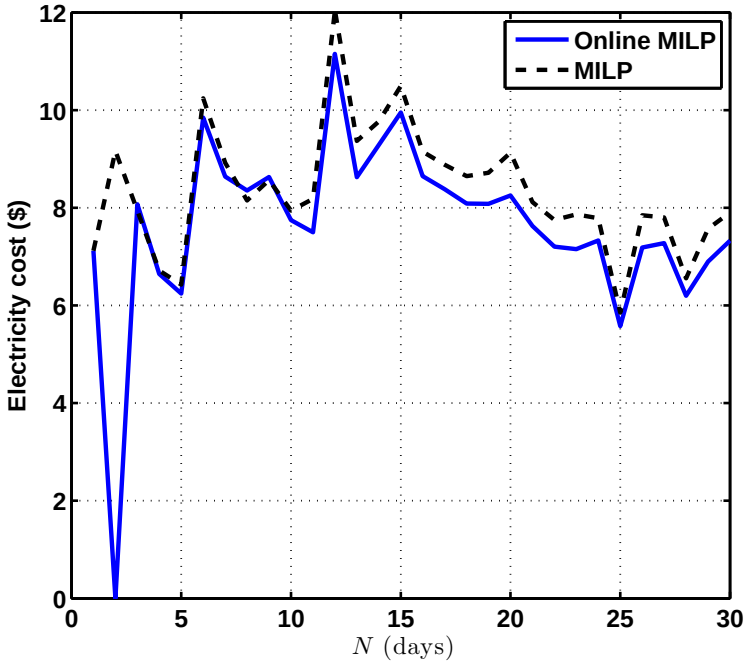


FIGURE 6.4. Electricity cost of each day.

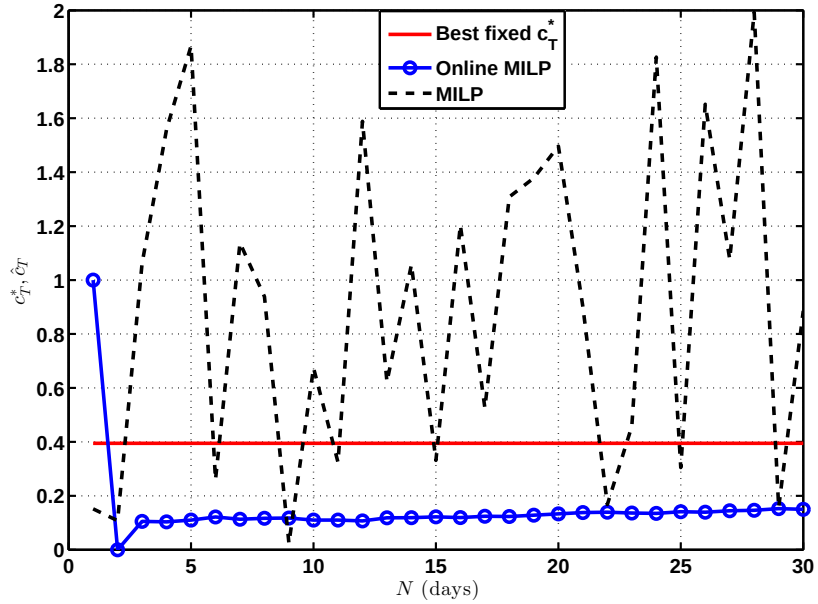


FIGURE 6.5. The time history for random satisfaction coefficient, estimated satisfaction coefficient, and the constant satisfaction coefficient.

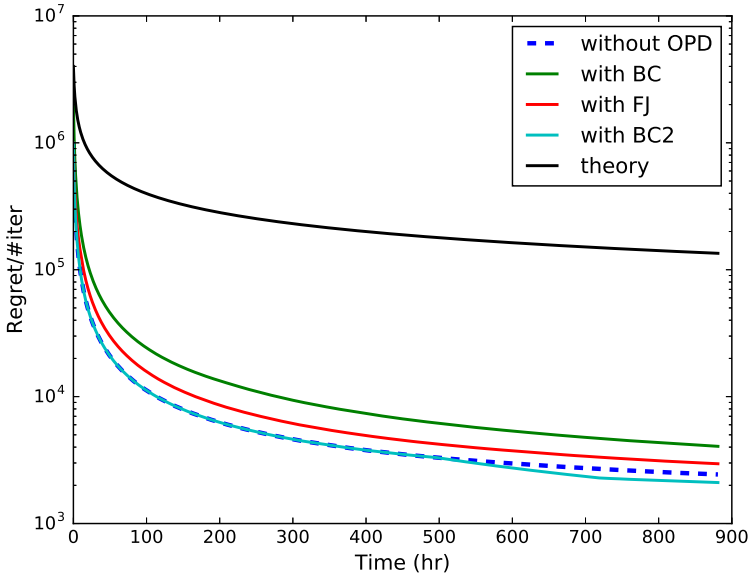


FIGURE 6.6. The time history for random satisfaction coefficient, estimated satisfaction coefficient, and the constant satisfaction coefficient.

Part 2

Large Scale Systems

II. Background and Preliminaries

In this section, we review basic concepts from graph theory and online algorithms, as well as the relevant assumptions for our analysis.

For the column vector $v \in \mathbb{R}^p$, v_i or $[v]_i$ denotes the ith element and e_i denotes the column vector which contains all zero entries except $[e_i]_i = 1$. The vector of all ones is denoted as $\mathbf{1}$. For matrix $M \in \mathbb{R}^{p\times q}$, $[M]_{ij}$, or simply M_{ij}, denotes the element in its ith row and jth column. The family of probability vectors is denoted by Ω and contains all non-negative vectors $\sigma \in [0,1]^n$ such that $\sum \sigma_i = 1$. A row stochastic matrix P is a non-negative matrix with rows in Ω. Moreover, the ergodic coefficient for a stochastic matrix $Q \in \mathbb{R}^{n\times n}$ is given by

$$\tau(Q) = 1 - \min_{i,j\in[n]} \sum_{k=1}^{n} \min\{Q_{ik}, Q_{jk}\}. \tag{II.1}$$

A doubly stochastic matrix P is a non-negative matrix with $\sum_{i=1}^n P_{ij} = \sum_{j=1}^n P_{ij} = 1$. A time varying matrix is denoted by P^t and a (backward) sequence of time varying matrices is presented by $P^{(t,0)} = P^t P^{t-1} \cdots P^0$. For any positive integer n, the set $\{1, 2, ..., n\}$ is denoted by $[n]$. The 2-norm,1-norm and infinity norm are denoted by $||.||$, $||.||_1$, and $||.||_\infty$, respectively; the dual norm of a vector u in the normed space with the norm $\|.\|$ is defined as $||u||_* = \sup_{||v||=1} \langle u, v\rangle = ||u||$, where $\langle\cdot,\cdot\rangle$ denotes the underlying inner product.

We denote the largest, second largest, and smallest singular values of $Q \in \mathbb{R}^{n\times n}$ by $\sigma_1(Q)$, $\sigma_2(Q)$ and $\sigma_n(Q)$, respectively. A function $f : \chi \to \mathbb{R}$ is called L-Lipschitz continuous if there exists a positive constant L for which

$$|f(u) - f(v)| \le L\|u - v\| \text{ for all } u, v \in \chi. \tag{II.2}$$

Although the dual of the 2-norm is the 2-norm itself, we derive some of the bounds in our subsequent analysis using the notion of the dual norm. The main reason is the connection between the Lipschitz continuity of a function (in the native norm) and the boundedness of its subgradient (by the Lipschitz constant) in the dual norm.

Moreover, the Bergman divergence is expressed as $\mathcal{B}_\phi$ and for a differentiable, strictly convex function ϕ, we have :

$$\mathcal{B}_\phi(x, y) = \phi(x) - \phi(y) - \nabla\phi(y).(x - y).$$

The standard k-simplex is denoted by $\Delta^k = \left\{ v \in \mathbb{R}^{k+1} | \sum_{i=0}^{k} v_i \; v_i \geq 0 \; \forall i \right\}$.

II.1. Graphs. A succinct way to represent the interactions of dynamic agents, e.g., sensors, over a network is through a graph. A weighted directed graph $\mathcal{G} = (V, E, W)$ is defined by a node set V with cardinality n, the number of nodes in the graph representing the agents in the network, and an edge set E comprising of pairs of nodes which represent the agents' interactions, i.e., agent i affects agent j's dynamics if there is an edge from i to j, denoted as $(i, j) \in E$. In addition, a function $W : E \to \mathbb{R}$ is given that associates a weight $w_{ji} \in W$ to every edge $(i, j) \in E$. Moreover, $\text{dist}(i, j)$ denotes the minimum number of edges of any directed path from node i to node j. The neighborhood of node i is defined as the set $N_i = \{j | (j, i) \in E\}$ and the adjacency matrix $A(\mathcal{G})$ is a matrix representation of $\mathcal{G}$ with $[A(\mathcal{G})]_{ji} = w_{ji}$ for $(i, j) \in E$ and $[A(\mathcal{G})]_{ji} = 0$, otherwise. A graph $\mathcal{G}$ is strongly connected if there exists a directed path between every pair of distinct vertices. For a graph $\mathcal{G}$, d_i is the weighted in-degree of i, defined as $d_i = \sum_{\{j|(j,i)\in E\}} w_{ij}$. In addition, $L(\mathcal{G}) = \Delta(\mathcal{G}) - A(\mathcal{G})$ is called the graph Laplacian where $\Delta(\mathcal{G})$ is the diagonal matrix of d_i's. Based on the construction of weighted directed graph Laplacian, every graph $\mathcal{G}$ has a right eigenvector of $\mathbf{1}$ associated with eigenvalue $\lambda = 0$ [**121**].

There are many families of graphs that are often used to model networks of practical interest. In this chapter , we use path and random graphs for some of our simulations. Specifically, Erdős-Rényi random graphs with edge probability p are constructed by having an edge $(i, j) \in E$ in the graph with probability p for all possible edges. A random tree is a particular realization of a random graph that is minimally connected and a random k-regular graph is a random graph in which $d_i = k$ for all vertices $i \in V$. In the path graph, $(i, j) \in E$ if and only if $|i - j| = 1$.

If there exists a directed path between every pair of distinct vertices, the graph $\mathcal{G}$ is referred to as strongly connected. In this work, we assume that the inter agent communication between the agents constitute a strongly connected graph, ensuring information flow among the agents.

The the interactions among nodes is presented by a time-varying graph $G^{(t)}(V^{(t)}, E^{(t)})$ defined by the nodes set $V^{(t)}$ and edges set $E^{(t)}$, where the number of nodes in the graph is $|V^{(t)}| = n_t$ nodes at time t. The underlying communication topology is expressed by $A^{(t)} \in [0,1]^{n\times n}$ where the elements of $A^{(t)}$ represent the probability of existing an edge between the corresponding nodes, i.e., $A_{ij}^{(t)} = 1$ if $(i,j) \in E^{(t)}$ and $A_{ij}^{(t)} = 0$ otherwise. Note that $A_{ii}^{(t)} = 1$ for all $i \in [n_t]$. The neighborhood set of node i is defined as $N^{(t)}(i) = \{j \in V^{(t)} | (i,j) \in E^{(t)}\}$.

II.2. Regret. Regret is one measure of performance for learning algorithms. In the online optimization setting, an algorithm is used to generate a sequence of decisions $\{x(t)\}_{t=1}^{T}$. The number of iterations is denoted by T which is unknown to the online player. At each iteration t, after committing to $x(t)$, a previously unknown convex cost function f_t is revealed, and a loss $f_t(x(t))$ is incurred. The goal of the online algorithm is to ensure that the time average of the difference between the total cost and the cost of the best fixed decision $x^* = \text{argmin} \sum_{t=1}^{T} f_t(x)$ is small. The difference between these two costs over $t = 1, 2, ..., T$, iterations is called the regret of the online algorithm, i.e.,

$$R_T(x^*, x) = \sum_{t=1}^{T} \left(f_t(x(t)) - f_t(x^*) \right). \tag{II.3}$$

An online algorithm performs well if its regret grows sub-linearly with respect to the number of iterations, i.e.,

$$\lim_{T\to\infty} R_T/T = 0.$$

This implies that the average loss of the algorithm tends to the average loss of the best fixed strategy in hindsight independent of the uncertainties associated with the global cost.[5] We

[5]The notion of regret is often received with a degree of skepticism upon initial encounter. The basic idea is that if there is a positive lower bound between the cost incurred by the algorithm and the best fixed decision in hindsight, then the regret will grow linearly. A sublinear regret implies that the algorithm has learned to match the performance of the best fixed decision in hindsight.

refer to **[138, 139, 140, 141]** for further discussions on online algorithms and their regret analysis.

The general definition of regret is presented in (II.3) for a single decision-making unit. In order to analyze the performance of *distributed online algorithms* two variations of the notion of regret are introduced. First is the regret due to agent i's decision,

$$R_T(x^*, x_i) = \sum_{t=1}^{T} \left(f_t(x_i(t)) - f_t(x^*) \right), \tag{II.4}$$

which is the cumulative penalty agent i incurs due to its local decisions $\{x_i(t)\}_{t=1}^T$ on the global cost sequence $\{f_t\}$. Second is the regret based on the *running average* of the decisions $\{x_i(t)\}_{t=1}^T$,

$$R_T(x^*, \widetilde{x}_i) = \sum_{t=1}^{T} \left(f_t(\widetilde{x}_i(t)) - f_t(x^*) \right), \tag{II.5}$$

where $\widetilde{x}_i(T) = \frac{1}{T} \sum_{t=1}^{T} x_i(t)$.

CHAPTER 4

Online Distributed Optimization on Dynamic Networks

1. Summary

This chapter presents a distributed optimization scheme over a network of agents in the presence of cost uncertainties and over switching communication topologies. Inspired by recent advances in distributed convex optimization, we propose a distributed algorithm based on a dual sub-gradient averaging. The objective of this algorithm is to minimize a cost function cooperatively. Furthermore, the algorithm changes the weights on the communication links in the network to adapt to varying reliability of neighboring agents. A convergence rate analysis as a function of the underlying network topology is then presented, followed by simulation results for representative classes of sensor networks.

2. Introduction

The past decade has witnessed successful applications of networked systems in areas ranging from environmental monitoring, robotics, target recognition, air traffic control, to industrial and manufacturing automation. By increasing the size and complexity of networked systems, decentralized optimization schemes are desired for reducing data transmission rates and ensuring robustness in the presence of local failures. These methods are particularly relevant when there is a lack of access to centralized information by individual agents.

In recent years, there has grown an extensive literature on distributed convex optimization [**53, 54, 90**] and the adaptation and monitoring of the underlying network has become of increasing interest [**55, 56, 57**]. Moreover, agent's communication range or disturbances may cause the underlying network topology to change dynamically. In this direction, a class of distributed sub-gradient algorithms for convex optimization has been

developed [**58, 59, 60, 111, 113**]. In these works, local convex cost functions are assumed to be known while the topology of network is allowed to vary.

In addition to uncertainties in the network's structure, the environment can also affect the corresponding cost functions. For such scenarios, traditional optimization approaches become unsuitable. One approach to improve the robustness of algorithms for convex optimization is via stochastic methods [**61, 62, 112**], where the probability distribution of uncertain variable is known *a priori*. One such approach has been pursued by Duchi *et al.* [**59**] who approached this problem using a stochastic sub-gradient method where the distribution of sub-gradients is known *a priori*.

Despite its many successes, stochastic optimization-based methods do not explicitly address the dynamic aspect of the problem in an unknown environment. Online learning is an extension of stochastic optimization where the uncertainty in the system is demonstrated by an *arbitrarily* varying cost function. In particular, at the time the relevant decision is made the cost function is assumed to be unknown, without probabilistic assumptions, to the decision-maker. Such learning algorithms have had a significant impact on modern machine learning [**63, 64, 141**]. One standard metric to measure the performance of these online algorithms is called *regret*. Regret measures the difference between the incurred cost and the cost of the *best fixed decision in hindsight*. An online algorithm is then declared "good" when its regret is sub-linear.

Distributed online optimization and its applications in multi-agent systems has not been studied at large by the systems and control community. Yan *et al.* in [**94**] introduced a decentralized online optimization based on a sub-gradient method in which the agents interact over a weighted strongly connected directed graph. Considering an undirected path graph with a fixed-radius neighborhood information structure, Raginsky *et al.* [**65**] proposed an online algorithm for distributed optimization based on sequential updates, proving a regret bound of $O(\sqrt{T})$. In [**95**], we proposed an extension to the work of Duchi *et al.* [**59**] on distributed optimization with convergence rate of $O(\sqrt{T}\log T)$ to an online setting. In addition, an improved regret bound of $O(\sqrt{T})$ has been derived for strongly

connected networks, also highlighting the dependence of the regret on the connectivity of the underlying network.

We note that the aforementioned works do not exploit a dynamic weight selection procedure to improve the performance of the corresponding distributed algorithms. In systems and control literature, certain metrics have been used for designing *adaptive* mechanisms for networks based on centralized [**66, 67, 68**] and distributed [**55, 69**] strategies. Chapman *et al.* [**70**] proposed an online distributed algorithm for re-weighting the network edges in order to dampen the effect of external disturbances on the system. Dynamic weight selection is also favorable in the area of sensor networks and distributed estimation due to power and data rate constraints as well as failure modes of the inter-sensor communication links [**71, 72, 73**].

In this chapter , we consider two types of uncertainties in the networked systems corresponding to disturbances in cost functions and the network structure. An adaptive algorithm for distributed optimization over fixed networks is proposed and further extended to switching graphs. The main assumption used for implementing this algorithm is that the local cost function and its sub-gradient are observable at each node and can be shared with the neighboring nodes in the network.

The contribution of this chapter is threefold. First, we present the Distributed Weighted Dual Averaging (DWDA) algorithm [**95**] for distributed optimization over networks. A distributed dynamic weight selection method based on an online weighted majority approach [**74, 75**] is then embedded in the DWDA algorithm allowing the weights on the network's edges to adaptively change in order to optimize the information diffusion in the network. The proposed algorithm is inspired by the Distributed Dual Averaging (DDA) algorithm [**59**]. Second, DWDA is applied on switching networks capturing the uncertainties in communication links. Third, the DWDA is further extended to Online Distributed Weighted Dual Averaging (Online-DWDA) algorithm which takes into account the uncertainties in cost functions and unavailability of reliable statistics on the noise characteristics. We then

proceed to derive regret bounds that highlight the link between the adaptive weight selection and the Online-DWDA algorithm and can thus be used to design networks with good regret performance.

The organization of the chapter is as follows. In §3, the formulation for the distributed convex optimization problem over networks is presented. This is then followed by the description for the DWDA algorithm and dynamic weight selection procedure in §4 and §5; the convergence analysis of the proposed algorithm is discussed over switching topologies in §6. In §7, the distributed convex optimization problem is extended to the online setting, with applications to networked systems operating in an uncertain environment. The performance of Online-DWDA is subsequently studied using the regret analysis. In §8, we examine online distributed estimation over sensor networks, demonstrating the viability of the online approach in distributed estimation. Finally, §9 provides our concluding remarks and future directions for utilizing the online framework for system and control problems.

3. Problem Statement

In this section a distributed decision process is considered in which a large number of agents cooperatively optimize a global objective function over the network denoted by $\mathcal{G} = (V, E, W)$.

The global objective to be minimized is

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x) \quad \text{subject to } \; x \in \chi, \tag{3.1}$$

where $f_i(x) : \mathbb{R}^d \to \mathbb{R}$ is a convex cost function associated with agent $i \in V$ and $\chi \subseteq \mathbb{R}^d$ is a closed convex set. The global optimization problem will be solved *locally* by each agent i via the local decision variable $x_i \in \chi$.

4. Weighted Dual Averaging

In order to solve the optimization problem (3.1), we adapt Nesterov's dual averaging algorithms [**110**] and our preliminary results on the Distributed Weighted Dual Averaging

(DWDA) algorithm [**95**], that in turn is inspired by [**59**]. The DWDA algorithm sequentially updates the *local* $x_i(t)$ and a *working* variable $y_i(t)$ for each agent i. The update itself is based on a provided local sub-gradient of the loss $f_i(x_i(t))$ denoted as $g_i(t)$. The centralized form of the dual averaging algorithm appears as a sub-gradient decent method followed by a projection step onto the constraint set χ, specifically,

$$y(t+1) = y(t) + g(t), \tag{4.1}$$

where $g(t) \in \partial f(x(t))$. Then

$$x(t+1) = \Pi_{\chi}^{\psi}\left(y(t+1), \alpha(t)\right), \tag{4.2}$$

where $\Pi_{\chi}^{\psi}(\cdot)$ is a regularized projection onto χ, to be formally defined shortly. The Distributed Weighted Dual Averaging (DWDA) algorithm is presented in Algorithm 3. The projection function used in this algorithm is defined as

$$\Pi_{\chi}^{\psi}(y(t), \alpha(t)) = \arg\min_{x \in \chi} \left\{ \langle y(t), x \rangle + \frac{1}{\alpha(t)} \psi(x) \right\}, \tag{4.3}$$

where $\alpha(t)$ is a non-increasing sequence of positive functions and $\psi(x) : \chi \to \mathbb{R}$ is a proximal function. The standard dual averaging algorithm uses proximal function $\psi(x)$ to avoid undesirable oscillations in the projection step. Without loss of generality, ψ is assumed to be strongly convex with respect to $\|.\|$, $\psi(x) \geq 0$, and $\psi(0) = 0$.

Algorithm 3: Distributed Weighted Dual Averaging (DWDA)

for $t = 1$ **to** T **do**
  Evaluate $f(t) = \{f_i(t);$ for all $i = 1, ..., n\}$
  foreach *Agent* i **do**
    Compute subgradient $g_i(t) \in \partial f_i(x_i(t))$
    $y_i(t+1) = \sum_{j \in N(i)} P_{ji}(t) y_j(t) + g_i(t)$
    $x_i(t+1) = \prod_{\chi}^{\psi}(y_i(t+1), \alpha(t))$
    $\widetilde{x}_i(t+1) = \frac{1}{t+1} \sum_{s=1}^{t+1} x_i(s)$
  end
end

The distributed algorithm can be considered as an approximated sub-gradient descent. The approximation is attained by an agent via a convex combination of local sub-gradients

provided by its neighbors. This operation can be represented compactly as a stochastic matrix $P \in \mathbb{R}^{n\times n}$ which preserves the zero structure of the Laplacian matrix $L(\mathcal{G})$. It is clear that for all agents to have access to each cost function f_i there must be a path from every agent i to every other agent. Consequently, a minimum requirement on the underlying network is that it must be strongly connected. The distributed dynamic weight selection procedure presented in the following section constructs a row stochastic matrix P of the required form that is associated with a weighted directed graph.

5. Distributed Dynamic Weight Selection

In this section, we propose an adaptation scheme for the network weight selection in order to improve the information diffusion in line 5 of Algorithm 3 such that the communication matrix P is a row stochastic matrix with positive diagonal elements. In this proposed distributed algorithm, each agent $i \in [n]$ estimates its loss function via a convex combination of loss functions available to it by its neighboring agents. This convex combination is specified by weights w_{ij}'s on each edge $(j,i) \in E$ and w_{ii}, respectively. The edge re-weighting problem parallels the Weighted Majority (WM) algorithm [**74**]. The context of the WM algorithm is the presence of $|N_i|+1$ experts, and the associated cost $h_j(t)$ assigning a loss value to expert $j \in \{N_i, i\}$, where $0 \leq h_j(t) \leq 1$. At each time-step, agent i selects a probability distribution $q(t)$ over the $|N_i|+1$ experts, i.e., $q(t) \in \mathcal{Q}_i = \{q \in \Omega | q_j = 0 \text{ for } j \notin \{N_i, i\}\}$, in order to minimize $l_{t,i}(q) = \sum_{j\in\{N(i),i\}} q_j(t)h_j(t)$. The regret for agent i for the WM algorithm is then defined as

$$L_T(q^*, q) = \sum_{t=1}^{T} l_{t,i}(q) - \sum_{t=1}^{T} l_{t,i}(q^*), \tag{5.1}$$

where

$$q^* = \text{argmin}_{q\in\mathcal{Q}_i'} \sum_{t=1}^{T} f_t(q_j)$$

with $\mathcal{Q}_i' = \{q \in \mathcal{Q}_i | q_j \in \{0,1\}\}$, and the best fixed strategy q^* is the best expert $j \in \{N_i, i\}$ in hindsight. Consequently, (5.1) is of the same form as (II.3).

The general form of the WM algorithm is presented in **[75]** as the Online Allocation (OA) algorithm that is applicable to any bounded loss function over general decision and outcome spaces. Based on the OA algorithm, the regret for each agent $i \in [n]$ is bounded as

$$L_T \leq M\left(\sqrt{2T\ln\left(|N_i|+1\right)}+\ln\left(|N_i|+1\right)\right), \tag{5.2}$$

where M is the upper bound on the loss function $h_j(t)$ for all $j \in \{N_i, i\}$. Since the regret in (5.2) is sub-linear over time, the weight allocation performs as well as the best strategy in hindsight.

A Distributed Online Allocation (DOA) algorithm is proposed based on the OA algorithm where agent $i \in [n]$ specifies the weights w_{ij}'s associated to each edge $(j,i) \in E$ as well as the weight on the self-loop w_{ii}. The DOA algorithm presented in Algorithm 4 is embedded in Algorithm 3 at each iteration.

Algorithm 4: Distributed Online Allocation (DOA)

foreach *Agent* i **do**
  Choose $\beta \in [0,1]$ and initial weight vector $\mathbf{w}_i(t) = \mathbf{1}$
  Let $q(t) = \frac{1}{n}\mathbf{w}_i(t)$
  for $t = 1$ **to** T **do**
    Adversary reveals $f(t) = \{f_j(t);\ \text{for all } j \in \{N(i), i\}\}$
    Suffer loss $l_{t,i} = \sum_{j\in\{N(i),i\}} q_j(t) f_j(t)$
    foreach *Agent* $j \in [n]$ **do**
      if $j \in \{N(i), i\}$ **then**
        $w_{ij}(t+1) = w_{ij}(t)\beta^{f_j(t)}$
      else
        $w_{ij}(t+1) = w_{ij}(t)$
      end
    end
    $q(t+1) = \frac{\mathbf{w}_i(t+1)}{\sum_{j\in\{N(i),i\}} w_{ij}(t+1)}$
  end
end

In the distributed optimization process considered, each agent decides on the weights associated with the information *received* from its neighboring agents. This information is based on the neighbor's local loss function. Intuitively, the algorithm places more weight on the link associated with the neighboring agent that has a higher confidence in its decision.

The positive diagonal entries represent the self-confidence of each agent and is updated based on the local loss.

In networks with fixed topology, the communication matrix $P(t)$ preserves its zero structure for all time t. In addition, the non-zero elements in each row of the communication matrix P is specified by line 14 in Algorithm 4:

$$P_{ij}(t) = \begin{cases} q_j(t) & \text{for } j \in \{N_i, i\} \\ 0 & \text{otherwise} \end{cases}. \tag{5.3}$$

Since for each agent i, $q(t) \in \mathcal{Q}_i$ is a probability distribution, the communication matrix $P(t)$ will be row stochastic at every time step. The weighted graph Laplacian can then be formed as

$$L(\mathcal{G}(t)) = I - P(t). \tag{5.4}$$

In addition, note that since the graph is strongly connected, the communication matrix P is 1-irreducible ([**76**]; Corollary 4) and given positive diagonal elements, it is in-decomposable and aperiodic (SIA). In the following section, the DOA algorithm is extended to construct a row stochastic communication matrix $P(t)$ for directed switching graphs with time-varying edge sets.

5.1. Switching Topologies. The network topology may change dynamically due to disturbances or communication range limitations. In this section we apply the dynamic weight selection procedure discussed in §5 to switching topologies. In this chapter we assume that the union of directed topologies $\mathcal{G}^{\cup_{i=0}^{\delta-1}} = \bigcup_{i=0}^{\delta-1} \mathcal{G}^i$ over some fixed uniform intervals δ, with $\delta \geq 1$ a positive integer, is strongly connected. We note that the communication matrix of $\mathcal{G}^{\cup_{i=0}^{\delta-1}}$ can be presented as

$$P^{\cup_{i=0}^{\delta-1}} = P^0 + P^1 + \cdots + P^{\delta-1}. \tag{5.5}$$

Thus, each row of the communication matrix P over switching topologies is specified by line 14 in Algorithm 4 as

$$P_{ij}(t) = \begin{cases} q_j(t) / \left(\sum_{k \in \{N_i^t, i\}} q_k(t) \right) & \text{for } j \in \{N_i^t, i\} \\ 0 & \text{otherwise} \end{cases}, \tag{5.6}$$

where $q(t) \in Q_i^t = \left\{ q \in \Omega | q_j = 0 \text{ for } j \notin \{N_i^t, i\} \right\}$ and is a probability distribution. Note that the communication matrix $P(t)$ will be row stochastic at every time step and thus the weighted graph Laplacian is the same as in (5.4). Since the graph $\mathcal{G}^{\cup_{i=0}^{\delta-1}}$ is strongly connected and $P^{\cup_{i=0}^{\delta-1}}$ has positive diagonal elements, the communication matrix $P^{\cup_{i=0}^{\delta-1}}$ is SIA ([**76**]; Corollary 4). These properties of communication matrices will be subsequently employed in the convergence analysis of the DWDA algorithm.

6. Convergence Analysis

Before presenting the convergence analysis of the distributed optimization algorithm, a few preliminary remarks and assumptions are in order. We assume that each convex function f_i is positive and L-Lipschitz with respect to $\|.\|$. Assuming that $\left\{ P^t \right\}$ is SIA, there exists a vector $\pi \in \Omega$ [**77**], such that

$$\pi_j = \sum_{i=1}^{n} \pi_i P_{ij}^t \text{ for all } t \in [T], \tag{6.1}$$

where π_i is referred to as the weighting factor for agent i.

In order to take advantage of the properties of the standard weighted dual averaging in our regret analysis, the sequences $\bar{y}(t)$ and $\bar{g}(t)$ are defined as

$$\bar{y}(t) = \sum_{i=1}^{n} \pi_i y_i(t), \text{ and } \bar{g}(t) = \sum_{i=1}^{n} \pi_i g_i(t), \tag{6.2}$$

signifying the (network-level) weighted average of dual variables and subgradients in the DWDA algorithm, respectively. Therefore, based on (6.1) and (6.2),

$$\bar{y}(t+1) = \sum_{i=1}^{n} \pi_i \left\{ \sum_{j=1}^{n} P_{ij}^t y_j(t) + g_i(t) \right\} = \sum_{j=1}^{n} y_j(t) \sum_{i=1}^{n} \pi_i P_{ij}^t + \bar{g}(t)$$

$$= \sum_{j=1}^{n} y_j(t)\pi_j + \bar{g}(t) = \bar{y}(t) + \bar{g}(t), \tag{6.3}$$

which is analogous to the dual averaging update (4.1). Thus, the following update rule is introduced which is analogous to the standard dual averaging algorithm projection step (4.2), where the primal variable is updated as

$$\phi(t+1) = \Pi_{\chi}^{\psi}(\bar{y}(t+1), \alpha(t)). \tag{6.4}$$

The performance analysis of the distributed optimization and adaptive weight selection can now be presented.

The following result by Duchi *et al.* implies that after T iterations of Algorithm 3, each agent's error in the evaluation of total cost is bounded by the error due to Dual-Averaging method.

THEOREM 6. **[59]** *Given the sequences $x_i(t)$ and $y_i(t)$ generated by lines 5 and 8 in* Algorithm 3*, for all $i \in [n]$ with proximal function ψ and $\alpha(t) > 0$, we have*

$$\frac{1}{T}\sum_{t=1}^{T} f(x_i(t)) - f(x^*) \leq \frac{L^2}{2}\sum_{t=1}^{T}\alpha(t-1) + \frac{1}{\alpha(T)}\psi(x^*)$$
$$+ \frac{L}{T}\sum_{t=1}^{T}\alpha(t)(\|\bar{y}(t) - y_i(t)\|_* + \frac{2}{n}\sum_{i=1}^{n}\|\bar{y}(t) - y_i(t)\|_*). \tag{6.5}$$

The last two terms on the right hand side of (6.5) represent the error due to the network which is defined as the deviation of local dual variable y_i from the weighted average of dual variables $\bar{y}$ over the network. Lemma 15 in the Appendix imposes an upper bound on the effect of network topology associated with $\|\bar{y}(t) - y_i(t)\|_*$ as

$$\|\bar{y}(t) - y_i(t)\|_* \leq L \sum_{k=0}^{t-2} \sum_{j=1}^{n} \left| P_{ij}^{(t-1,k+1)} - \pi_j \right| + 2L. \tag{6.6}$$

Inequality (6.6) highlights the importance of the underlying network topology through the communication matrix P^t and its products. Note that the network effect is analogues to the consensus problems [**121**]. Therefore we proceed to extend the distributed optimization algorithm to switching graphs in the following subsection and provide a sub-linear convergence rate for the DWDA algorithm.

6.1. Switching Topologies. In this section we employ the weak ergodicity of inhomogeneous Markov chains to reason about the convergence of the DWDA algorithm. This property implies that the product of stochastic SIA matrices converges exponentially to a rank-one matrix of the form $\mathbf{1}\pi^T$ as $t \to \infty$, where $\pi \in \Omega$. Applying the following result from [**78**], it thus suffices to show that $P^{(\delta-1,0)}$ is SIA.

LEMMA 7. [**78**] *Let $m \geq 1$ be a positive integer and P^τ be non-negative matrices with positive diagonal elements for $\tau = 0, 1, \ldots, m$. Then, $P^m P^{m-1} \ldots P^0 \geq \mu^m (P^0 + P^1 + \cdots + P^m)$, where $\mu > 0$ is specified by the diagonal elements of matrices P^τ for all $\tau = 0, 1, \ldots, m$.*

From Lemma 7 and (5.5), $P^{(\delta-1,0)}$ is bounded below by an SIA matrix. Moreover, we note that $P^{(\delta-1,0)}$ is also a stochastic matrix, thus it must be SIA. Therefore, based on the weak ergodicity of inhomogeneous Markov chains, the product

$$P^{(k\delta-1,(k-1)\delta)} \ldots P^{(2\delta-1,\delta)} P^{(\delta-1,0)}$$

converges exponentially to a rank-one matrix of the form $\mathbf{1}\pi^T$ as $t \to \infty$, and based on Theorem 1 of [**77**], we have

$$\left| P^{(k\delta-1,0)} - \pi_j \right| \leq \gamma^{\lfloor \frac{k}{\nu} \rfloor}, \tag{6.7}$$

where

$$\gamma = \max_{\nu \geq 1} \left\{ \tau(P^{(\delta\nu-1,0)}) < 1 \right\}. \tag{6.8}$$

Note that the maximization is over all realizations of the sequence $P^{(\delta\nu-1,0)}$ and ν is bounded as stated in the following proposition.

PROPOSITION 8. *Consider a set $\mathcal{P}$ of stochastic matrices with positive diagonal elements, representing arbitrarily strongly connected topologies over n nodes, i.e., $P^k \in \mathcal{P}$ for all positive integers k. Then, there exists an integer ν, $1 \leq \nu \leq n-1$, for which if the sequence $Q = P^{(m+\nu-2,m)}$ of matrices in $\mathcal{P}$ is not scrambling, $P^{m+\nu-1}Q$ is scrambling.*

PROOF. Let $Q_1 = P^m$ and $Q_2 = P^{m+1}P^m$. Then every entry of Q_2 is represented as

$$[Q_2]_{ij} = \sum_{k=1}^{n} \left[P^{m+1}\right]_{ik} [P^m]_{kj}.$$

Let $E^{\cup_{i=m}^{m+1}}$ represent the edge set of the union of directed graphs associated with P^m and P^{m+1}. Since $[P^m]_{ii} > 0$ for all $i \in [n]$ and integer $m \geq 1$, the entry $[Q_2]_{ij}$ is positive if $(j,i) \in E^{\cup_{i=m}^{m+1}}$ or if there exists a node $k \in [n]$ such that $(j,k) \in E^m$ and $(k,i) \in E^{m+1}$. Thus, the corresponding zero entry of Q_1 that has one of the aforementioned properties will be positive in Q_2. By induction, it follows that the entry of $[Q_\nu]_{ij}$ will be positive if $(j,i) \in E^{\cup_{i=m}^{m+\nu-1}}$, or if there exists a set of nodes $\{k_m, k_{m+1}, \ldots, k_{m+\nu-2}\}$ such that $\{(j,k_m), (k_m,k_{m+1}), \ldots, (k_{m+\nu-2},i)\} \in E^m \times E^{m+1} \times \cdots \times E^{m+\nu-1}$. Therefore, for each row i of Q_ν, all entries will be positive when

$$\nu_i = \max_{j \in [n], m \geq 1} \{\text{dist}(j,i) \text{ for } \mathcal{G}^m = (V, E^m, W)\}. \tag{6.9}$$

Note that the maximization in (6.9) is over all possible strongly connected graphs with the directed cycle graph representing the worst case with $\nu_i = n-1$. Since every element of any row of the sequence $Q = P^{(m+\nu-2,m)}$ of matrices in $\mathcal{P}$ is positive, the matrix Q is scrambling and thus $1 \leq \nu \leq n-1$. □

We also note that the fixed topology is a special case of switching graphs with $\delta = 1$ in (5.5). Moreover, Proposition 16 of the Appendix presents a less conservative bound on ν for fixed topologies.

Now, we can state the following theorem for the rate of convergence of DWDA over switching graphs.

THEOREM 9. *Given the sequences $x_i(t)$ and $y_i(t)$ generated by lines 5 and 8 in* Algorithm 3, *for all $i \in [n]$ with $\psi(x^*) \leq R^2$ and $\alpha(t) = k/\sqrt{t}$, we have*

$$\frac{1}{T}\sum_{t=1}^{T} f(x_i(t)) - f(x^*) \leq \left(\frac{R^2}{k} + kL^2\left(\frac{6n}{1-\gamma} + 6n\delta\nu + 1\right)\right)\frac{1}{\sqrt{T}}, \tag{6.10}$$

where $\gamma < 1$ is a function of the ergodicity of the communication matrix (see (6.8)) while ν is a measure of network connectivity and is bounded by the diameter of the graph $\mathcal{G}^t$ (see also Proposition 8). In addition, $k > 0$ is an arbitrary constant and $\delta \geq 1$ is a positive integer as presented in (5.5).

PROOF. Based on (6.6) and (6.7) we have

$$\|\bar{y}(t) - y_i(t)\|_* \leq nL\sum_{k=1}^{t-1}\gamma^k + nL(\delta\nu - 1) + 2L, \tag{6.11}$$

and since $\gamma < 1$, (6.11) is further bounded as

$$\|\bar{y}(t) - y_i(t)\|_* \leq nL\left(\frac{1}{1-\gamma} + \delta\nu - 1\right) + 2L. \tag{6.12}$$

Therefore, the integral test on $\alpha(t) = k/\sqrt{t}$ provides a bound[1] on the first and last terms in (6.10) as

$$\frac{1}{T}\sum_{t=1}^{T} f(x_i(t)) - f(x^*) \leq \frac{kL^2}{\sqrt{T}} + \frac{\psi(\theta^\star)}{k\sqrt{T}} + \frac{6kL^2}{\sqrt{T}}\left(\frac{n}{1-\gamma} + n\delta\nu + 1\right).$$

[1]Note that $\sum_{t=1}^{T}\frac{k}{\sqrt{t}} \leq 2k\sqrt{T} - k$.

Given $\psi(x^*) \leq R^2$, the statement of the theorem now follows. □

Theorem 9 states that Algorithm 3 performs "well" as it exhibits a sub-linear convergence rate. It also highlights the importance of the underlying network topology through the parameters γ and ν. In particular, ν corresponds to the diameter of the graph as expressed in Proposition 8 and γ is proportional to the ergodic coefficient $\tau(P^t)$ of the communication matrix P^t as formed by Algorithm 3. The ergodic coefficient bounds the second largest eigenvalue of P^t, $\lambda_2(P^t)$, as $|\lambda_2(P^t)| \leq \tau(P^t) < 1$. Thus, based on (5.4), $1 - \lambda_2(P^t) = \lambda_{n-1}(\mathcal{G}^t)$ where $\lambda_{n-1}(\mathcal{G}^t)$ is the second smallest eigenvalue of the weighted graph Laplacian $L(\mathcal{G}^t)$ and a well known measure of network connectivity. Consequently, high network connectivity promotes good performance of the proposed algorithm.

In the following section we study the effect of the proposed dynamic weight selection on the network connectivity and the convergence rate (6.10).

6.2. Adaptive Weight Selection. In this section, we show that embedding Algorithm 4 within Algorithm 3 improves the network information flow and the speed of convergence in (6.10). To this end the following result provides a bound on the ergodic coefficient.

THEOREM 10. *Suppose the sequence $q_j(t)$ generated by line* 14 *of Algorithm* 4 and *the communication matrices P^t are constructed by (5.6). Then*

$$\tau(P^{(\delta\nu-1,0)}) \leq 1 - \frac{n}{(\max_{i\in[n],t\in[\delta\nu-1]} |N_i^t| + 1)^{\delta\nu}},$$

where ν is a measure of network connectivity and is bounded by the diameter of the graph $\mathcal{G}^t$ (see also Proposition 8). In addition, $\delta \geq 1$ is a positive integer as presented in (5.5).

PROOF. Based on line 14 of Algorithm 4, we have that for all $k \in \{N_i, i\}^t$,

$$q_k(t) = \frac{\beta^{-\left(\sum_{s=1}^{r_{k,i}(t-1)} f_k(s)\right)}}{\sum_{j\in\{N_i^{t-1},i\}} \beta^{-\left(\sum_{s=1}^{r_{j,i}(t-1)} f_j(s)\right)}}, \tag{6.13}$$

where $r_{ji}(t)$ represents the number of communication rounds through the directed edge (j,i) up to time t. Subsequently (5.6) and (6.13) imply

$$P_{ik}^t = \begin{cases} \dfrac{\beta^{-\left(\sum_{s=1}^{r_{k,i}(t-1)} f_k(s)\right)}}{\sum_{j\in\{N_i^t,i\}} \beta^{-\left(\sum_{s=1}^{r_{j,i}(t-1)} f_j(s)\right)}} & (k,i)\in E^t, \\ 0 & \text{otherwise.} \end{cases}$$

Since $\min_{j\in[n]} \sum_{s=1}^{r_{j,i}(t-1)} f_j(s) \leq \sum_{s=1}^{r_{k,i}(t-1)} f_k(s)$, we have

$$\beta^{-\left(\sum_{s=1}^{r_{k,i}(t-1)} f_k(s)\right)} \leq \beta^{-\left(\min_{j\in[n]} \sum_{s=1}^{r_{j,i}(t-1)} f_j(s)\right)}$$

and one can bound P_{ik}^t from below for all $(k,i)\in E$ as

$$P_{ik}^t \geq \frac{1}{(|N_i^t|+1)} \beta^{-C_k(t-1)}, \tag{6.14}$$

where $C_k(t) = \sum_{s=1}^{r_{k,i}(t)} f_k(s) - \min_{j\in[n]} \sum_{s=1}^{r_{j,i}(t)} f_j(s)$. Since $C_k(t-1)\geq 0$ for all $k\in[n]$ and $\beta\in[0,1]$, we have

$$P_{ij}^t \geq \frac{1}{(|N_i^t|+1)},$$

for all $t\in[T]$ and subsequently

$$P_{ij}^{(\delta\nu-1,0)} \geq \frac{1}{(\max_{i\in[n],t\in[\delta\nu-1]} |N_i^t|+1)^{\delta\nu}}.$$

Based on (II.1), the statement of the theorem now follows. □

Theorem 10 in conjunction with (6.8) imply

$$\gamma \leq 1 - \frac{n}{(\max_{i\in[n],t\in[\delta\nu-1]} |N_i^t|+1)^{\delta\nu}},$$

which proves to be a conservative bound as the DOA algorithm leads to a tighter upper bound capturing the performance of agents. In other words, based on (6.14), we can show that

$$\gamma \leq 1 - \frac{n\beta^{-J_{\delta\nu}}}{(\max_{i\in[n]} |N_i|+1)^{\delta\nu}},$$

where $J_{\delta(n-1)} = C_i(1)+C_j(2)+...+C_k(\delta\nu)$ and $i, j, \ldots, k \in [n]$. In addition, we know that $\beta^{-J_{\delta\nu}} > 1$ and $J_{\delta\nu}$ is an increasing sequence. If the agents are not performing well, $J_{\delta\nu}$ will increase and subsequently $\tau(P^{(\delta\nu,0)})$ will decrease which suggests that the DOA algorithm mitigates the effect of the network topology in (6.10). Moreover, Theorem 10 implies that the DWDA algorithm performs well for certain types of graphs such as k-regular and expander graphs where the maximum number of neighbors can be bounded.

7. Online Distributed Optimization

We now consider the effect of uncertainties in the environment on distributed decision processes where the global objective is to minimize

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_{t,i}(x) \quad \text{subject to } \; x \in \chi, \tag{7.1}$$

where $f_{t,i} : \mathbb{R}^d \to \mathbb{R}$ is a convex cost function associated with agent $i \in [n]$, assumed to be revealed to the agent only after the agent commits to the decision $x(t)$. In other words, the function $f_{t,i}$ is allowed to change over time in an unpredictable manner due to modeling errors and uncertainties in the environment. The optimization variable $x_i \in \mathbb{R}^d$ belongs to a closed convex set $\chi \subseteq \mathbb{R}^d$ and represents the local decision made by agent i. Furthermore, the online-DWDA scheme is analogous to the DWDA presented in Algorithm 3. The regret analysis is presented in the following result quantifying the performance of the proposed algorithm.

THEOREM 11. *Given the sequences $x_i(t)$ and $y_i(t)$ generated by lines* 5 *and* 8 *in* Algorithm 3, *for all $i \in [n]$ with $\psi(x^*) \leq R^2$ and $\alpha(t) = k/\sqrt{t}$, we have*

$$R_T(x^*, x_i) \leq \left(\frac{R^2}{k} + kL^2\left(\frac{6n}{1-\gamma} + 6n\delta\nu + 1\right)\right)\sqrt{T}, \tag{7.2}$$

where γ is a function of the ergodicity of the communication matrix (see (6.8)) while ν is a measure of network connectivity and is bounded by the diameter of the graph $\mathcal{G}^t$ (see also Proposition 16). In (7.2), $k > 0$ is an arbitrary constant and $\delta \geq 1$ is a positive

integer at which the union of directed topologies $\mathcal{G}^{\cup_{i=0}^{\delta-1}} = \bigcup_{i=0}^{\delta-1} \mathcal{G}^i$ *over some fixed uniform intervals* δ *is strongly connected.*

PROOF. Consider an arbitrary fixed decision $x^* \in \chi$ and a sequence $\phi(t)$ generated by (6.4). From the L-Lipschitz continuity of $f_{t,i}$'s and the definition of regret in (II.4), the regret is bounded as

$$R_T(x^*, x_i) \leq \sum_{t=1}^{T} \left(f_t(\phi(t)) - f_t(x^*) + L\|x_i(t) - \phi(t)\| \right). \tag{7.3}$$

Note that we can reformulate the first term on the right hand side of (7.3) as

$$f_t(\phi(t)) - f_t(x^*) = \left(\frac{1}{n} \sum_{i=1}^{n} f_{t,i}(x_i(t)) - f_t(x^*) \right) + \left(\frac{1}{n} \sum_{i=1}^{n} \left[f_{t,i}(\phi(t)) - f_{t,i}(x_i(t)) \right] \right). \tag{7.4}$$

Based on the convexity of $f_{t,i}$'s, we have

$$\sum_{t=1}^{T} \left(\frac{1}{n} \sum_{i=1}^{n} f_{t,i}(x_i(t)) - f_t(x^*) \right) \leq \sum_{t=1}^{T} \left(\frac{1}{n} \sum_{i=1}^{n} \langle g_i(t), x_i(t) - x^* \rangle \right), \tag{7.5}$$

where $g_i(t) \in \partial f_{t,i}(x_i(t))$ is the sub-gradient of $f_{t,i}$ at $x_i(t)$. Thereby, we can express the regret bound based on (7.4), (7.5), and the L-Lipschitz continuity of $f_{t,i}$'s as,

$$\begin{aligned} R_T(x^*, x_i) \leq \sum_{t=1}^{T} (& \frac{1}{n} \sum_{i=1}^{n} \langle g_i(t), x_i(t) - x^* \rangle \\ & + \frac{L}{n} \sum_{i=1}^{n} \|x_i(t) - \phi(t)\| + L\|x_i(t) - \phi(t)\|). \end{aligned} \tag{7.6}$$

The first term on the right had side of (7.6) can be expanded as

$$\begin{aligned} & \sum_{t=1}^{T} \left(\frac{1}{n} \sum_{i=1}^{n} \langle g_i(t), x_i(t) - x^* \rangle \right) \\ & = \sum_{t=1}^{T} (\frac{1}{n} \sum_{i=1}^{n} \langle g_i(t), x_i(t) - \phi(t) \rangle \\ & + \frac{1}{n} \sum_{i=1}^{n} \langle g_i(t), \phi(t) - x^* \rangle). \end{aligned} \tag{7.7}$$

Now, we need to bound the terms on the right hand side of (7.7). The first term is bounded based on the convexity and L-Lipschitz continuity of $f_{t,i}$.[2] In other words,

$$\langle g_i(t), x_i(t) - \phi(t)\rangle \leq L\|x_i(t) - \phi(t)\|. \tag{7.8}$$

Since $x_i(t)$ and $\phi(t)$ are the projections of $y_i(t)$ and $\bar{y}(t)$ respectively, the Lipschitz continuity of $\Pi^{\psi}_{\chi}(.,\alpha)$ presented in Lemma 13 of the Appendix imposes a bound on $\|x_i(t) - \phi(t)\|$ as

$$\|x_i(t) - \phi(t)\| \leq \alpha(t)\|\bar{y}(t) - y_i(t)\|_*, \tag{7.9}$$

where $||.||_*$ is the dual norm. Therefore, using the bound in Lemma 14 of the Appendix and noting that $\|g_i(t)\|_* \leq L$, we can write (7.7) as

$$\begin{aligned}
&\sum_{t=1}^{T}\left(\frac{1}{n}\sum_{i=1}^{n}\langle g_i(t), x_i(t) - x^*\rangle\right)\\
&\leq \frac{L}{n}\sum_{t=1}^{T}\sum_{i=1}^{n}\alpha(t)\|\bar{y}(t) - y_i(t)\|_*\\
&\quad + \frac{L^2}{2}\sum_{t=2}^{T}\alpha(t-1) + \frac{1}{\alpha(T)}\psi(x^*).
\end{aligned} \tag{7.10}$$

Thus, (7.6), (7.9), and (7.10) imply that

$$\begin{aligned}
R_T(x^*, x_i) &\leq \frac{L^2}{2}\sum_{t=2}^{T}\alpha(t-1) + \frac{1}{\alpha(T)}\psi(x^\star)\\
&\quad + L\sum_{t=1}^{T}\alpha(t)\left(\|\bar{y}(t) - y_i(t)\|_* + \frac{2}{n}\sum_{i=1}^{n}\|\bar{y}(t) - y_i(t)\|_*\right).
\end{aligned} \tag{7.11}$$

[2]Note that convexity of $f_{t,i}$ implies $\langle g_i(t), x - y\rangle \leq f_{t,i}(x) - f_{t,i}(y)$. Therefore, based on L-Lipschitz continuity of $f_{t,i}$'s, we have $||g_i||_* \leq L$ and we can deduce (7.8).

On the other hand, Lemma 15 of the Appendix imposes an upper bound on the last term on the right hand side of (7.11). Thus, using (6.12) the regret is further bounded as

$$R_T(x^*, x_i) \leq \frac{L^2}{2} \sum_{t=1}^{T-1} \alpha(t) + \frac{1}{\alpha(T)} \psi(x^*) + 3L^2 \left(\frac{n}{1-\gamma} + n\delta\nu + 2\,(1-n) \right) \sum_{t=1}^{T} \alpha(t). \tag{7.12}$$

The statement of the theorem now follows from the integral test on $\alpha(t) = k/\sqrt{t}$ and $\psi(x^*) \leq R^2$. □

Theorem 11 indicates the "good" performance of online-DWDA through sub-linear regret and highlights the importance of the underlying network topology through the parameters γ and ν examined in §6.

Next we present the regret analysis for the (temporal) running average estimates at each agent exhibiting a similar dependence on the network connectivity.

COROLLARY 12. *Given the sequence $\widetilde{x}_i(t)$ generated by line* 7 *in* Algorithm 3 *for all $i \in [n]$ with $\psi(x^*) \leq R^2$ and $\alpha(t) = k/\sqrt{t}$, we have*

$$R_T(x^*, \widetilde{x}_i) \leq 2 \left(\frac{R^2}{k} + kL^2 \left(\frac{6n}{1-\gamma} + 6n\delta\nu + 1 \right) \right) \sqrt{T}.$$

PROOF. Since the cost function $f_t(x(t))$ is convex, $f_t(\widetilde{x}_i(t)) \leq \frac{1}{t} \sum_{s=1}^{t} f_t(x_i(s))$. Therefore, we have

$$f_t(\widetilde{x}_i(t)) - f_t(x^*) \leq \frac{1}{t} R_t(x^*, x_i). \tag{7.13}$$

Thus, the running average regret is bounded as

$$R_T(x^*, \widetilde{x}_i) \leq \sum_{t=1}^{T} \left(\frac{1}{t} R_t(x^*, x_i) \right). \tag{7.14}$$

On the other hand, the regret bound (7.2) implies that

$$R_T(x^*, \widetilde{x}_i) \leq \left(\frac{R^2}{k} + kL^2 \left(\frac{6n}{1-\gamma} + 6n\delta\nu + 1 \right) \right) \times \left(\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \right). \tag{7.15}$$

The statement of the corollary now follows from the integral test on the right hand side of (7.15). □

8. Online Distributed Estimation

Adopting the least squares point of view, a model for online estimation over a distributed sensor network is presented in this section. The distributed sensor network aims to estimate a random vector $\theta \in \Theta = \left\{\theta \in \mathbb{R}^d | \, \|\theta\|_2 \leq \theta_{\max}\right\}$. Note that Θ is a closed convex set containing the origin. The observation vector $z_{t,i} : \mathbb{R}^d \to \mathbb{R}^{p_i}$ represents the ith sensor measurement at time t which is uncertain and time-varying due to the sensor's susceptibility to unknown environmental factors such as jamming. The sensor is assumed (not necessarily accurately) to have a linear model of the form $h_i(\theta) = H_i\theta$, where $H_i \in \mathbb{R}^{p_i \times d}$ is the observation matrix of sensor i and $\|H_i\|_1 \leq h_{\max}$ for all i. Consider now the interconnection topology between the sensors defined via the directed graph $\mathcal{G} = (V, E, W)$, where the set of n sensors are represented by V. The presence of an edge $(j, i) \in E$ indicates an information flow from sensor j to sensor i. The set of agents that are communicating with agent i is defined as the neighborhood set $N(i) = \{j \in V | (j, i) \in E\}$. Figure 8.1 graphically summarizes the problem setup. The objective is to find the argument $\hat{\theta}$ that minimizes the cost function

$$f_t(\hat{\theta}) = \frac{1}{n} \sum_{i=1}^{n} f_{t,i}(\hat{\theta}) \quad \text{subject to } \hat{\theta} \in \Theta, \tag{8.1}$$

where

$$f_{t,i}(\hat{\theta}) = \frac{1}{2} \left\| z_{i,t} - H_i\hat{\theta} \right\|_2^2 \tag{8.2}$$

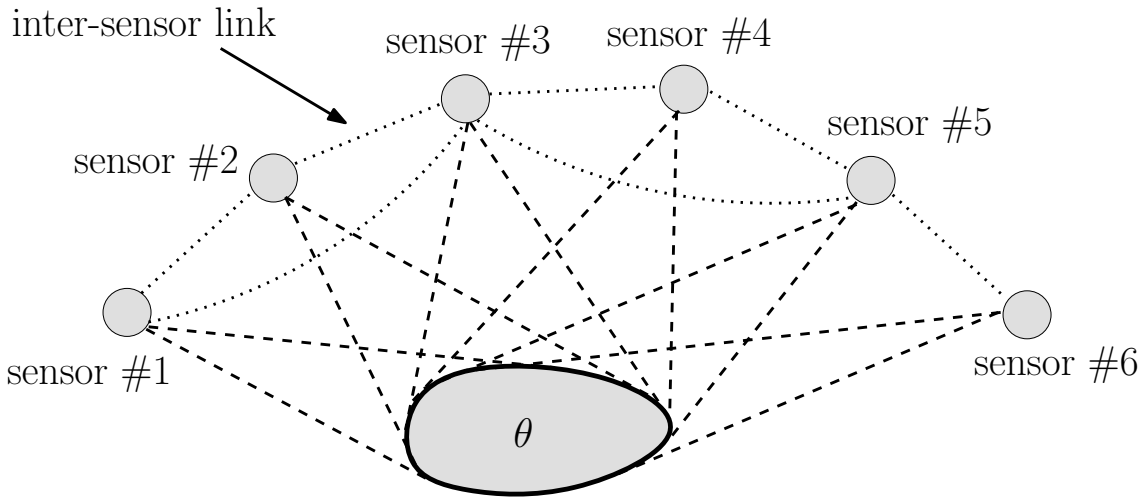


FIGURE 8.1. A graphical representation of a distributed sensor network.

is a convex cost function associated with sensor $i \in [n]$. It is assumed that the value of this local cost at time t is only revealed to the sensor after $\hat{\theta}(t)$ has been computed, that is, the local error functions are allowed to change over time in an unpredictable manner due to modeling errors and uncertainties in the environment. The (sub)gradient of the local estimation error (8.2),

$$\partial f_{t,i}(\hat{\theta}) = H_i^T \left(z_{t,i}(\theta) - H_i \hat{\theta} \right), \tag{8.3}$$

is also assumed to be known to the sensor and its neighbors. We note that the cumulative cost at time T is defined as $f(\hat{\theta}) = \sum_{t=1}^{T} f_t(\hat{\theta})$.

In an offline setting, for all $t \in [T]$, each sensor i has a noisy observation $z_{t,i} = H_i\theta + v_{t,i}$, where $v_{t,i}$ is generally assumed to be (independent) white noise. In this case, the centralized time-averaged optimal estimate for (8.1) is

$$\theta^* = \frac{1}{T} \sum_{t=1}^{T} \left(\sum_{i=1}^{n} H_i^T \Sigma_{t,i}^{-1} H_i \right)^{-1} \left(\sum_{i=1}^{n} H_i^T \Sigma_{t,i}^{-1} z_{i,t} \right), \tag{8.4}$$

where $\Sigma_{t,i}$ is the covariance of the error observed by sensor i at time t **[79]**. For the case where $\theta \in \mathbb{R}$, $\Sigma_{t,i} = I$, and $H_i = 1$, the optimal estimate is $\theta^* = \frac{1}{nT} \sum_{i=1}^{n} \sum_{t=1}^{T} z_{t,i}$. However, this approach to estimation problems is not suitable in scenarios where the noise characteristics are unknown. For example when a wireless sensor network is employed

in an unknown and dynamic environment, the measurement signal can be blocked or degraded due to obstructions such as walls, furniture, trees, or buildings. This is known as the shadowing effect and usually modeled as a function of the environment in which the network is deployed. Another example is jamming of one or more sensors in the network. When the sensor resolution and noise characteristics are not known ahead of time, the dynamic weight selection procedure discussed in §5 can be employed to eliminate the information from the jammed sensors.

An online framework is particularly suitable for such estimation problems without relying on prior assumption or knowledge of the statistical properties of the data. In the proposed distributed estimation algorithm, at time step t, each sensor i estimates $\hat{\theta}_i \in \Theta$ based on the local information available to it and then an "oracle" reveals the cost $f_t(\hat{\theta}_i)$.

The bounds presented in Theorem 11 apply after selecting $\psi(\hat{\theta}) = \frac{1}{2}\|\hat{\theta}\|_2^2$ and the parameter $\alpha(t)$ accordingly. In order to find the constants R and L featured in the result, we note that for $\hat{\theta} \in \Theta$, $\psi(\hat{\theta}) \leq \frac{1}{2}\theta_{\max}^2$, and thus $R \leq \frac{1}{\sqrt{2}}\theta_{\max}$. In this example, we assume that the observation for agent i at time t is of the form $z_{t,i} = a_t\theta + b_t$ for some $a \in (0, a_{\max})$ and $b \in (-b_{\max}, b_{\max})$. Therefore,

$$\sup_{\theta \in \chi} \|z_{t,i}(\theta)\|_2 \leq a_{\max}\theta_{\max} + b_{\max}.$$

Further, the function $f_{t,i}$ is Lipschitz as it is convex on a compact domain and the Lipschitz constant can be found by observing that

$$\begin{aligned}
&\left|f_{k,i}(\hat{\theta}) - f_{k,i}(\phi)\right| \\
&\leq \frac{1}{2}\left|\left(\hat{\theta} - \phi\right)^T H_i^T H_i \left(\hat{\theta} - \phi\right)\right| + \left|z^T H_i \left(\hat{\theta} - \phi\right)\right| \\
&\leq \left(\frac{1}{2}\|H_i\|_F^2 \left\|\hat{\theta} - \phi\right\|_2 + \|z_{t,i}\|_2 \|H_i\|_F\right) \left\|\hat{\theta} - \phi\right\|_2
\end{aligned}$$

and thus $L = \left(\frac{1}{2}\theta_{\max}h_{\max} + a_{\max}\theta_{\max} + b_{\max}\right)h_{\max}$. Hence $R_T(\theta^*, \hat{\theta}_i)/T \to 0$ and the algorithm performs as well as best fixed estimate θ^* in hindsight (8.4) "on average". For the case where $\theta_t = \theta_{t+1}$ for $t = 1, 2, \ldots, T$, θ^* is the optimal estimate.

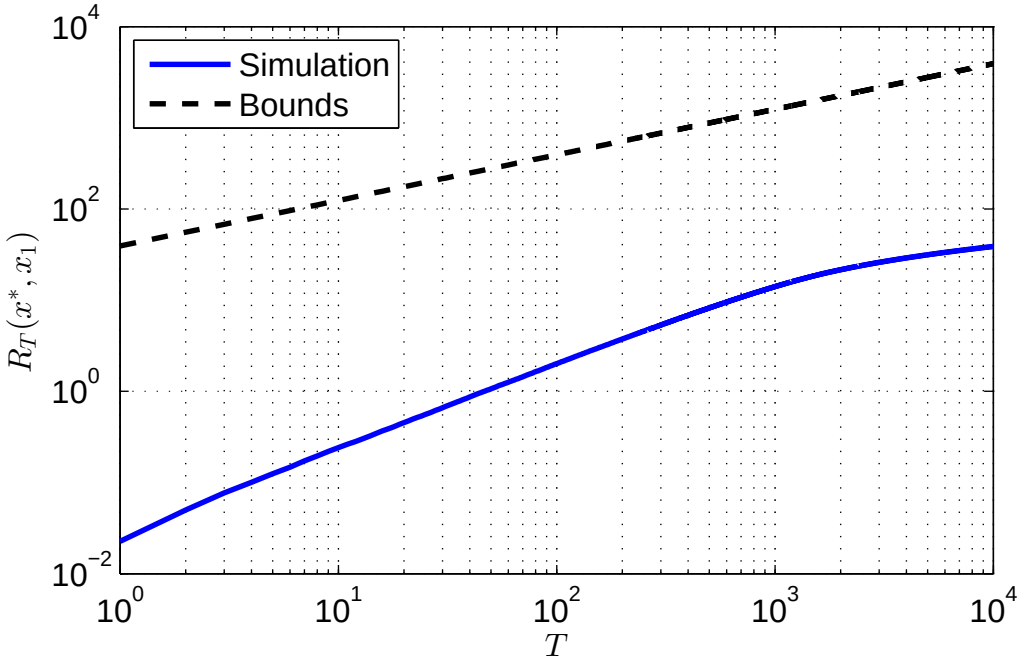


FIGURE 8.2. Accuracy of the bounds in (7.2) where $\mathcal{G}$ is a 100 node random directed graph with edge probability $p = 0.08$, $\nu = 5$, and $\gamma = 0.2034$.

The online-DWDA and DOA algorithms have been implemented on the described distributed sensor setup for $n = 100$ sensors. The objective is to estimate a scalar $\theta \in \left(-\frac{1}{2}, \frac{1}{2}\right)$ with a fixed $H_i \in \left(0, \frac{1}{4}\right)$ for each agent; hence $\sup_i |H_i| = \frac{1}{4}$. In this example, we have assumed $a \in (0, 1)$, $b \in \left(-\frac{1}{4}, \frac{1}{4}\right)$, $\beta = 0.9$, and $k = \frac{1}{4}$. Thus, $d = 1$, $\Theta = \left(-\frac{1}{2}, \frac{1}{2}\right)$, $h_{\max} = \frac{1}{4}$, $\theta_{\max} = \frac{1}{2}$, $R = \frac{1}{2\sqrt{2}}$, and $L = \frac{13}{64}$.

The online-DWDA and DOA algorithms were also applied to random sensor network with edge probability $p = 0.08$. Figure 8.2 shows a qualitative agreement of the theoretical regret bound (7.2) and simulation results, indicating that $R_T(x^*, x_1) = O(\sqrt{T})$. The improved performance of the adaptive network topology has been emphasized in Figure 8.3 in the context of a jamming scenario, where a number of sensors in the random regular network are assumed to have been jammed. This figure also demonstrates that the adaptive sensor network has a better regret performance as compared with the fixed topology sensor network.

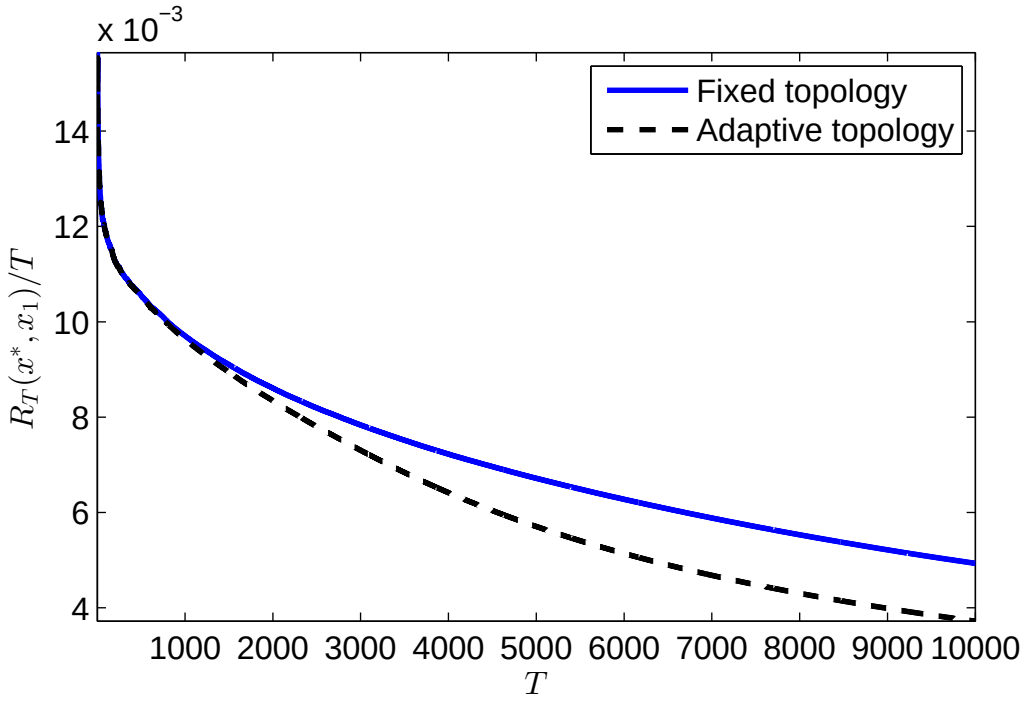


FIGURE 8.3. Regret performance over fixed and adaptive network topologies where $\mathcal{G}$ is a 100 node random 4-regular graph and 25 sensors are assumed to have been jammed. For the jammed sensors, $b_t = b_{\max}$ and $a_t = H_i$.

In addition, the performance of the proposed adaptive online distributed estimation in the presence of various noise types is presented in Figure 8.4. These simulation results indicate that $R_T(\theta^*, \hat{\theta}_1) = O(\sqrt{T})$ for all noise types considered without a prior assumption on the noise characteristics.

Furthermore, the role of network connectivity in the performance of the algorithm has been emphasized in Figure 8.5 for various classes of network topologies, directly correlated to the network connectivity measure γ. This result can be applied to designing sensor network topologies that operate in highly uncertain environments. Suitable metrics for such a topology design procedure include $\lambda_2(P(\mathcal{G}^0))$ that predictably scales with n, such as random regular graphs and expander graphs [**80**].

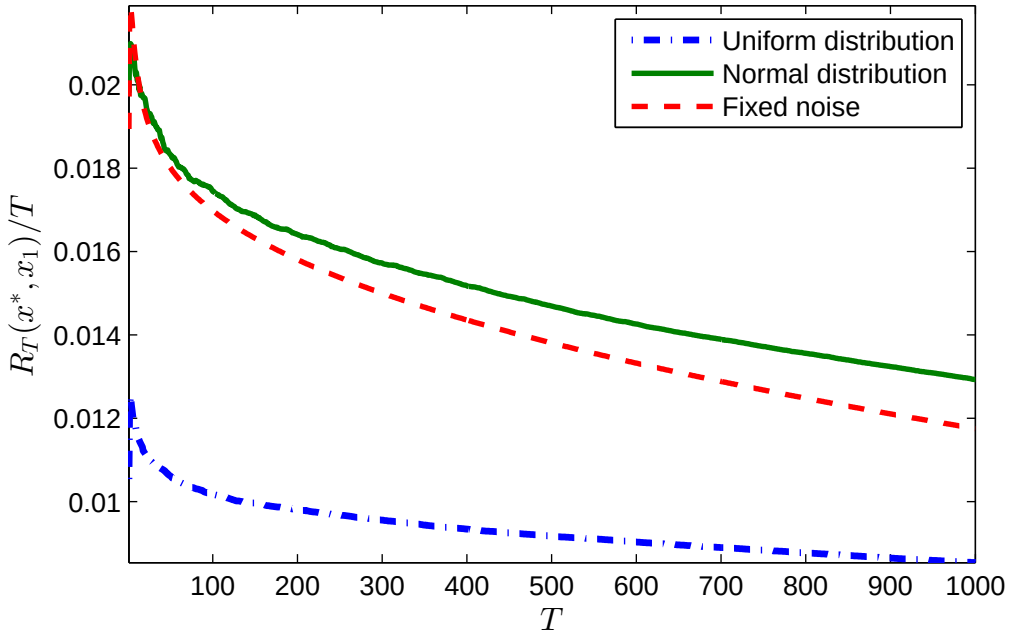


FIGURE 8.4. Regret performance for three different observation noise characteristics, where $\mathcal{G}$ is a 100 node random 4-regular graph. The noise signals have been generated from distributions with mean $-b_{\max}$ and standard deviation $b_{\max}$.

9. Remarks

This chapter studies the problem of decentralized optimization on dynamic networks operating in an uncertain environment. An algorithm has been presented that evolves distributively using only local information available to the agents in the network. Our analysis provided a convergence rate of $O(1/\sqrt{T})$ and a sub-linear regret of $O(\sqrt{T})$ in the online setting. In addition, the convergence analysis of the distributed optimization algorithm highlighted the role of two measures of network connectivity.

A distributed dynamic weight selection procedure has also been proposed that on average, performs as well as the best strategy for information diffusion in hindsight. It was demonstrated that this approach improves the convergence rate by mitigating the network effects.

This work can be applied in the context of a range of applications such as mobile sensor networks where the network is susceptible to unknown errors, jamming, link failure, and

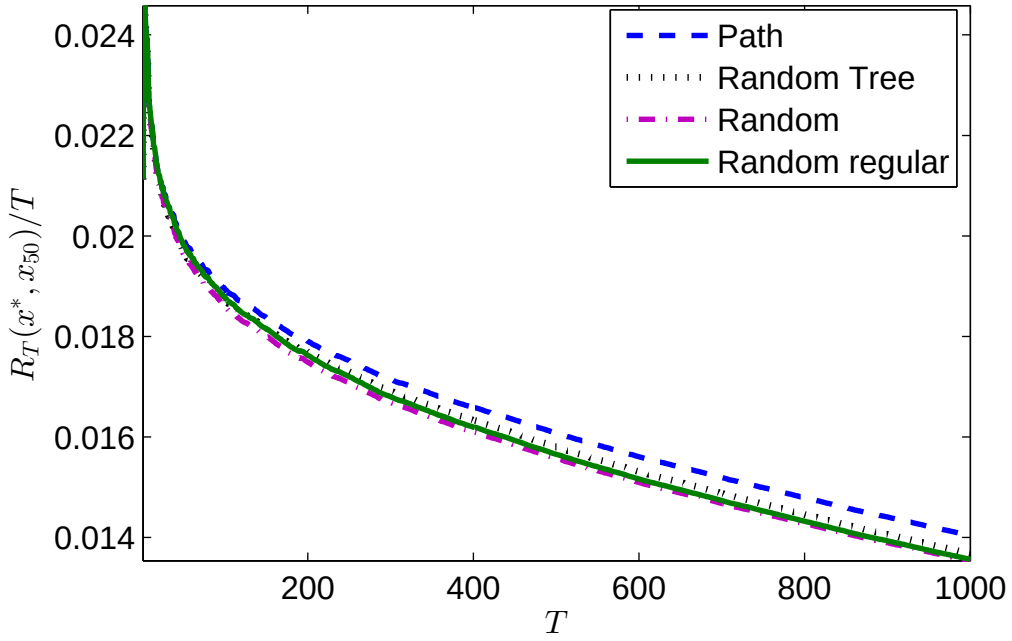


FIGURE 8.5. The performance of the online distributed estimation algorithm on four different 100 node graphs with $\gamma = \{0.8999, 0.8998, 0.7939, 0.4110\}$, for the path, random tree, random k-regular with $k = 4$, and random graph with edge probability $p = 0.08$, respectively, presented in increasing order of performance.

a varying network topology. Moreover, this work can be extended in several directions. One such extension, which is the subject of our future work, involves examining online distributed filtering. More generally, the online approach can be adopted for a host of network dynamic systems that operate in unstructured environments, requiring that a learning algorithm is embedded in the network-level decision-making process.

10. Appendix

We note that Lemmas 13 and 14 have been shown by Duchi *et al.*, [**59**] and are presented here for reference.

LEMMA 13. [**59**] *For any $u, v \in \mathbb{R}^m$, and under the conditions stated for proximal function ψ and step size $\alpha(t)$, we have $\|\Pi_\chi^\psi(u, \alpha) - \Pi_\chi^\psi(v, \alpha)\| \leq \alpha\|u - v\|_*$.*

LEMMA 14. **[59]** *For any positive and non-increasing sequence $\alpha(t)$ and $x^* \in \chi$,*

$$\sum_{t=1}^{T} \langle \overline{g}(t), \phi(t) - x^*(t) \rangle \leq \frac{1}{2} \sum_{t=1}^{T} \alpha(t-1) \|\overline{g}(t)\|_\star^2 + \frac{1}{\alpha(T)} \psi(x^*),$$

where the sequence $\phi(t)$ is generated by (6.4).

The following result presents a bound on $\|\bar{y}(t) - y_i(t)\|_*$ proportional to the error incurred by the decentralized update in Algorithm 3.

LEMMA 15. *For sequences $y_i(t)$ and $\bar{y}(t)$ generated by line 5 of Algorithm 3 and (6.3), respectively, we have,*

$$\|\bar{y}(t) - y_i(t)\|_* \leq L \sum_{k=0}^{t-2} \sum_{j=1}^{n} \left| P_{ij}^{(t-1,k+1)} - \pi_j \right| + 2L,$$

for all $i \in [n]$.

PROOF. Reformulating the update in line 5 of Algorithm 3 for all $i \in [n]$, by induction through s steps we have,

$$y_i(t) = \sum_{j=1}^{n} P_{ij}^{(t-1,t-s)} y_j(t-s) + \sum_{k=t-s}^{t-2} \sum_{j=1}^{n} P_{ij}^{(t-1,k+1)} g_j(k) + g_i(t-1). \tag{10.1}$$

Since $\bar{y}(t)$ evolves as in (6.3), by setting $s = t$ in (10.1) and assuming $y_i(0) = 0$, we get,

$$\bar{y}(t) - y_i(t) = \sum_{k=0}^{t-2} \left(\sum_{j=1}^{n} \left(\pi_j - P_{ij}^{(t-1,k+1)} \right) g_j(k) \right) + \bar{g}(t-1) - g_i(t-1). \tag{10.2}$$

Thus, the dual norm of (10.2) is bounded as

$$\|\bar{y}(t) - y_i(t)\|_* \leq \| \sum_{k=0}^{t-2} \left(\sum_{j=1}^{n} \left(\pi_j - P_{ij}^{(t-1,k+1)} \right) g_j(k) \right) \|_*$$

$$+ \|\bar{g}(t-1) - g_i(t-1)\|_*, \tag{10.3}$$

and the right hand side of (10.3) can be bounded by

$$\|\bar{y}(t) - y_i(t)\|_* \leq \sum_{k=0}^{t-2} \sum_{j=1}^{n} \left| P_{ij}^{(t-1,k+1)} - \pi_j \right| \|g_j(k)\|_*$$

$$+ \|\bar{g}(t-1) - g_i(t-1)\|_*. \tag{10.4}$$

Since $\|g_i(t)\|_\star \leq L$,

$$\|\bar{y}(t) - y_i(t)\|_* \leq L \sum_{k=0}^{t-2} \sum_{j=1}^{n} \left| P_{ij}^{(t-1,k+1)} - \pi_j \right| + 2L. \tag{10.5}$$

□

The following proposition provides an upper bound on ν in the convergence rate (6.10) over fixed topology networks.

PROPOSITION 16. *Consider a set $\mathcal{P}$ of stochastic matrices with positive diagonal elements, representing an arbitrarily strongly connected topologies over n nodes, i.e., $P^k \in \mathcal{P}$ for all positive integers k. Suppose that any two matrices P^{k_1} and P^{k_2} are of the same type.*[3] *Then, there exists an integer ν,*

$$1 \leq \nu \leq \min_{i \in [n]} \max_{j \in [n]} dist(j,i), \tag{10.6}$$

for which if the sequence $Q = P^{(m+\nu-2,m)}$ of matrices in $\mathcal{P}$ is not scrambling, $P^{m+\nu-1}Q$ is scrambling.

[3]The matrices A and B are of the same type if they have zero elements and positives elements in the same place.

PROOF. Let $Q_1 = P^m$ and $Q_2 = P^{m+1}P^m$. Thus every entry of Q_2 is represented as

$$[Q_2]_{ij} = \sum_{k=1}^{n} \left[P^{m+1}\right]_{ik} [P^m]_{kj}\,.$$

Since $[P^m]_{ii} > 0$ for all $i \in [n]$ and integer $m \geq 1$, the entry $[Q_2]_{ij}$ is positive if $(j,i) \in E$, $(i,j) \in E$, or if there exists a node $k \in [n]$ in the directed path from node j to node i with $\text{dist}(j,i) = 2$. Thus, the corresponding zero entry of Q_1 that has one of the aforementioned properties will be positive in Q_2. By induction, it follows that the entry of $[Q_\nu]_{ij}$ will be positive if $(j,i) \in E$, $(i,j) \in E$, or if there exists a node $k \in [n]$ in the directed path from node j to node i with $\text{dist}(j,i) = \nu_i$. Therefore, for each row i of Q_ν, all entries will be positive when

$$\nu_i = \max_{j \in [n]} \text{dist}(j,i).$$

Note that every element of any row of the sequence $Q = P^{(m+\nu-2,m)}$ of matrices in $\mathcal{P}$ is positive, the matrix Q is scrambling and ν satisfies the bound (10.6). □

A similar observation for the adjacency matrix of $\mathcal{G}$ can be found in the algebraic graph theory literature such as **[81]**.

CHAPTER 5

Online Distributed Optimization with Local Constraints

1. Summary

This chapter examines online distributed Alternating Direction Method of Multipliers (ADMM). The goal is to distributively optimize a global objective function over a network of decision makers under linear constraints. The global objective function is composed of convex cost functions associated with each agent. The local cost functions, on the other hand, are assumed to have been decomposed into two distinct convex functions, one of which is revealed to the decision makers over time and one known a priori. In addition, the agents must achieve consensus on the global variable that relates to the private local variables via linear constraints. In this work, we extend online ADMM to a distributed setting based on dual-averaging and distributed gradient descent. We then propose a performance metric for such online distributed algorithms and explore the performance of the sequence of decisions generated by the algorithm as compared with the best fixed decision in hindsight. This performance metric is called the social regret. A sub-linear upper bound on the social regret of the proposed algorithm is then obtained that underscores the role of the underlying network topology and certain condition measures associated with the linear constraints. The online distributed ADMM algorithm is then applied to a formation acquisition problem demonstrating the application of the proposed setup in distributed robotics.

2. Introduction

Distributed convex optimization over networks arises in diverse application domains, including multi-agent coordination, distributed estimation in sensor networks, decentralized tracking, and event localization [**83, 84**]. A subclass of these problems can be posed

as optimization problems consisting of a composite convex objective function subject to local linear constraints. This chapter examines two extensions of the well known Alternating Direction Method of Multipliers (ADMM) algorithm [**85**] for solving this class of problems. The first extension involves proposing two effective means for distributed implementation of the ADMM algorithm. The second extension pertains to addressing the situation where part of the cost function has an online feature, representing uncertainties in the cost incurred by each decision-maker prior to committing to a decision. ADMM is an appealing approach that blends the benefits of augmented Lagrangian and dual decomposition methods to solve the optimization problem of the form,

$$\min_{x\in\chi, y\in Y} f(x)+\phi(y),\ \text{s.t. } Ax+By=c, \tag{2.1}$$

where $f:\mathbb{R}^{d_x}\to\mathbb{R}$ and $\phi:\mathbb{R}^{d_y}\to\mathbb{R}$ are convex functions, and $\chi\subseteq\mathbb{R}^{d_x}$ and $Y\subseteq\mathbb{R}^{d_y}$ are convex sets; d_x and d_y represent, respectively, the dimensions of the underlying Euclidean spaces for the variables x and y. to the scenarios where the cost function is not known *a priori*. In other words, when the relevant decisions are made, one part of the cost function might be varying with time, or poorly characterized by a probability distribution, for example due to uncertainties in the environment. In this case, the time varying nature of this cost function is often signified by the notation f_t. Such problem formulations fall under the class of online optimization problems [**86**]. Stochastic and online ADMM (O-ADMM) have consequently been proposed to address this scenario in the context of the following optimization problem at time $T>0$:

$$\min_{x\in\chi, y\in Y} \sum_{t=1}^{T}(f_t(x)+\phi(y)),\ \text{s.t. } Ax+By=c. \tag{2.2}$$

In this direction, stochastic ADMM has been introduced by Ouyang *et al.* [**87**], where an identical and independent distribution for the uncertainties in the functions f_t have been considered and a convergence rate of $O(1/\sqrt{T})$ for convex functions has been shown. The O-ADMM algorithms proposed in [**88, 89**] also provide similar convergence rates without any assumption on the distribution of uncertainties.

On the other hand, ADMM has been considered in the setting of distributed convex optimization, particularly in the context of the consensus problem [**90**], where agreement is required on each agent's local variable y_i. In this case, the problem considered is of the form,

$$\min_{x\in\chi, y_1,\ldots,y_n\in Y} \sum_{i=1}^{n} \phi_i(y_i), \text{ s.t. } x = y_i, \text{ for } i = 1, 2, \ldots, n. \tag{2.3}$$

In this consensus ADMM problem formulation, the local variables y_i's are required to reach consensus through the global variable x; thus the linear constraint that ties the local variables to the global variable is an equality condition. The consensus constraint set can also be enforced through a network, where each agent coordinates on satisfying the equality constraint with its neighboring agents. An important distinction between the consensus ADMM and the problem of interest in our work is that the objective functional in the former problem setup does not explicitly have a term dictated by the global variable. a natural extension of (2.3) for the solution of distributed ADMM considered in this chapter (by replacing the global variable by its local copies and enforcing consensus) does not naturally lead to a distributed solution strategy without resorting to a sequential update [**91**] or inclusion of a fusion center [**90**]. works in distributed consensus ADMM include those based on stochastic asynchronous edge based ADMM [**92, 93**] and distributed gradient descent [**59, 94, 95, 96**], where under the online objective (2.2) and the time-invariant objective (2.3), the rate of convergence of $O(1/\sqrt{T})$ and $O(1/T)$ can be achieved, respectively. From an algorithmic perspective, the approach proposed in this work is also distinct from the stochastic asynchronous edge based ADMM proposed in [**92, 93**]. In particular, the embedding of dual averaging in the distributed algorithm offers a privacy preserving feature for the agents in the network. is, in the approach proposed in the present work, the local variables remain private for each agent and only the dual variables are communicated throughout the network. In applications such as cloud computing, the privacy preserving feature of the proposed algorithm might be of great interest for the security and reliability of the overall system. In addition, there is no constraint on the active nodes and each node can communicate with all of its neighbors at each time step.

Distributed ADMM has also been adopted for implementation on sensor networks [**99, 100**]. For example, Schizas *et al.* have proposed an algorithm that combines ADMM and block coordinate descent that guarantees the sensors collectively converge to the maximum likelihood estimate. The approach adopted by Schizas *et al.* is similar to one examined in [**90**], and as such, requires an averaging step for each subset of nodes at each iteration and exchanging the primal variables among the sensors. In [**100**], the stability of several algorithms for solving the stochastic consensus problem involving an aggregate mean-square-error cost function have been studied. The stability range is found to be a function of the number of nodes in the network as well as the regularization parameters and the network topology which indicates the poor performance of adaptive primal-dual strategies. However, this chapter does not discuss the speed of convergence and implementation of adaptive parameters which is the by-product of our analysis.

In addition, ADMM has been examined in the context of optimization over certain types of graphs. For example, Mota *et al.* [**101**] have studied the ADMM consensus problem for connected bipartite graphs. In particular, in [**101**] it is shown that distributed ADMM algorithm requires less communication between agents compared with other algorithms for a given accuracy of the solution. Other works in this area include that of Deng *et al.* [**102**] which has proposed a proximal Jacobian ADMM suitable for parallel computation. However, this method requires an all-to-all communication over a complete graph in each iteration.

The main contribution of this work is twofold. First, we show that both dual averaging and distributed gradient descent can seamlessly be integrated in the ADMM setup, providing effective means for its distributed implementation, or when the local variables are naturally associated with decision-makers operating over a network. Second, we show how network-level regret for such distributed ADMM can be derived, highlighting the effect of the underlying network structure on the performance of the algorithm and certain condition measures for the linear constraints, when part of the cost structure has an online character and is only revealed to the decision-makers over time. As such, the chapter extends and unifies some of the aforementioned results on online and distributed ADMM.

In the meantime, the chapter does not claim novelty in relation to developing a new class of ADMM algorithms and instead builds on, and extends the existing ADMM iterations for the purpose of its discussion. The chapter considers the extension of the optimization problem (2.2) of the form,

$$\min_{x \in \mathcal{X},\, y_1, \ldots y_n \in Y} \quad \frac{1}{n} \sum_{t=1}^{T} \left[\sum_{i=1}^{n} \{ f_{i,t}(x) + \phi_i(y_i) \} \right], \tag{2.4}$$

$$\text{s.t. } A_i x + B_i y_i = c_i \text{ for } i = 1, 2, \ldots, n,$$

involving a network of n agents, each cooperatively solving for the global optimal variable x and the respective local variables $y_1, \ldots y_n$. Here, the functions that compose problem (2.4) are decoupled, specifically only agent i has access to functions $f_{i,t}$, ϕ_i, and its privately known local linear constraint. of this problem include balancing sensing and communication in sensor networks, analyzing large data sets in cloud computing, and cooperative mission planning for a group of autonomous vehicles. The formulation of cooperative forest firefighting using the optimization model (2.4) and online distributed ADMM for its solution are discussed in §5.

outline of the chapter is as follows. The optimization problem formulation and the network-level measure of performance are introduced in §3 followed by the description of the OD-ADMM algorithm and the corresponding regret analysis in §4. Then in §5, the distributed formation acquisition problem is solved based on the proposed algorithm, and simulation results are presented to support the analysis. Finally, concluding remarks are provided in §6.

3. Problem Statement

In this section, we consider a large scale network of agents cooperatively optimizing a global objective function. Let the communication geometry amongst the n decision-makers or agents be denoted by the graph $\mathcal{G} = (V, E)$. Each node $i \in V$ is an agent that communicates with its neighbor $j \in N(i)$ through the edge $(i, j) \in E$. An equivalent online

distributed convex optimization problem to (2.4) is as follows,

$$\min_{x \in \chi, y_1, \ldots, y_n \in Y} \sum_{t=1}^{T} F_t(x,y) := \sum_{t=1}^{T} \{ f_t(x) + \frac{1}{n} \sum_{i=1}^{n} \phi_i(y_i) \} \tag{3.1}$$

subject to

$$r_i(x, y_i) := A_i x + B_i y_i - c_i = 0 \text{ for all } i \in [n], \tag{3.2}$$

where $f_t(x) = \frac{1}{n} \sum_{i=1}^{n} f_{i,t}(x)$, and $f_{i,t} : \mathbb{R}^{d_x} \to \mathbb{R}$ and $\phi_i : \mathbb{R}^{d_y} \to \mathbb{R}$ are convex for each i. The matrices in the local linear constraints are denoted as $A_i \in \mathbb{R}^{m_i \times d_x}$, $B_i \in \mathbb{R}^{m_i \times d_y}$, and $c_i \in \mathbb{R}^{m_i}$ at node $i \in [n]$. We assume that B_i^T is left invertible, i.e., $\sigma_{m_i}(B_i B_i^T)$ is non-zero, for all $i \in [n]$. The functions $f_{i,t}$ and ϕ_i are further assumed to be Lipschitz continuous with Lipschitz constants L_f and L_ϕ, respectively, that is,

$$|f_{i,t}(u) - f_{i,t}(v)| \leq L_f \|u - v\| \quad \text{for all } u, v \in \chi,$$
$$|\phi_i(u') - \phi_i(v')| \leq L_\phi \|u' - v'\| \quad \text{for all } u', v' \in Y.$$

The distributed nature of the optimization is illustrated in Figure 3.1.

Throughout this chapter, we suppose the following Slater condition holds:

Assumption 1 (Slater condition): There exist a point $(x^*, y_1^*, \cdots, y_n^*)$ in relative interior of $\chi \times Y^n$ such that $r_i(x, y_i) = 0$ for all $i \in [n]$.

This assumption is naturally used in the analysis of the duality gap for deriving bounds on the social regret. Moreover, we assume that

Assumption 2 (optimal solution): the set of optimal solutions of (3.1) is nonempty and the finite optimum value is $\mathcal{P}^*$.

The diameter of the set χ, defined as $\mathbf{diam}(\chi) = \sup_{x,x' \in \chi} ||x - x'||$, is assumed to be finite and denoted by D_χ.

The *local* decisions made by agent i is represented by the optimization variables $x_i \in \mathcal{X}$ and $y_i \in Y$; note that we allow the agents to have a local (not necessary exact) version of

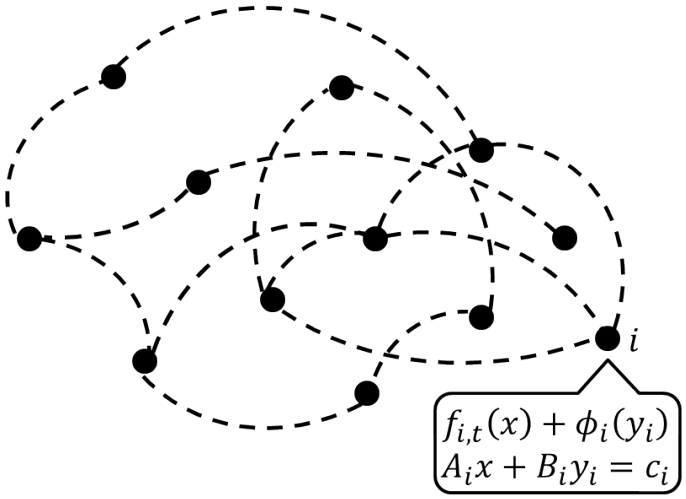


FIGURE 3.1. Distributed ADMM problem over a network; each agent operates based of the local objective $f_{i,t}(x)+\phi_i(y_i)$ and the local linear constraint $A_i x + B_i y_i = c_i$.

the global variable x, namely x_i. In addition, we assume that subgradients $\partial f_{i,t}(x)$ can be computed for every $x \in \chi$. In the online setting, based on the available local information, each decision maker i selects a global variable $x_{i,t} \in \chi$ and local variable $y_{i,t} \in Y$, at time t. The cost $f_{i,t}(x_{i,t})$ is then revealed to this agent after its local decision $x_{i,t}$ has been committed to at time t.

3.1. Regret for Constrained Optimization. We now examine a measure for evaluating the performance of OD-ADMM based on variational inequalities. is inspired by the convergence analysis of Douglas-Rachford ADMM presented in [**103**].

Consider the Lagrangian for the constrained optimization problem (3.1) as

$$\mathcal{L}_T(x, y, \lambda) = \sum_{t=1}^{T} \left\{ f_t(x) + \frac{1}{n} \sum_{i=1}^{n} (\phi_i(y_i) + \langle \lambda_i, r_i(x, y_i) \rangle) \right\}, \tag{3.3}$$

where $x \in \mathcal{X}$ and $y_i \in Y$, as well as assuming $\lambda_i \in \mathbb{R}^{m_i}$, for all $i \in [n]$. Then, the Lagrange dual function is defined as

$$\mathcal{D}(\lambda) = \inf_{x \in \chi, y_i \in Y} \mathcal{L}_T(x, y, \lambda), \tag{3.4}$$

implying that $\mathcal{D}(\lambda)$ is concave and yields a lower bound on the optimal value of (3.1) **[90]**. The penalty dual problem is given by

$$\max_{\lambda \in \mathcal{Z}^n} \mathcal{D}(\lambda) = \max_{\lambda \in \mathcal{Z}^n} \mathcal{L}_T(x^*, y^*, \lambda), \tag{3.5}$$

where $\mathcal{Z}^n = \mathbb{R}^{m_i \times n}$ and the penalty dual optimal value is denoted by $\mathcal{D}^*$. Slater condition guarantees zero duality gap and the existence of a dual optimal solution $\lambda^* \in \mathcal{Z}^n$ **[104]**. Therefore, when $(x^*, y_1^*, \cdots, y_n^*) \in \chi \times Y^n$ solves the primal problem (3.1)-(3.2), λ^* solves the penalty dual problem (3.5), and

$$\sup_{\lambda \in \mathcal{Z}^n} \inf_{x \in \chi} \mathcal{L}_T(x, y, \lambda) = \inf_{x \in \chi} \sup_{\lambda \in \mathcal{Z}^n} \mathcal{L}_T(x, y, \lambda),$$

the primal and dual optimal vectors form a saddle-point for the Lagrangian $\mathcal{L}_T$ **[106]**. Thus, based on the saddle point definition **[105]**, for all $w = (x, y_1, \cdots, y_n, \lambda_1, \cdots, \lambda_n) \in \Omega$, where $\Omega = \chi \times Y^n \times \mathcal{Z}^{n*}$, we have

$$\mathcal{L}_T(x^*, y^*, \lambda) \leq \mathcal{L}_T(x^*, y^*, \lambda^*) \leq \mathcal{L}_T(x, y, \lambda^*). \tag{3.6}$$

Moreover, from the KKT conditions we can express the dual optimal set as

$$\mathcal{Z}^{n*} = \left\{ \lambda^* | \sum_{t=1}^{T} \left(\nabla_x f_t(x^*) + \frac{1}{n} \sum_{i=1}^{n} A_i^T \lambda_i^* \right) = 0, \right.$$
$$\left. \sum_{t=1}^{T} \left(\nabla_y \phi_i(y^*) + \frac{1}{n} \sum_{i=1}^{n} B_i^T \lambda_i^* \right) = 0 \right\}.$$

Since there exist a solution to the penalty dual problem, the dual optimal set is non-empty. Moreover, based on Lipschitz continuity of cost functions, $\nabla_x f_t(x^*)$ and $\nabla_y \phi_i(y^*)$ are finite and bounded; hence λ^* can be picked from $\mathcal{Z}^{n*}$ such that

$$D_\lambda = \min_{\lambda \in \mathcal{Z}^{n*}} \max_{i \in [n]} ||\lambda_i||, \text{ and } D_\lambda \text{ is finite.}$$

Based on Assumption 1 (Slater condition) and 2 (optimal solution), there always exist a non-empty set of saddle points of $\mathcal{L}_T$ and throughout this chapter, we assume that:

Assumption 3 (saddle point): vector $w^* \in \Omega$ denotes a saddle point for (3.3).

Given w^* is a saddle point of $\mathcal{L}_T$, a consequence of inequality (3.6) is that $\widetilde{w} = \left(\widetilde{x}, \widetilde{y}, \widetilde{\lambda}\right) \in \Omega$ approximately solves the primal problem with accuracy $\epsilon_T^{\mathcal{P}} \geq 0$ if it satisfies,

$$0 \leq \mathcal{L}_T(\widetilde{x}, \widetilde{y}, \lambda^*) - \mathcal{L}_T(x^*, y^*, \lambda^*) \leq \epsilon_T^{\mathcal{P}},$$

that is,

$$0 \leq \mathcal{L}_T(\widetilde{x}, \widetilde{y}, \lambda^*) - \mathcal{P}^* \leq \epsilon_T^{\mathcal{P}}. \tag{3.7}$$

Based on (3.4), the inequality (3.7) can also be referred as dual feasibility. In addition, $\widetilde{w} = \left(\widetilde{x}, \widetilde{y}, \widetilde{\lambda}\right) \in \Omega$ approximately solves the dual problem with accuracy $\epsilon_T^{\mathcal{D}} \geq 0$ if

$$0 \leq \mathcal{L}_T(x^*, y^*, \lambda^*) - \mathcal{L}_T(x^*, y^*, \widetilde{\lambda}) \leq \epsilon_T^{\mathcal{D}},$$

that is,

$$0 \leq \mathcal{D}^* - \mathcal{D}(\widetilde{\lambda}) \leq \epsilon_T^{\mathcal{D}}, \tag{3.8}$$

which represents the dual sub-optimality. The conditions in (3.7)-(3.8) can be combined to represent the duality gap as

$$\sum_{t=1}^{T} f_t^{\Delta}(\widetilde{w}, w^*) + \frac{1}{n}(\sum_{i=1}^{n} \phi_i^{\Delta}(\widetilde{w}, w^*) + H_i^{\Delta}(\widetilde{w}, w^*)) \leq \epsilon_T.$$

where

$$\begin{aligned}
f_t^{\Delta}(w, w^*) &= f_t(x) - f_t(x^*) \\
\phi_i^{\Delta}(w, w^*) &= \phi_i(y_i) - \phi_i(y_i^*) \\
H_i^{\Delta}(w, w^*) &= h_{1i}^{\Delta}(w, w^*) + h_{2i}^{\Delta}(w, w^*) \\
h_{1i}^{\Delta}(w, w^*) &= \left\langle x - x^*, A_i^T \lambda_i^* \right\rangle + \langle \lambda_i - \lambda_i^*, -r_i(x^*, y_i^*) \rangle \\
h_{2i}^{\Delta}(w, w^*) &= \left\langle y_i - y_i^*, B_i^T \lambda_i^* \right\rangle,
\end{aligned}$$

and $\epsilon_T = \epsilon_T^{\mathcal{P}} + \epsilon_T^{\mathcal{D}} \geq 0$.

Analogous to the regret definition for O-ADMM algorithm **[108]**, we can consider a sequence of decisions w_t, where $w_t \in \Omega$ for each t, instead of a fixed decision $\widetilde{w}$. Consequently, the sequence w_t approximately solves (3.1) and (3.2) with accuracy ϵ_T if

$$\sum_{t=1}^{T} f_t^{\Delta}(w_t, w^*) + \frac{1}{n}\left\{\sum_{i=1}^{n} \phi_i^{\Delta}(w_t, w^*) + H_i^{\Delta}(w_t, w^*)\right\} \leq \epsilon_T, \tag{3.9}$$

for the optimal solution $w^* \in \Omega$, referred to as *fixed case solutions* to distinguish them from the time-varying online solution sequence w_t. Moreover, the mapping $H_i^{\Delta}(w, w^*)$ can be expressed as

$$H_i^{\Delta}(w, w^*) = \langle w_i(x) - w_i^*(x), H_i(w^*)\rangle\,,$$

where $w_i(x) = \begin{bmatrix} x & y_i & \lambda_i \end{bmatrix}^T$, $w_i^*(x) = \begin{bmatrix} x & y_i^* & \lambda_i^* \end{bmatrix}^T$, and

$$H_i(w) = \begin{bmatrix} 0 & 0 & A_i^T \\ 0 & 0 & B_i^T \\ -A_i & -B_i & 0 \end{bmatrix} w_i(x) + \begin{bmatrix} 0 \\ 0 \\ c_i \end{bmatrix}.$$

Since, the mapping $H_i(w)$ is affine in $w_i(x)$ and is defined through a skew symmetric matrix, it is monotone, and consequently **[109]**

$$\langle w_i(x) - w_i^*(x), H_i(w) - H_i(w^*)\rangle \geq 0$$

$$\langle w_i(x) - w_i^*(x), H_i(w)\rangle \geq \langle w_i(x) - w_i^*(x), H_i(w^*)\rangle\,. \tag{3.10}$$

Therefore, the inequality

$$\begin{aligned} \sum_{t=1}^{T} f_t^{\Delta}(w_t, w^*) + \frac{1}{n}\sum_{i=1}^{n} \big(&\phi_i^{\Delta}(w_t, w^*) \\ &+\langle w_{i,t}(x) - w_i^*(x), H_i(w_t)\rangle\big) \quad \leq \epsilon_T \end{aligned} \tag{3.11}$$

is a sufficient condition for (3.9).

Finally, motivated by the inclusion of regularization terms in the augmented Lagrangian method **[105]**, the term on the left hand side of (3.11) is supplemented with terms of the form $\frac{\rho}{2}||r_i(x_{i,t}, y_{i,t})||^2$, where $\rho > 0$, to promote agents satisfying the local

primal feasibility constraints. In our setting, the sequence $w_{i,t}$ is constructed from the distributed algorithm adopted by each agent i, specifically $w_t(x_{i,t}) = (x_{j,t}, y_t, \lambda_{t+1}) \in \Omega$ at time t, where $y_t = (y_{1,t}, \cdots, y_{n,t})$ and $\lambda_{t+1} = (\lambda_{1,t+1}, \cdots, \lambda_{n,t+1})$. The social regret is thus defined as,[1]

$$R_T = \max_{j\in[n]} R_{j,T}(w^*),$$

where

$$\begin{aligned} R_{j,T}(w^*) &= \sum_{t=1}^{T} f_t^{\Delta}(w_t(x_{j,t}), w^*) \\ &+ \frac{1}{n}\sum_{i=1}^{n}(\phi_i^{\Delta}(w_t(x_{j,t}), w^*) \\ &+ \langle w_{i,t}(x_{j,t}) - w_i^*(x_{j,t}), H_i(w_t(x_{j,t}))\rangle \\ &+ \frac{\rho}{2}||r_i(x_{i,t}, y_{i,t})||^2). \end{aligned}$$

Based on (3.11) we say that the sequence $w_{i,t}$ approximately solves (3.1) and (3.2) with accuracy ϵ_T if it satisfies $R_T \leq \epsilon_T$. Therefore, if the social regret is sub-linear with time, the online algorithm performs as well as the best fixed case decision provided with the complete sequence of cost functions a priori. In addition, the sub-linearity of the social regret ensures that the local linear constraints will be satisfied asymptotically.

4. Online Distributed ADMM on Networks: Social Regret, Network Effect, and Condition Measures

The main contribution of this chapter is extending O-ADMM [**89**] via Nesterov's Dual Averaging (DA) algorithm [**110**] and distributed subgradients (descent) method discussed in [**94, 111, 112, 113**], to provide a distributed decision-making process for the optimization problem discussed in §3 with a sub-linear social regret; we refer to this procedure as online distributed ADMM (OD-ADMM). The main challenge for the seamless integration of ADMM with dual averaging and distributed gradient descent for OD-ADMM is deriving and utilizing bounds on the network effect and the sub-optimality of the local decisions on

[1]Note that this form of regret penalizes the deviation of each agent's local copy of the global variable from the best fixed global decision in hindsight.

the social regret. This objective is achieved by building on the existing results reported in **[59, 94, 95]** regarding the network contribution in distributed optimization, as well as extensions of results discussed in **[82, 88, 89]**. basic idea behind our convergence analysis is as follows.[2] First, in Lemmas 22 and 23, we derive the gap between the local decisions and the average decision over the network. Then, we provide the sub-optimality gap in Lemmas 24 and 25. Finally, building on previous results, bounds on the social regret are presented in Theorems 17 and 18.

The proposed algorithm updates the vector $(x_i, y_i, z_i, \lambda_i)$ for each agent $i \in [n]$ by alternately minimizing the Lagrangian and augmented Lagrangian. In addition, the Lagrangian is linearized based on network-level update, leading to a subgradient descent method followed by a projection step onto the constraint set χ. Specifically, in the DA method, we let

$$z_{t+1} = z_t + \tilde{g}_t,$$

where $\tilde{g}_t = \nabla \mathcal{L}_t(x_t)$, followed by

$$x_{t+1} = \prod_{\chi}^{\psi} (z_{t+1}, \alpha_t)\,; \tag{4.1}$$

in this case, the parameter α_t is a non-increasing sequence of positive functions and $\prod_{\chi}^{\psi}(\cdot)$ is the projection operator onto χ defined as

$$\prod_{\chi}^{\psi}(z_{t+1}, \alpha_t) \equiv \arg\min_{x \in \chi} \left\{ \langle z_{t+1}, x \rangle + \frac{1}{\alpha_t} \psi(x) \right\}; \tag{4.2}$$

the inclusion of the proximal function $\psi(x) : \chi \to \mathbb{R}$ in the DA method as a regularizer prevents oscillations in the projection step. Note that ψ can be picked arbitrary with accordance with the following definition:

Definition 1 (proximal function): proximal function $\psi(x)$ has the following properties: (i) $\psi(x)$ is continuously differentiable, (ii) $\psi(x)$ is strongly convex with respect to $\|.\|$, (iii) $\psi \geq 0$, and (iv) $\psi(0) = 0$.

[2]All referenced lemmas are discussed in the Appendix.

On the other hand, in the subgradient descent (GD) method, the aforementioned steps in DA are replaced by

$$h_{t+1} = x_t - \alpha_t \tilde{g}_t,$$

followed by

$$x_{t+1} = \prod_{\mathcal{X}} h_{t+1} \equiv \arg\min_{x \in \mathcal{X}} ||x - h_{t+1}||. \tag{4.3}$$

Finally, the proposed online algorithm minimizes the augmented Lagrangian over y as

$$y_{t+1} = \arg\min_{y \in Y} \left\{ \mathcal{L}_t(x_{t+1}, y, \lambda_{t+1}) + \frac{\rho}{2} ||r(x_{t+1}, y)||^2 \right\},$$

and update the dual variable λ as[3]

$$\lambda_{t+2} = \lambda_{t+1} + \rho(Ax_{t+1} + By_{t+1} - c).$$

The distributed algorithm can be considered as an approximate ADMM by an agent i via a convex combination of information provided by its neighbors $N(i)$. Specifically, the global update step (4.1) and (4.3) can be reformulated with a distributed method. The underlying communication network can be represented compactly as a doubly stochastic matrix $P \in \mathbb{R}^{n \times n}$ which preserves the zero structure of the Laplacian matrix $L(\mathcal{G})$. For agents to have access to information contained in the subgradients $\tilde{g}_{i,t} = \nabla \mathcal{L}_{i,t}(x_{i,t})$ there must be information flow amongst the agents; as such, in our subsequent analysis it will be assumed that the graph $\mathcal{G}$ is strongly connected. A method to construct a doubly stochastic matrix P of the required form from the Laplacian of the network is provided in Proposition 19.

The online distributed ADMM (OD-ADMM) is presented in Algorithm 5.

The function $H_{\alpha_t}(\lambda_{i,t+1}, g_{i,t})$, referred in line 8 of the algorithm, represents a distributed update on the primal variable. this chapter, we consider two alternatives for this update.

[3]Note that the index for the dual variable is one time step ahead of the primal variables.

Algorithm 5: Online Distributed ADMM (OD-ADMM)

Input: $\rho > 0$, $\{\alpha_t\}_{t=1}^{T}$
Initialize $z_{i,1} = \lambda_{i,1} = 0$ and $x_{i,1} = 0$, $y_{i,1} = 0$ for $\forall i = 1, ..., n$
for $t = 1$ **to** T **do**
  Adversary reveals $f_t(t) = \{f_{i,t}(t);$ for $\forall i = 1, ..., n\}$
  Compute subgradient $g_i(t) \in \partial f_{i,t}(x_{i,t})$
  foreach *Agent* i **do**
    $\lambda_{i,t+1} = \lambda_{i,t} + \rho(A_i x_{i,t} + B_i y_{i,t} - c_i)$
    $x_{i,t+1} = H_{\alpha_t}(\lambda_{i,t+1}, g_{i,t})$
    $r_i(x_{i,t+1}, y) = A_i x_{i,t+1} + B_i y - c_i$
    $y_{i,t+1} = \text{argmin}_{y \in Y}(\phi_i(y) + \lambda_{i,t+1}^T r_i(x_{i,t+1}, y) + \frac{\rho}{2}||r_i(x_{i,t+1}, y)||^2)$
  end
end

4.1. OD-ADMM via Distributed Dual Averaging. In this method, the dual sub-gradient at each node is updated as a convex combination of its neighbor's dual sub-gradients and itself, namely,

$$z_{i,t+1} = \sum_{j=1}^{n} P_{ji} z_{j,t} + g_{i,t} + A_i^T \lambda_{i,t+1}, \tag{4.4}$$

and

$$x_{i,t+1} = \prod_{\chi}^{\psi}(z_{i,t+1}, \alpha_t), \tag{4.5}$$

where the projection operator $\prod_{\chi}^{\psi}(\cdot)$ is defined in (4.2). Before presenting the convergence rate of the proposed OD-ADMM algorithm we provide a few preliminary remarks and definitions. Let us define the sequences of (network) average dual subgradients z_t's and average subgradients g_t's as

$$z_t = \frac{1}{n}\sum_{i=1}^{n} z_{i,t}, \quad g_t = \frac{1}{n}\sum_{i=1}^{n} g_{i,t}. \tag{4.6}$$

Thus, in the distributed DA method, the following update rule is introduced similar to the standard DA algorithm,

$$z_{t+1} = z_t + g_t + \frac{1}{n}\sum_{i=1}^{n} A_i^T \lambda_{i,t+1}, \tag{4.7}$$

where the primal update is

$$\theta_{t+1} = \Pi_{\chi}^{\psi}(z_{t+1}, \alpha_t). \tag{4.8}$$

Now, the regret analysis will be presented, where the intermediate results required for its proof are relegated to the Appendix, namely Lemmas 21, 22, and 24. In particular, we show that with a proper choice of learning rate, the network effect and the sub-optimality of average decisions are sub-linear over time. Subsequently, building on these results, a sub-linear regret bound for OD-ADMM using distributed DA method can be established as formalized by the following result.

THEOREM 17. *For every saddle point w^* satisfying $\psi(x^*) \leq \Psi^2$, the sequence $w_{i,t}$, generated by Algorithm 5 where line 8 applies distributed dual-averaging method with $\alpha(t) = k/\sqrt{t}$, we have*

$$R_T \leq J_1 + J_2 k\sqrt{T}, \tag{4.9}$$

where

$$J_1 = \frac{D_\lambda}{\rho n} \sum_{i=1}^{n} \frac{\zeta_i}{\sigma_1(A_i)},$$

$$J_2 = 2\mathcal{Q}(L_f + \zeta^{\max})(\frac{2}{n} \sum_{i} (D_\lambda \sigma_1(A_i) + 2\zeta_i)),$$

with

$$\overline{\zeta} = \frac{1}{n} \sum_{i=1}^{n} \zeta_i,\ \zeta^{\max} = \max_i \zeta_i,$$

$$\zeta_i = \sqrt{m_i} L_\phi \frac{\sigma_1(A_i)}{\sigma_{m_i}(B_i^T)},\ \textit{and}\ \mathcal{Q} = \frac{\sqrt{n}}{1 - \sigma_2(P)}.$$

PROOF. Based on the definition of f_t we have

$$f_t^{\Delta}(w_t(x_{j,t}), w^*) = \frac{1}{n} \sum_{i=1}^{n} f_{i,t}^{\Delta}(w_t(x_{j,t}), w^*),$$

where $w_t(x_{j,t}) = (x_{j,t}, y_t, \lambda_{t+1}) \in \Omega$ and thus

$$R_{j,T}(w^*) = \frac{1}{n}\sum_{i,t} f_{i,t}^{\Delta}(w_t(x_{j,t}), w^*) + \phi_i^{\Delta}(w_t(x_{j,t}), w^*) \\ + H_i^{\Delta}(w_t(x_{j,t}), w^*) + \frac{\rho}{2}||r_i(x_{i,t}, y_{i,t})||^2.$$

In the meantime as f_t is L-Lipschitz and convex, we have

$$f_t^{\Delta}(w_t(x_{j,t}), w^*) = f_t(x_{j,t}) - f_t(\theta_t) + f_t(\theta_t) - f_t(x^*) \\ \leq L_f \left\|x_{j,t} - \theta_t\right\| + \left\langle g_t, \theta_t - x^* \right\rangle. \tag{4.10}$$

The first term in (4.10) represents the network effect in the regret bound, i.e., the deviation of local primal variable at each node from the average primal variable. Lemma 22 in the Appendix on the other hand, provides a bound on the network effect using the DA method. Therefore, replacing line 8 of Algorithm 5 with the distributed DA method implies that

$$\|\theta_t - x_{j,t}\| \leq \alpha_{t-1} \frac{\sqrt{n}\,(L_f + \zeta^{\max})}{1 - \sigma_2(P)}. \tag{4.11}$$

Moreover, from the integral test with $\alpha_t = k/\sqrt{t}$ it follows that[4]

$$\sum_{t=1}^{T} \alpha_{t-1} \leq 2k\sqrt{T}. \tag{4.12}$$

Hence, from (4.11) and (4.12) it follows that

$$\sum_{t=1}^{T} \|x_{j,t} - \theta_t\| \leq 2k\sqrt{T}\mathcal{Q}(L_f + \zeta^{\max}). \tag{4.13}$$

The second term in (4.10) represents the sub-optimality of the procedure due to using the first order approximation of f_t via subgradients. Applying Lemma 24 (Appendix) with

[4]Note that $\frac{1}{\sqrt{t}}$ is a non increasing positive function and the integral test leads to $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq 2\sqrt{T} - 1$.

(4.7) and (4.8) implies that

$$\sum_{t=1}^{T} \langle g_t, \theta_t - x^* \rangle \leq \sum_{t=1}^{T} [\frac{\alpha_t}{2} \parallel g_{t+1} + \frac{1}{n} \sum_{i=1}^{n} A_i^T \lambda_{i,t+2} \parallel_*^2$$

$$+ \frac{1}{n} \sum_{i=1}^{n} \langle \lambda_{i,t+1}, A_i(x^* - \theta_t) \rangle] + \frac{1}{\alpha_T} \psi(x^*). \tag{4.14}$$

The first term on the right hand side of (4.14) represents the gradient of Lagrangian function $\mathcal{L}_T$, defined in (3.3), with respect to the global variable x and is bounded as[5]

$$\sum_{t=1}^{T} \frac{\alpha_t}{2} \parallel g_{t+1} + \frac{1}{n} \sum_{i=1}^{n} A_i^T \lambda_{i,t+2} \parallel_*^2$$

$$\leq (\max_t \|g_{t+1}\|_* + \frac{1}{n} \sum_{i=1}^{n} \sigma_1(A_i) \max_t \|\lambda_{i,t}\|_*)^2 \sum_{t=1}^{T} \frac{\alpha_t}{2}. \tag{4.15}$$

Note that through KKT conditions, the above inequality represents the suboptimality of choice θ_t.

We now proceed to bound the individual terms in (4.15). By optimality of line 10 in Algorithm 5 and applying line 7, we have

$$\nabla_y \phi_i(y_{i,t}) = -B_i^T (\lambda_{i,t} + \rho r_i(x_{i,t}, y_{i,t})) = -B_i^T \lambda_{i,t+1},$$

for all $i \in [n]$ and $t \in [T]$. Moreover, since $||\nabla_y \phi_i(y_{i,t})|| \leq L_\phi$, we have $||B_i^T \lambda_{i,t+1}|| \leq L_\phi$. Thus, $\lambda_{i,t}$ is bounded as

$$||\lambda_{i,t}|| \leq ||(B_i B_i^T)^{-1} B_i||_F \, ||B_i^T \lambda_i||$$

$$\leq L_\phi (\sum_{j=1}^{m_i} \frac{1}{\sigma_j^2(B_i^T)})^{1/2} \leq \frac{\sqrt{m_i} L_\phi}{\sigma_{m_i}(B_i^T)}, \tag{4.16}$$

which implies that $||A_i^T \lambda_{i,t}|| \leq \sqrt{m_i} L_\phi \sigma_1(A_i)/\sigma_{m_i}(B_i^T)$. Based on Lipschitz continuity of f_t, we have that $\|g_{t+1}\|_* \leq L_f$, and subsequently (4.15) is bounded as

$$\sum_{t=1}^{T} \frac{\alpha_t}{2} \parallel g_{t+1} + \frac{1}{n} \sum_{i=1}^{n} A_i^T \lambda_{i,t+2} \parallel_*^2 \leq (L_f + \overline{\zeta})^2 k \sqrt{T}. \tag{4.17}$$

[5]Note that $\|Qx\| \leq \sigma_1(Q)||x||$ for any matrix $Q \in \mathbb{R}^{m \times n}$ and vector $x \in \mathbb{R}^n$.

The second term in the inequality (4.14) represents the sub-optimality of centralized decision θ_t with respect to the linear constraints. In order to analyze this term, it is first expanded into two terms representing the sub-optimality of local decision $x_{i,t}$ and the network effect, respectively, i.e.,

$$\begin{aligned}
&\langle \lambda_{i,t+1}, A_i(x^* - \theta_t)\rangle \\
&= \langle \lambda_{i,t+1}, A_i(x^* - x_{j,t})\rangle + \langle \lambda_{i,t+1}, A_i(x_{j,t} - \theta_t)\rangle \\
&= \langle \lambda_{i,t+1}, A_i(x^* - x_{j,t})\rangle + \langle \lambda_i^* - \lambda_{i,t+1}, -r_i(x_{j,t}, y_{i,t})\rangle \\
&\quad + \langle \lambda_{i,t+1}, A_i(x_{j,t} - \theta_t)\rangle + \langle \lambda_i^* - \lambda_{i,t+1}, r_i(x_{j,t}, y_{i,t})\rangle \\
&= -h_{1i}^{\Delta}(w_t(x_{j,t}), w^*) + \langle \lambda_{i,t+1}, A_i(x_{j,t} - \theta_t)\rangle \\
&\quad + \langle \lambda_i^* - \lambda_{i,t+1}, r_i(x_{j,t}, y_{i,t})\rangle \,. \qquad (4.18)
\end{aligned}$$

Based on the network effect introduced in (4.11), we have

$$\langle \lambda_{i,t+1}, A_i(x_{j,t} - \theta_t)\rangle \leq \sigma_1(A_i)||\lambda_{i,t+1}||\, \|x_{j,t} - \theta_t\|. \qquad (4.19)$$

Moreover, applying (4.13) and (4.16) to (4.19), it follows that

$$\sum_{t=1}^{T} \langle \lambda_{i,t+1}, A_i(x_{j,t} - \theta_t)\rangle \leq 2k\sqrt{T}\zeta_i \mathcal{Q}(L_f + \zeta^{\max}). \qquad (4.20)$$

The final term in (4.18) represents the first order necessary condition for optimality of the dual problem at $\lambda_{i,t+1}$. By applying line 7 of the algorithm and an inner product equality,

we obtain[6]

$$\begin{aligned}
&\langle \lambda_i^* - \lambda_{i,t+1}, r_i(x_{j,t}, y_{i,t})\rangle \\
&= \frac{1}{\rho}\langle \lambda_i^* - \lambda_{i,t+1}, \lambda_{i,t+1} - \lambda_{i,t}\rangle + \langle \lambda_i^* - \lambda_{i,t+1}, A_i(x_{j,t} - x_{i,t})\rangle \\
&= \frac{1}{2\rho}(-\|\lambda_{i,t+1} - \lambda_{i,t}\|^2 + \|\lambda_i^* - \lambda_{i,t}\|^2 - \|\lambda_i^* - \lambda_{i,t+1}\|^2) \\
&\quad + \langle \lambda_i^* - \lambda_{i,t+1}, A_i(x_{j,t} - x_{i,t})\rangle \\
&= \frac{1}{2\rho}(\|\lambda_i^* - \lambda_{i,t}\|^2 - \|\lambda_i^* - \lambda_{i,t+1}\|^2) - \frac{\rho}{2}\|r_i(x_{i,t}, y_{i,t})\|^2 \\
&\quad + \langle \lambda_i^* - \lambda_{i,t+1}, A_i(x_{j,t} - x_{i,t})\rangle\,. \qquad (4.21)
\end{aligned}$$

Resolving the telescoping sum

$$\sum_{t=1}^{T} \|\lambda_i^* - \lambda_{i,t}\|^2 - \|\lambda_i^* - \lambda_{i,t+1}\|^2\,,$$

using the fact $\lambda_{i,1} = 0$, it now follows that

$$\begin{aligned}
&\sum_{t=1}^{T} \langle \lambda_i^* - \lambda_{i,t+1}, r_i(x_{i,t}, y_{i,t})\rangle \\
&\leq \frac{1}{2\rho}(\|\lambda_i^*\|^2 - \|\lambda_i^* - \lambda_{i,T+1}\|^2) - \frac{\rho}{2}\sum_{t=1}^{T}\|r_i(x_{i,t}, y_{i,t})\|^2 \\
&\leq \frac{1}{2\rho}(2\,\|\lambda_i^*\|\,\|\lambda_{i,T+1}\|) - \frac{\rho}{2}\sum_{t=1}^{T}\|r_i(x_{i,t}, y_{i,t})\|^2\,.
\end{aligned}$$

Applying (4.16) in conjunction with the assumption $\|\lambda_i^*\| \leq D_\lambda$,

$$\begin{aligned}
&\sum_{t=1}^{T} \langle \lambda_i^* - \lambda_{i,t+1}, r_i(x_{i,t}, y_{i,t})\rangle \\
&\leq \frac{D_\lambda \zeta_i}{\rho\sigma_1(A_i)} - \frac{\rho}{2}\sum_{t=1}^{T}\|r_i(x_{i,t}, y_{i,t})\|^2\,. \qquad (4.22)
\end{aligned}$$

[6]Namely, using the identity $\langle v_1 - v_2, v_3 + v_4\rangle = \frac{1}{2}(||v_4 - v_2||^2 - ||v_4 - v_1||^2 + ||v_3 + v_1||^2 - ||v_3 + v_2||^2$.

The last term in (4.21) can also be bounded as

$$\begin{aligned}
&\langle \lambda_i^* - \lambda_{i,t+1}, A_i(x_{j,t} - x_{i,t})\rangle \\
&\leq (D_\lambda \sigma_1(A_i) + \zeta_i)\|x_{j,t} - \theta_t\|\|x_{i,t} - \theta_t\| \\
&\leq 2\alpha_{t-1}\mathcal{Q}(L_f + \zeta^{\max})(D_\lambda \sigma_1(A_i) + \zeta_i) \qquad (4.23)
\end{aligned}$$

Substituting (4.20), (4.22), and (4.23) into (4.18),

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\langle \lambda_{i,t+1}, A_i\left(x^* - \theta_t\right)\rangle \leq J_1 \\
&- \frac{1}{n}\sum_{i,t}\left\{h_{1i}^{\Delta}(w_t(x_{j,t}), w^*) + \frac{\rho}{2}||r_i(x_{i,t}, y_{i,t})||^2\right\} \\
&+ 4k\sqrt{T}\mathcal{Q}(L_f + \zeta^{\max})(\frac{1}{n}\sum_{i=1}^{n} D_\lambda \sigma_1(A_i) + \overline{\zeta}). \qquad (4.24)
\end{aligned}$$

Applying $\psi\left(x^*\right) \leq \Psi^2$, $\alpha_T = k/\sqrt{T}$, and substituting (4.17), (4.24) into (4.14) and simplifying, the sub-optimality of the primal problem at the global decision θ_t can now be represented as

$$\begin{aligned}
&\sum_{t=1}^{T}\langle g_t, \theta_t - x^*\rangle \leq J_1 \qquad (4.25) \\
&- \frac{1}{n}\sum_{i,t}\left\{h_{1i}^{\Delta}(w_t(x_{j,t}), w^*) + \frac{\rho}{2}||r_i(x_{i,t}, y_{i,t})||^2\right\} \\
&+ k\sqrt{T} \times (2\mathcal{Q}(L_f + \zeta^{\max})(\frac{2}{n}\sum_{i}(D_\lambda \sigma_1(A_i) + 2\zeta_i)). \qquad (4.26)
\end{aligned}$$

Based on our assumption of convexity of $\phi_i(\cdot)$, we have

$$\begin{aligned}
\phi_i^{\Delta}(w_{i,t}, w^*) &\leq \langle \nabla_y \phi_i(y_{i,t}), y_{i,t} - y_i^*\rangle \\
&\leq -\left\langle B_i^T \lambda_{i,t+1}, y_{i,t} - y_i^*\right\rangle \\
&= -h_{2i}^{\Delta}(w_t(x_{j,t}), w^*). \qquad (4.27)
\end{aligned}$$

Combining (4.26) and (4.13) into (4.10) and adding (4.27), the regret can thereby be bounded as

$$R_{j,T}(w^*) \leq J_1 + J_2 k\sqrt{T}$$

for all $j \in [n]$ and thus the social regret is bounded as $R_T \leq J_1 + J_2 k\sqrt{T}$. Note that the social regret represents the worst case regret amongst the agents in the network. □

The above theorem validates the "good" performance of OD-ADMM via dual averaging by demonstrating a sub-linear social regret. In addition, this social regret highlights the importance of the underlying interaction topology through $\sigma_2(P)$ and certain condition measure of local linear constraints through $\sigma_1(A_i)$ and $\sigma_{m_i}(B_i)$. A well known measure of network connectivity is the second smallest eigenvalue of the graph Laplacian $L(\mathcal{G})$ denoted by $\Lambda_2(\mathcal{G})$. Since the communication matrix P is formed as proposed in Proposition 19, $1 - \sigma_2(P)$ is proportional to $\Lambda_2(\mathcal{G})$ implying that high network connectivity promotes good performance of the proposed OD-ADMM algorithm with the embedded distributed implementation of dual averaging.

4.2. OD-ADMM via Distributed Gradient Descent. In this section, the local primal variable is updated using distributed GD method. In this method $x_{i,t}$ is updated as a convex combination of its neighbor's local primal variables and itself, moving in the direction of decreasing the Lagrangian function,

$$h_{i,t+1} = \sum_{j=1}^{n} P_{ji} x_{j,t} - \alpha_t (g_{i,t} + A_i^T \lambda_{i,t+1}), \tag{4.28}$$

followed by the projection onto the convex set χ,

$$x_{i,t+1} = \prod_{\chi} h_{i,t+1}. \tag{4.29}$$

In the distributed GD method, we first define the (network) average primal variable as

$$\theta_t = \frac{1}{n} \sum_{i=1}^{n} x_{i,t}. \tag{4.30}$$

The regret analysis for the OD-ADMM via the GD method can now be presented as follows, where the intermediate results required for its proof are relegated to the Appendix, namely Proposition 20, and Lemmas 23 and 25. In particular, with a proper choice of the learning rate in Lemmas 23 and 25, we can show that the network effect and sub-optimality of the average decision are sub-linear over time. Then, building on these results, a sub-linear social regret bound for OD-ADMM using distributed GD method can be obtained; this is formalized in the following theorem.

THEOREM 18. *Given the sequence $w_{i,t}$ generated by Algorithm 5, where line 8 applies the distributed GD method with $\alpha(t) = k/\sqrt{t}$, we have*

$$R_T \leq J_1 + J_2 k\sqrt{T}, \tag{4.31}$$

where

$$J_1 = \frac{D_\lambda}{\rho n}\sum_{i=1}^{n}\frac{\zeta_i}{\sigma_1(A_i)} + \frac{D_\chi^2}{2k},$$

$$J_2 = 4\mathcal{Q}(L_f + \zeta^{\max})(\frac{1}{n}\sum_{i=1}^{n} D_\lambda \sigma_1(A_i) + 2\overline{\zeta}) + 2(L_f + \overline{\zeta})^2 + 8L_f\mathcal{Q}(L_f + \overline{\zeta}),$$

with

$$\overline{\zeta} = \frac{1}{n}\sum_{i=1}^{n}\zeta_i,\ \zeta^{\max} = \max_i \zeta_i,$$

$$\zeta_i = \sqrt{m_i}L_\phi\frac{\sigma_1(A_i)}{\sigma_{m_i}(B_i^T)},\ \textit{and}\ \mathcal{Q} = \frac{\sqrt{n}}{1-\sigma_2(P)}.$$

PROOF. Based on the definition of f_t we have

$$f_t^\Delta(w_t(x_{j,t}), w^*) = \frac{1}{n}\sum_{i=1}^{n} f_{i,t}^\Delta(w_t(x_{j,t}), w^*),$$

where $w_t(x_{j,t}) = (x_{j,t}, y_t, \lambda_{t+1}) \in \Omega$ and thus

$$R_{j,T}(w^*) = \frac{1}{n}\sum_{i,t} f_{i,t}^{\Delta}(w_t(x_{j,t}), w^*) + \phi_i^{\Delta}(w_t(x_{j,t}), w^*)$$
$$+H_i^{\Delta}(w_t(x_{j,t}), w^*) + \frac{\rho}{2}||r_i(x_{i,t}, y_{i,t})||^2.$$

As f_t is L-Lipschitz and convex, we have

$$f_t^{\Delta}(w_t(x_{j,t}), w^*) = f_t(x_{j,t}) - f_t(\theta_t) + f_t(\theta_t) - f_t(x^*)$$
$$\leq L_f \left\|x_{j,t} - \theta_t\right\| + \left\langle g_t, \theta_t - x^*\right\rangle. \tag{4.32}$$

The first term in (4.32) represents the network effect in the regret bound, i.e., the deviation of local primal variable at each node from the average primal variable. In the meantime, Lemmas 22 and 23 in the Appendix provide bounds on the network effect when the DA and GD methods are used in line 8 of Algorithm 5, respectively. Therefore, replacing line 8 of Algorithm 5 with the distributed DA method implies

$$\|\theta_t - x_{j,t}\| \leq \alpha_{t-1}\frac{\sqrt{n}(L_f + \zeta^{\max})}{1 - \sigma_2(P)}, \tag{4.33}$$

and with the distributed GD,

$$\|\theta_t - x_{j,t}\| \leq 2\sqrt{n}(L_f + \zeta^{\max})\sum_{k=1}^{t-1}\alpha_{t-k}\sigma_2(P)^{k-1}. \tag{4.34}$$

Moreover, from (4.33) (and (4.12)) it follows that

$$\sum_{t=1}^{T}\|x_{j,t} - \theta_t\| \leq 4k\sqrt{T}(\mathcal{Q}(L_f + \zeta^{\max}). \tag{4.35}$$

Note that the upper bound in (4.35) is more conservative than the distributed DA method by a factor of 2.

The second term in (4.32) represents the sub-optimality due to using sub-gradient method. Applying Lemma 25 with (4.28)-(4.30) then leads to

$$\begin{aligned}\sum_{t=1}^{T}\langle g_t,\theta_t-x^*\rangle \leq \frac{2}{n^2}\sum_{t=1}^{T}\alpha_t(\sum_{i=1}^{n}||g_{i,t}+A_j^T\lambda_{i,t+1}||)^2\\ +(\sum_{t=1}^{T}(4L_f\sum_{k=0}^{t-1}\alpha_{t-k}\sigma_2(P)^k)\\ \times\frac{1}{n}\sum_{i=1}^{n}||(g_{i,t}+A_i^T\lambda_{i,t+1}||)\\ +\frac{1}{n}\sum_{t=1}^{T}\sum_{i=1}^{n}\left\langle A_i^T\lambda_{i,t+1},x^*-\theta_t\right\rangle+\frac{1}{2\alpha_1}D_\chi^2. \end{aligned} \tag{4.36}$$

The first term on the right hand side of (4.36) is bounded as

$$\begin{aligned}&\frac{2}{n^2}\sum_{t=1}^{T}\alpha_t(\sum_{i=1}^{n}||g_{i,t}+A_i^T\lambda_{i,t+1}||)^2\\ &\leq 2(\frac{1}{n}\sum_{i=1}^{n}\max_t\|g_{t+1}\|+\frac{1}{n}\sum_{i=1}^{n}\sigma_1(A_i)\max_t\|\lambda_{i,t}\|)^2\sum_{t=1}^{T}\sum_{t=1}^{T}\alpha_t\\ &\leq 2(L_f+\bar{\zeta})^2k\sqrt{T}. \end{aligned} \tag{4.37}$$

Similarly, the second term on the right hand side of (4.36) is bounded as

$$\begin{aligned}&\sum_{t=1}^{T}(4L_f\sum_{k=0}^{t-1}\alpha_{t-k}\sigma_2(P)^k)\frac{1}{n}\sum_{i=1}^{n}||(g_{i,t}+A_i^T\lambda_{i,t+1}||\\ &\leq 8L_fk\sqrt{T}\mathcal{Q}(L_f+\bar{\zeta}). \end{aligned} \tag{4.38}$$

We now proceed to bound the third term in (4.36) and from (4.18) we have

$$\begin{aligned}\langle\lambda_{i,t+1},A_i(x^*-\theta_t)\rangle=-h_{1i}^{\Delta}(w_t(x_{j,t}),w^*)\\ +\langle\lambda_{i,t+1},A_i(x_{j,t}-\theta_t)\rangle+\langle\lambda_i^*-\lambda_{i,t+1},r_i(x_{j,t},y_{i,t})\rangle. \end{aligned} \tag{4.39}$$

Analogous to the proof of Theorem 17, the second term of (4.39) is bounded as

$$\sum_{t=1}^{T} \langle \lambda_{i,t+1}, A_i(x_{j,t} - \theta_t) \rangle \leq 4k\sqrt{T}\zeta_i \mathcal{Q}(L_f + \zeta^{\max}). \tag{4.40}$$

Based on (4.22) and (4.23), we have

$$\begin{aligned} &\sum_{t=1}^{T} \langle \lambda_i^* - \lambda_{i,t+1}, r_i(x_{j,t}, y_{i,t}) \rangle \\ &\leq \frac{D_\lambda \zeta_i}{\rho \sigma_1(A_i)} - \frac{\rho}{2} \sum_{t=1}^{T} \|r_i(x_{i,t}, y_{i,t})\|^2 \\ &+ 4k\sqrt{T}\mathcal{Q}(L_f + \zeta^{\max})(D_\lambda \sigma_1(A_i) + \zeta_i). \end{aligned} \tag{4.41}$$

Substituting (4.40) and (4.41) into (4.39),

$$\begin{aligned} &\frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{T} \langle \lambda_{i,t+1}, A_i \left(x^* - \theta_t\right) \rangle \\ &\leq -\frac{1}{n} \sum_{i,t} [h_{1i}^{\Delta}(w_t(x_{j,t}), w^*) + \frac{\rho}{2} ||r_i(x_{i,t}, y_{i,t})||^2] \\ &+ 4k\sqrt{T}\mathcal{Q}(L_f + \zeta^{\max})(\frac{1}{n} \sum_{i=1}^{n} D_\lambda \sigma_1(A_i) + 2\overline{\zeta}) + \frac{D_\lambda}{\rho n} \sum_{i=1}^{n} \frac{\zeta_i}{\sigma_1(A_i)}. \end{aligned} \tag{4.42}$$

From (4.37), (4.38), and (4.42), the bound on (4.36) for $\alpha_t = k/\sqrt{t}$ is simplified to

$$\begin{aligned} &\sum_{t=1}^{T} \sum_{t=1}^{T} \langle g_t, \theta_t - x^* \rangle \\ &\leq J_1 - \frac{1}{n} \sum_{i,t} [h_{1i}^{\Delta}(w_t(x_{j,t}), w^*) + \frac{\rho}{2} ||r_i(x_{i,t}, y_{i,t})||^2] \\ &+ 2k\sqrt{T}(\mathcal{Q}(L_f + \zeta^{\max})(\frac{2}{n} \sum_{i=1}^{n} D_\lambda \sigma_1(A_i) + 4\overline{\zeta}) \\ &+ (L_f + \overline{\zeta})^2 + 4L_f \mathcal{Q}(L_f + \overline{\zeta})). \end{aligned} \tag{4.43}$$

In the meantime, based on the convexity of ϕ_i,

$$\phi_i^{\Delta}(w_t, w^*) \leq -h_{2i}^{\Delta}(w_t(x_{j,t}), w^*). \tag{4.44}$$

Combining (4.43) and (4.35) into (4.32) and adding (4.44) we obtain,

$$R_{j,T}(w^*) \leq J_1 + J_2 k\sqrt{T}.$$

for all $j \in [n]$ and thus $R_T \leq J_1 + J_2 k\sqrt{T}$. □

Theorems 17 and 18 and their proofs provide a basis for comparing two effective methods for evaluating OD-ADMM. The bounds provided in (4.13) and (4.35) for the distributed DA and GD, respectively, although conservative, hint at the fact that in the distributed DA, the local copies of the global variable $x \in \chi$ might converge faster to consensus in the worst case scenario. Moreover, as discussed in the introduction, the distributed DA approach does not require sharing the primal variables $x_{i,t}$ amongst the agents, preserving their privacy during the distributed decision making process; this feature of the DA approach however is not shared by embedding the distributed GD in OD-ADMM.

5. Formation Acquisition with Points of Interest and Boundary Constraints

Consider a formation acquisition problem amongst n agents where the position of agent i, denoted as y_i, is restricted to the convex set $Y = [-1, 1]^2$. The centroid of the formation is $x \in \mathbb{R}^2$ which is similarly constrained to $\mathcal{X} = Y$. The formation shape is defined for each agent by its offset c_i from the centroid, namely $x - y_i = c_i$. There is a known boundary S which agents are required to avoid by increasing the distance to the boundary $\text{dist}(y_i, S) = \inf_{x \in S} \|x - y_i\|$. This is achieved with a penalty function $\phi_i(y_i) = (\text{dist}(y_i, S) + 1)^{-1}$ associated with agent y_i's proximity to S. Assuming that $\text{int}\,(S \bigcap \mathcal{X})$ is an empty set, $\phi_i(y_i)$ is convex. At each time step t, each agent i obtains a location of interest $q_{i,t}$ and the centroid is ideally located close to these locations of interest promoted through the minimization of the function $f_{i,t}(x) = \frac{1}{2}\|x - q_{i,t}\|_2^2$. The example illustrated in Figure 5.1 takes the form of problem (3.1), namely

$$\begin{aligned} \min_{x \in \mathcal{X}, y_1, \ldots y_n \in Y} \quad & \textstyle\sum_{t=1}^{T} \sum_{i=1}^{n} (f_{i,t}(x) + \phi_i(y_i)) \\ \text{s.t.} \quad & A_i x + B_i y_i = c_i \,\text{for all}\, i \in [n]\,, \end{aligned}$$

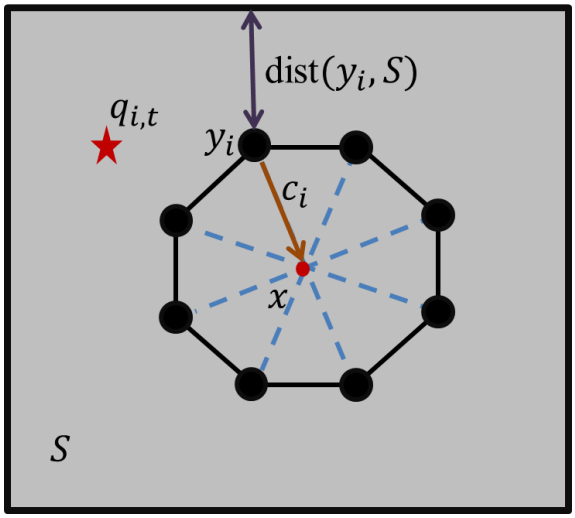


FIGURE 5.1. Formation acquisition problem amongst 6 agents.

where $A_i = -B_i = I_2$ for all $i \in [n]$.

Consider $S = \{(x, y) \in \mathbb{R} | \, |x| = 1.5, |y| = 1.5\}$ and so $\phi_i(y_i) = (2.5 - \|y_i\|_\infty)^{-1}$. The relevant parameters of the ADMM algorithm are $g_i(t) = \nabla f_{i,t}(x_i) = x_i - q_{i,t}$, $k = 2$, $\rho = 0.5$, and $\psi(x) = \|x\|_2^2$. The remaining terms of the regret bound are $L_\phi = 4/9$, $L_f = \sqrt{2}$, $\sigma_1(A_i) = \sigma_{m_i}(B_i) = 1$, $D_\lambda = 2$, and $K = 1$.

The algorithm was applied to $n = 8$ agents connected over a random graph (see Figure 5.4) with $\sigma_2(P) = 0.78$ with c_i's selected to acquire a formation with n agents equidistant apart on the circumference of a circle of radius 0.4. Locations of interest switch at each time step between a uniform distribution over the area of a length 0.5 square centered at $(-0.75, 0)$ and a Gaussian distribution with mean $(0, -0.75)$ and standard deviation $0.01 I_2$, with bounds outside of $\mathcal{X}$ ignored. The distribution is displayed in Figure 5.2. The convergence of the global variables $x_{i,t}$ to agreement as well as the reduction of the residue over time are displayed in Figure 5.3. Note that the local copies of the global variable converge faster to consensus using the distributed DA as compared with embedding the distributed GD for OD-ADMM.

The performance of the algorithm was compared for different graph topologies, namely path, star, cycle, random, cube and complete graph. These graph topologies are displayed in Figure 5.4. The matrix P was formed as proposed in Proposition 19, with $\epsilon = d_{\max}+1$, as

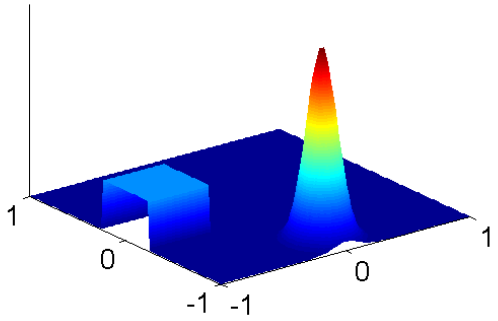


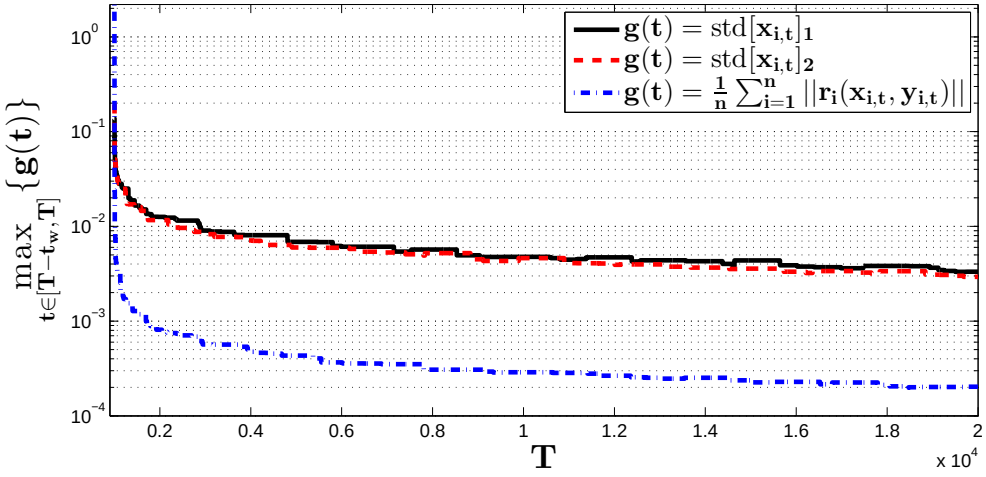
FIGURE 5.2. Distribution of locations of interest $q_{i,t}$.

such $\sigma_2(P) = 1 - \frac{1}{\epsilon}\Lambda_2(L(\mathcal{G}))$. Under the same locations of interest as described previously the performance of the regret per time R_T/T for each graph topology is compared in Figure 5.5. The performance is strongly correlated to $\sigma_2(P)$, as predicted in Theorem 17, with smaller $\sigma_2(P)$ exhibiting improved performance.

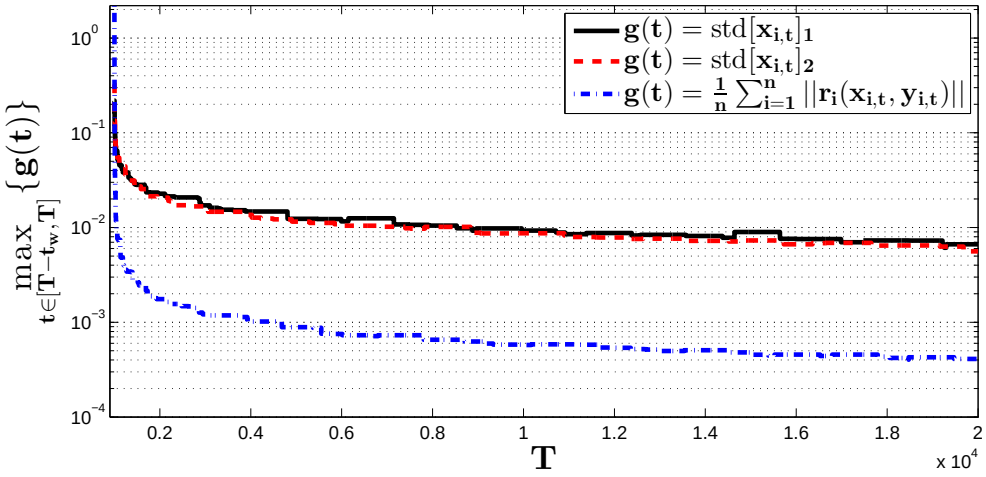
6. Remarks

In this work, online distributed ADMM has been introduced and analyzed, where a network of decision-makers or agents cooperatively optimize an objective that decomposes to global and local objectives, and is partially online. Moreover, the local variables and the global variable are linearly constraint (specific to each agent). This problem setup has a wide range of applications in networked systems, such as in distributed robotics and computer networks. A distributed algorithm allows us to make decisions across the network based on local data and information exchange with neighboring agents.

The online distributed algorithm developed in this chapter, achieves a sub-linear social regret of $O(\sqrt{T})$, that simultaneously captures sub-optimality of the objective function



(a) Distributed Dual Averaging



(b) Distributed Gradient Descent

FIGURE 5.3. The standard deviation of the global variable x_i and the average residue for each agent over times smoothed by taking the maximum over a $t_w = 1000$ sliding window.

and the violations in the linear local constraints. In particular, this algorithm is competitive with respect to the best fixed decision performance in hindsight. Moreover, we have highlighted the role of the underlying network topology in achieving a "good" social regret, i.e., the regret bound improves with increased connectivity in the network. The proposed algorithm was then applied to a formation acquisition problem.

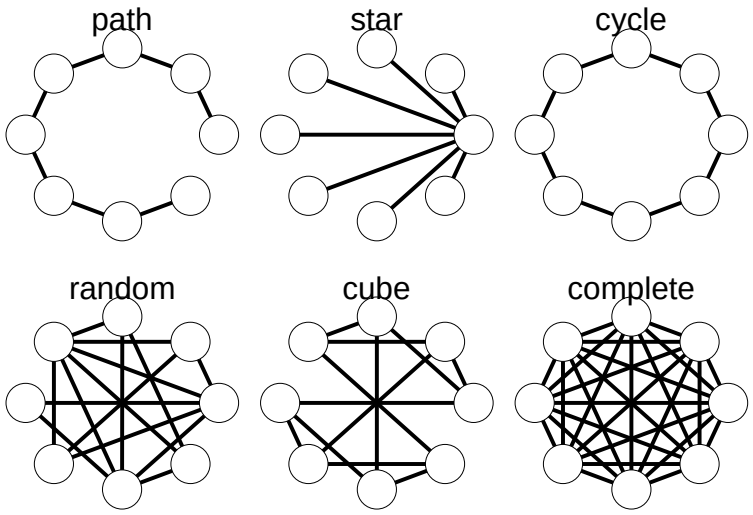


FIGURE 5.4. Topologies of the 6 different graph types.

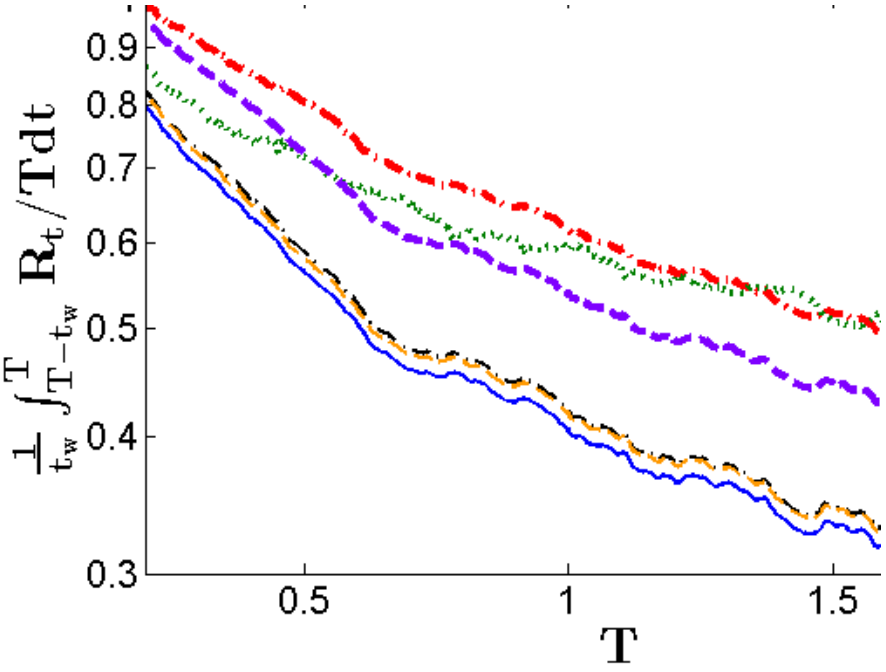


FIGURE 5.5. The regret per time R_T/T performance of 6 different graph types, specifically a path, star, cycle, random, cube and complete graph with $\sigma_2(P) = \{0.95, 0.88, 0.80, 0.78, 0.50, 0.00\}$, respectively. The trajectories are smoothed by taking the average over a $t_w = 1000$ sliding window.

Future work of particular interest includes exploring social regret over a time varying network, and investigating favorable network characteristics for the proposed online distributed ADMM algorithm.

7. Appendix

The following results can be found in **[59, 95, 114]**; as such they are presented here with no or abridged proofs.

PROPOSITION 19. *If graph $\mathcal{G}$ is strongly connected then the matrix $P = I - \frac{1}{\epsilon} diag\,(v)\, L(\mathcal{G})$ is doubly stochastic, where $v^T L\,(\mathcal{G}) = 0$ with positive vector $v = [v_1, v_2, \ldots, v_n]^T$ and $\epsilon \in (\max_{i \in V}(v_i d_i), \infty)$. If graph $\mathcal{G}$ is balanced then the matrix $P = I - \frac{1}{\epsilon} L(\mathcal{G})$ is doubly stochastic, where $\epsilon \in (d_{\max}, \infty)$.*

PROPOSITION 20. *For any $u \in \mathbb{R}^m$, $v \in \chi$, and orthogonal projection operator $\prod_\chi$ onto χ we have*

$$\left\langle u - \prod_{\chi}(u), u - v \right\rangle \geq 0.$$

LEMMA 21. *For any $u, v \in \mathbb{R}^m$, and under the conditions stated for the proximal function ψ and step size α, we have*

$$\| \prod_{\chi}^{\psi}(u, \alpha) - \prod_{\chi}^{\psi}(v, \alpha)\| \leq \alpha \|u - v\|_*.$$

LEMMA 22. *For sequences $z_{i,t}$ and $x_{i,t}$ generated by Algorithm 5, using the distributed DA method where,*

$$z_{i,t+1} = \sum_{j=1}^{n} P_{ji} z_{j,t} + g_{i,t} + A_i^T \lambda_{i,t+1}$$

and $x_{i,t+1} = \prod_{\chi}^{\psi}(z_{i,t+1}, \alpha_t)$, we have

$$\|\theta_t - x_{i,t}\|_* \leq \alpha_{t-1} \frac{\sqrt{n}(L_f + \zeta^{\max})}{1 - \sigma_2(P)}$$

for all $i \in [n]$ and $t \in [T]$, where the sequence θ_t is generated by (4.8), $\zeta_i = L_\phi \sigma_1(A_i)/\sigma_1(B_i)$, and $\zeta^{\max} = \max_i \zeta_i$.

PROOF. Based on the definition of $z_{i,t}$ we have

$$z_{i,t} = \sum_{j=1}^{n} \left[P^{t-1}\right]_{ji} z_{j,1} + \sum_{k=1}^{t-1}\sum_{j=1}^{n} \left[P^{k-1}\right]_{ji} \times \left(g_{j,t-k} + A_j^T \lambda_{j,t-k+1}\right).$$

In addition, z_t evolves as

$$z_t = z_1 + \sum_{k=1}^{t-1}\sum_{j=1}^{n} \frac{1}{n}(g_{j,t-k} + A_j^T \lambda_{j,t-k+1}). \tag{7.1}$$

Assuming $z_{i,1} = 0$ for all $i \in [n]$ and based on (7.1) we have

$$z_t - z_{i,t} = \sum_{k=1}^{t-1}\sum_{j=1}^{n} (\frac{1}{n} - \left[P^{k-1}\right]_{ji})(g_{j,t-k} + A_j^T \lambda_{j,t-k+1}) \tag{7.2}$$

Thus, the dual norm of $z_t - z_{i,t}$ can be bounded as

$$\begin{aligned} \|z_t - z_{i,t}\|_* &\leq \sum_{k=1}^{t-1}\sum_{j=1}^{n} \left\|g_{j,t-k} + A_j^T \lambda_{j,t-k+1}\right\|_* |\frac{1}{n} - \left[P^{k-1}\right]_{ji}| \\ &\leq \sum_{k=1}^{t-1} \max_j \left\|g_{j,t-k} + A_j^T \lambda_{j,t-k+1}\right\|_* \|P^{k-1} e_i - \frac{\mathbf{1}}{n}\|_1. \end{aligned} \tag{7.3}$$

Since $\|g_{i,t}\|_* \leq L_f$ and $||A_i^T \lambda_{i,t}||_* \leq \zeta_i \leq \zeta^{\max}$, the dual norm of $z_t - z_{i,t}$ is further bounded as [7]

$$\|z_t - z_{i,t}\|_* \leq \sqrt{n}(L_f + \zeta^{\max}) \sum_{k=1}^{t-1} \sigma_2(P)^{k-1}. \tag{7.4}$$

In addition, as P is a doubly stochastic matrix, $\sigma_2(P) \leq 1$ **[115]**. Thus, the inequality (7.4) is bounded as

$$\|z_t - z_{i,t}\|_* \leq \frac{\sqrt{n}(L_f + \zeta^{\max})}{1 - \sigma_2(P)},$$

Since $\theta_t = \prod_{\chi}^{\psi}(z_t, \alpha_{t-1})$ and $x_{i,t} = \prod_{\chi}^{\psi}(z_{i,t}, \alpha_{t-1})$, the statement of the lemma follows from Lemma 21. □

[7] Note that $\|P^t x - \frac{\mathbf{1}}{n}\|_1 \leq \sigma_2(P)^t \sqrt{n}$, where the vector x belongs to $\{x \in \mathbb{R}^n | x \geq 0, \sum_{i=1}^n x_i = 1\}$; this property of stochastic matrices was similarly used by Duchi *et al.* **[59]**.

LEMMA 23. *For sequences $x_{i,t}$ and $h_{i,t}$ generated by Algorithm* 5 using distributed GD method, *where*

$$h_{i,t} = \sum_{j=1}^{n} P_{ji} x_{j,t-1} - \alpha_{t-1}(g_{i,t-1} + A_i^T \lambda_{i,t})$$

and $x_{i,t} = \prod_{\chi} h_{i,t}$, we have

$$\|\theta_t - x_{i,t}\| \leq 2\sqrt{n}(L_f + \zeta^{\max}) \sum_{k=1}^{t-1} \alpha_{t-k} \sigma_2(P)^{k-1}$$

for all $i \in [n]$ and $t \in [T]$, where the sequence θ_t is generated by

$$\theta_t = \frac{1}{n} \sum_{i=1}^{n} x_{i,t},$$

$\zeta_i = L_\phi \sigma_1(A_i)/\sigma_1(B_i)$, and $\zeta^{\max} = \max_i \zeta_i$.

PROOF. Denote $r_{i,t} = x_{i,t} - h_{i,t}$; thus based on the definition of $h_{i,t}$ we have

$$x_{i,t} = h_{i,t} + r_{i,t} = \sum_{j=1}^{n} P_{ji} x_{j,t-1} - \alpha_{t-1}(g_{i,t-1} + A_i^T \lambda_{i,t}) + r_{i,t} \tag{7.5}$$

Subsequently, we can represent $x_{i,t}$ as

$$x_{i,t} = \sum_{j=1}^{n} \left[P^{t-1}\right]_{ji} x_{j,1} - \sum_{k=1}^{t-1} \sum_{j=1}^{n} \left[P^{k-1}\right]_{ji} \times (\alpha_{t-k} \left(g_{j,t-k} + A_j^T \lambda_{j,t-k+1}\right) - r_{j,t-k+1}).$$

In addition, based on (7.5), the average primal variable θ_t evolves as

$$\theta_t = \theta_1 - \sum_{k=1}^{t-1} \sum_{j=1}^{n} \frac{1}{n} (\alpha_{t-k}(g_{j,t-k} + A_j^T \lambda_{j,t-k+1}) - r_{j,t-k+1}). \tag{7.6}$$

Assuming $x_{i,1} = 0$ for all $i \in [n]$ and based on (7.6), we can represent the network effect, that is the difference between the average primal variable over the network and individual

primal variables, as

$$\theta_t - x_{i,t} = \sum_{k=1}^{t-1}\sum_{j=1}^{n}(\frac{1}{n} - \left[P^{k-1}\right]_{ji})$$

$$\times (r_{j,t-k+1} - \alpha_{t-k}(g_{j,t-k} + A_j^T \lambda_{j,t-k+1})). \tag{7.7}$$

Thus, the network effect (7.7) can be bounded as

$$\|\theta_t - x_{i,t}\| \leq \sum_{k=1}^{t-1}\sum_{j=1}^{n} \big\| r_{j,t-k+1} - \alpha_{t-k}(g_{j,t-k} + A_j^T \lambda_{j,t-k+1}) \tag{7.8}$$

$$\times \| |\frac{1}{n} - \left[P^{k-1}\right]_{ji}| \tag{7.9}$$

$$\leq \sum_{k=1}^{t-2} \max_j \big\| r_{j,t-k+1} - \alpha_{t-k}(g_{j,t-k} + A_j^T \lambda_{j,t-k+1}) \big\| \tag{7.10}$$

$$\times \|P^{k-1} e_i - \frac{1}{n}\mathbf{1}\|_1 \tag{7.11}$$

Moreover, the difference between $h_{i,t}$ and its projection onto χ is bounded as

$$||r_{i,t}|| = || \prod_{\chi} h_{i,t} - h_{i,t}|| \leq || \sum_{j=1}^{n} P_{ji} x_{j,t-1} - h_{i,t}||$$

$$\leq \alpha_{t-1} ||g_{i,t-1} + A_i^T \lambda_{i,t}||.$$

Since $\|g_{i,t}\|_* \leq L_f$ and $||A_i^T \lambda_{i,t}|| \leq \zeta_i \leq \zeta^{\max}$, the network effect is further bounded as

$$\|\theta_t - x_{i,t}\| \leq 2\sqrt{n}(L_f + \zeta^{\max}) \sum_{k=1}^{t-1} \alpha_{t-k} \sigma_2(P)^{k-1}. \tag{7.12}$$

□

LEMMA 24. *For any positive and non-increasing sequence $\alpha(t)$ and $x^* \in \chi$*

$$\sum_{t=1}^{T} \langle g_t, \theta_t - x^* \rangle \leq \sum_{t=1}^{T} \frac{1}{n} \sum_{i=1}^{n} \langle A_i^T \lambda_{i,t+1}, x^* - \theta_t \rangle$$

$$+ \frac{1}{\alpha_T}\psi(x^*) + \sum_{t=2}^{T} \frac{\alpha_{t-1}}{2} ||g_t + \frac{1}{n}\sum_{i=1}^{n} A_i^T \lambda_{i,t+1}||_*^2,$$

where the sequence θ_t is generated by (4.7)-(4.8).

PROOF. Based on Lemma 3 in **[59]**, we have

$$\sum_{t=1}^{T}\left\langle g_t+\frac{1}{n}\sum_{i=1}^{n}A_i^T\lambda_{i,t+1},\theta_t-x^*\right\rangle \\ \leq \frac{1}{\alpha_T}\psi(x^*)+\sum_{t=2}^{T}\frac{\alpha_{t-1}}{2}\,||g_t+\frac{1}{n}\sum_{i=1}^{n}A_i^T\lambda_{i,t+1}||_*^2,$$

and the statement of the lemma follows. □

LEMMA 25. *For any positive and non-increasing sequence $\alpha(t)$ and $x^*\in\chi$,*

$$\sum_{t=1}^{T}\langle g_t,\theta_t-x^*\rangle\leq\frac{1}{n}\sum_{t=1}^{T}\sum_{j=1}^{n}\left\langle A_j^T\lambda_{j,t+1},x^*-\theta_t\right\rangle \\ +\frac{2}{n^2}\sum_{t=1}^{T}\alpha_t(\sum_{j=1}^{n}||g_{j,t}+A_j^T\lambda_{j,t+1}||)^2 \\ +\frac{1}{2\alpha_1}D_\chi^2+\sum_{t=1}^{T}(4L_f\sum_{k=0}^{t-1}\alpha_{t-k}\sigma_2(P)^k) \\ \times\frac{1}{n}\sum_{j=1}^{n}||(g_{j,t}+A_j^T\lambda_{j,t+1}||,$$

where the sequence of θ_t is generated by (4.28)*,* (4.29)*, and* (4.30)*.*

PROOF. Denote $r_{i,t}=x_{i,t}-h_{i,t}$; thus based on the definition of $h_{i,t}$, we have

$$\begin{aligned}x_{i,t+1}&=h_{i,t+1}+r_{i,t+1}\\&=\sum_{j=1}^{n}P_{ji}x_{j,t}-\alpha_t(g_{i,t}+A_i^T\lambda_{i,t+1})+r_{i,t+1}.\end{aligned}\tag{7.13}$$

Subsequently, based on (4.30), the average primal variable θ_t evolves as

$$\theta_{t+1}=\theta_t-\frac{1}{n}\sum_{j=1}^{n}(\alpha_t(g_{j,t}+A_j^T\lambda_{j,t+1})-r_{j,t+1}).\tag{7.14}$$

Now, we can represent the deviation of average primal variable θ_t from x^* as

$$\begin{aligned}
&||\theta_{t+1} - x^*||^2 = ||\theta_t - x^*||^2 \\
&+ \frac{1}{n^2}||\sum_{j=1}^{n}(\alpha_t(g_{j,t} + A_j^T\lambda_{j,t+1}) - r_{j,t+1})||^2 \\
&- \frac{2\alpha_t}{n}\sum_{j=1}^{n}\left\langle(g_{j,t} + A_j^T\lambda_{j,t+1}), \theta_t - x^*\right\rangle \\
&+ \frac{2}{n}\sum_{j=1}^{n}\left\langle r_{j,t+1}, \theta_t - x^*\right\rangle. \qquad (7.15)
\end{aligned}$$

Note that

$$\begin{aligned}
&||\sum_{j=1}^{n}(\alpha_t(g_{j,t} + A_j^T\lambda_{j,t+1}) - r_{j,t+1})||^2 \\
&\leq \left\{\sum_{j=1}^{n}||\alpha_t(g_{j,t} + A_j^T\lambda_{j,t+1}|| + ||r_{j,t+1}||\right\}^2 \\
&\leq 4\alpha_t^2(\sum_{j=1}^{n}||g_{j,t} + A_j^T\lambda_{j,t+1}||)^2,
\end{aligned}$$

and from Proposition 20 we have

$$\begin{aligned}
&\langle r_{j,t+1}, \theta_t - x^*\rangle \leq \langle r_{j,t+1}, \theta_t - h_{j,t+1}\rangle \\
&+ \left\langle\prod_{\chi} h_{j,t+1} - h_{j,t+1}, h_{j,t+1} - x^*\right\rangle \\
&\leq \langle r_{j,t+1}, \theta_t - h_{j,t+1}\rangle \leq \alpha_t||(g_{j,t} + A_j^T\lambda_{j,t+1}||||\theta_t - h_{j,t+1}||.
\end{aligned}$$

Based on Lemma 8 in **[94]**, we have

$$\langle r_{j,t+1}, \theta_t - x^*\rangle \leq 4L_f\alpha_t||(g_{j,t} + A_j^T\lambda_{j,t+1}||\sum_{k=0}^{t-1}\alpha_{t-k}\sigma_2(P)^k.$$

Thus, by rearranging the terms in (7.15), we have

$$\begin{aligned}\sum_{t=1}^{T}\langle g_t, \theta_t - x^*\rangle &\leq \frac{1}{2\alpha_1}||\theta_1 - x^*||^2 - \frac{1}{2\alpha_{T+1}}||\theta_{T+1} - x^*||^2 \\ &+ \frac{2}{n^2}\sum_{t=1}^{T}\alpha_t(\sum_{j=1}^{n}||g_{j,t} + A_j^T\lambda_{j,t+1}||)^2 \\ &+ \frac{1}{n}\sum_{t=1}^{T}\sum_{j=1}^{n}\left\langle A_j^T\lambda_{j,t+1}, x^* - \theta_t\right\rangle \\ &+ \sum_{t=1}^{T}(4L_f\sum_{k=0}^{t-1}\alpha_{t-k}\sigma_2(P)^k)\frac{1}{n}\sum_{j=1}^{n}||(g_{j,t} + A_j^T\lambda_{j,t+1}||. \end{aligned} \tag{7.16}$$

Since the diameter of χ is bounded by D_χ, we have

$$||\theta_1 - x^*||^2 \leq D_\chi^2$$

and the statement of the lemma follows. □

CHAPTER 6

Online Optimization over Time-varying and State-Dependent Networks

1. Summary

In this chapter, we study online convex optimization problems in distributed systems operating over a time-varying information network. These problems arise in diverse application domains, including social networks, mobile sensors, and the Internet. In this problem, the network topology is not known a priori and is modeled as a state-dependent network. Moreover, the state of the nodes are affected by their neighboring nodes which is inspired by opinion dynamics. Therefore, based on the history of communication links and opinion dynamics, we provide predictions for nodes' state and network topology. A sampling approach is then applied to scale the estimation algorithm for large networks. We present regret analysis for topology estimation with limited and online observations of the underlying network. The online algorithm is then applied to Twitter political social networks demonstrating the application of the proposed setup in opinion detection.

2. Introduction

This chapter is motivated by the analysis of distributed dynamic systems operating over an information network. These systems appear in a wide range of applications, including social networks, Internet, mobile sensor networks, the genome, and brain networks [**116, 117, 118, 119, 120**]. In these multi-agent systems, agents interact according to local rules; in many cases the geometry of the network is also a function of the underlying system's states. The resulting structure is referred to as a *state dependent* network [**121**] and its topology evolves as the state of the agents changes over time. Consider a swarm of unmanned vehicles with limited range of communication. The vehicles are moving and

their pairwise communication links drops when the two corresponding vehicles are outside each other's communication range. Another example is a network of people with different political orientations. Sometimes, like-minded people tend to interact and exchange information and sometimes, ideologically-opposed individuals interact with each other through arguments [**128**]. In other words, the communication links are a function of agents opinion.

Moreover, each node of the network represents a dynamical system and these individual systems are coupled according to the network topology. In a robotics swarm, vehicles make decisions based on the information they receive from their neighboring vehicles. In social networks, the opinion of agents gets affected by many factors including the people they interact with. This phenomenon is referred to as *opinion dynamics* with many standard models proposed for self-organized dynamics in social, biological, and physical sciences [**120, 122, 123, 124, 125, 126, 127**]. Most of these models assume that the intensity of alignments increases as agents state get closer [**122, 123, 124, 125, 126**], reflecting agents tendency to align with those who think or act alike. While, another approach is Heterophily [**127**], which assumes agents interact more with those who are different rather than with those who are similar.

In this chapter, we propose a sub linear online algorithm that predicts the communication topology and the latent state of the nodes in time varying complex networks. The proposed algorithm is a variant of block coordinate descent method called Online Stochastic Randomized Block Coordinate Descent (OSR-BCD). At each round $t = 0, 1, \ldots, T$ of the OSR-BCD algorithm, we sample a sub-graph of the network to scale the algorithm to massive networks. These sub-graphs can be chosen to maximize the statistical information for efficiently finding the properties of the network topology. Therefore, the network is not required to be fully observable at each iteration. Then, each node updates its state to collaboratively predict the communication matrix of the sampled subgraph and simultaneously follow an opinion dynamics. Then, the network topology is revealed and we incur a loss due to inaccurate link predictions. We assume the underlying communication graph is modeled as a temporal matrix which is a convex function of all nodes states and provide a sub linear regret for the aforementioned framework.

Next, we identify communities in a real-world network using the OSR-BCD algorithm. This network is the political re-tweet network by more than 18,000 users collected from Twitter during the six weeks leading up to the 2010 U.S. congressional midterm elections [**128, 129**]. This problem is similar to the Netflix competition where the primary goal is to estimate the missing entries of the communication matrix using the state of the art Non-negative Matrix Factorization (NMF) methods. However, our primary goal is to estimate non-negative latent factors that model the political ideology of each node towards each party. By combining a variant of the NMF method with opinion dynamics, we achieve a scalable method for predicting the election result. In addition, we study the effect of incorporating dynamics into the proposed data driven decision making process. This analysis is based on the effect of incorporating opinion dynamics into the time-varying communication matrix estimation problem. We implement several local dynamics model and compare their estimation errors with the baseline that is purely based on the observed links and not assuming any opinion dynamics on the agents.

2.1. Related Works. Identifying opinions or state of the agents in social networks is closely related to community detection in complex networks [**130, 131**] which is a well known problem in network analysis. Community detection methods generally find the groups of related nodes with dense internal link connections and few external connections. In large real-world networks, overlapping communities refer to organizational units in social networks, scientific disciplines in citation and collaboration networks, or functional units in biochemical networks. The mixed-membership stochastic block-model (MMSB)[**132**] is a powerful probabilistic model that allows each node of a network to exhibit a mixture of communities and given an observed network, it estimates the posterior community memberships of its nodes. In this work, we take advantage of the MMSB to model state-dependent social networks where the hidden opinion of nodes is equivalent to their membership vector in MMSB. The applications of classical MMSB have been limited because its corresponding inference algorithms have not been scaled to large networks. However, in [**133**], an algorithm has been developed to scale the MMSB for large networks that were previously

out of reach for this model. At each iteration of this algorithm, a sub-graph of the network is analyzed leading to faster analysis of the network.

The aforementioned works have focused on static networks, which either represent a single time snapshot of the network being investigated or an aggregated view over time. However, most complex networks, including social, biological, and robotic networks, are time-varying, which has led researchers to consider dynamic, time-evolving networks. In the context of community detection [**134, 135, 136, 137**], dynamic networks are represented by a sequence of snapshots of the underlying interactions at discrete time steps. The latent state evolution is modeled by a stochastic dynamic system in [**135**] where a linear dynamic transformation of the mixed membership prior with logistic normal distribution is placed on all agents. Later in [**136**], the aforementioned approach was extended to modeling dynamic processes on clusters, rather than on individuals by generalizing the prior on actors to a mixture of time evolving logistic normal distributions. This mixture prior is multi-modal and captures correlations between communities and accommodates for the agents heterogeneous behaviors. Similarly, in [**137**], the state evolution follows a multivariate Gaussian random walk and a modification of the extended Kalman filter is used in the inference procedure.

However, identifying realistic overlapping communities in very large networks that are increasingly common is still an open problem. The existing algorithms for community detection over time varying networks iteratively analyze each pair of nodes. Consequently, these algorithms run in $\mathcal{O}(n^2)$, where n is the number of nodes making analyzing massive networks computationally intractable.

2.2. Contributions. The contribution of this chapter is two folds; first, we present a novel model for *state dependent* social networks which considers the effect of the *opinion dynamics* on the network topology. Second, we propose a fast novel algorithm to provide online estimation on *massive streaming* graphs through *network sampling*. Finally, we implement several traditional opinion dynamics model on a real world network to study the robustness of such a data driven approach to inaccuracies in dynamic model.

3. Problem Statement

Our goal is to find the best fixed topology, i.e., adjacency matrix A^*, in an online setting where the topology of the network is changing in an unpredictable manner. The state of the nodes is denoted by the matrix $X \in \mathbb{X}$, where $\mathbb{X} = \overbrace{\chi \times \cdots \times \chi}^{n_t}$ and $\chi \subset \mathbb{R}^k$ is a compact convex set with diameter D ($||x||_2 \leq D$ for all $x \in \chi$). Note that the dimension of state vectors is k and each column of X represents the state vector of the node associated with that column, i.e., $X_{.i} \in \chi$ is the state vector of ith node. Moreover, throughout the chapter we assume:

Assumption 1: Graph $G^{(t)}$ is state-dependent and its adjacency matrix $A^{(t)}$ is a function of nodes' state, i.e., $A^{(t)} = f_t(X)$, where $f_t(.) : \mathbb{X} \to \mathbb{R}^{n_t \times n_t}$ is an unknown time varying convex function. Furthermore, $[f_t(X)]_{ij}$ only depends on the state of nodes i and j for all $i \in [n_t]$ and $j \in [n_t]$.

Assumption 2: The function $\mathcal{D}(A^{(t)}, f_t(X))$ is a distance function that is a convex function of X. For example $\mathcal{D}(A^{(t)}, f_t(X)) = \frac{1}{2}||A^{(t)} - f_t(X)||_F^2$ is a distance function and $A^{(t)} - f_t(X) \leq 0$ is a sufficient condition for the convexity of $\mathcal{D}(A^{(t)}, f_t(X))$.

Therefore, the optimization problem can be stated as:

$$\begin{aligned} &\min_{X \in \mathbb{X}} \sum_{t=1}^{T} \mathcal{D}(A^{(t)}, f_t(X)) \\ &\text{s.t. } X_{.i} \in \chi \text{ for all } i \in [n_t]. \end{aligned} \tag{3.1}$$

Throughout this chapter, we assume that there exists $X^* \in \mathbb{X}$ that is the global optima for (3.1). Note that the cost function can be expanded as:

$$\mathcal{D}(A^{(t)}, f_t(X)) = \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} \mathcal{D}_{i,j}(A_{ij}^{(t)}, f_t(X_{.,i}, X_{.,j})),$$

which is a decomposable function and we are interested in minimizing

$$\sum_{t=1}^{T} \mathcal{D}(A^{(t)}, f_t(X)) = \sum_{t=1}^{T} \sum_{i=1}^{n_t} l_{t,i}(X)$$

under the assumptions:

Assumption 3: Cost $l_{t,i}(X) = \sum_{j=1}^{n_t} \mathcal{D}_{i,j}(A_{ij}^{(t)}, f_t(X_{.,i}, X_{.,j}))$ is L_i-Lipschitz on $\mathbb{X}$. Note that the convexity of $l_{t,i}$ can be easily deduced from the convexity of $f_t(X)$ and Assumption 2.

Moreover, we assume the nodes are affected by their neighboring nodes which is captured by the following assumption:

Assumption 4 (opinion dynamics): Each individual agent i follows the distributed dynamics;

$$X_{.i}^{(t+1)} = \sum_{j \in \{i, N_s^{(t)}(i)\}} h_i(X_{.j}^{(t)}, X_{.i}^{(0)}),$$

where each h_i is convex on $\mathbb{X}$ and upper bounded by H_i.

Therefore, we can express the modified cost function as:

$$\min_{X \in \mathbb{X}} \frac{1}{2} \sum_{t=1}^{T} \sum_{i=1}^{n_t} \left(l_{t,i}(X) + \frac{1}{1-\eta_t} ||X_{.i} - \sum_{j \in \{i, N^{(t)}(i)\}} h_i((X_{.j}^{(t)}, X_{.i}^{(0)})||_2^2 \right)$$
$$\text{s.t. } X_{.i} \in \chi \text{ for all } i \in [n_t]. \tag{3.2}$$

The size of graphs are generally large and gradient descent methods are found to be effective in solving large scale online convex optimization problems. The gradient of the cost function (3.1) can be expressed as:

$$g_t(X) \coloneqq \nabla \mathcal{D}(A^{(t)}, f_t(X)) = \begin{bmatrix} g_{1,t}(X) & \cdots & g_{n_t,t}(X) \end{bmatrix}^T \in \mathbb{R}^{n_t \times k}, \tag{3.3}$$

where

$$g_{i,t}(X) = \frac{\partial \mathcal{D}(A^{(t)}, f_t(X))}{\partial X_{.i}} = \sum_{j=1}^{n_t} \frac{\partial \mathcal{D}_{i,j}(A_{ij}^{(t)}, f_t(X_{.,i}, X_{.,j}))}{\partial X_{.i}}.$$

Note that (3.3) implies the very expensive computation of gradient for large networks, i.e., $n_t \ggg 1$. Thus, the network structure is often sampled in order to select a (tractable) subset of the nodes and edges from which we make inferences about the full network. The gradient calculated based on the sampled subset of node pairs is called stochastic gradient.

4. Network Sampling

In network sampling, we assume a subset of graph $G^{(t)}(V^{(t)}, E^{(t)})$ is sampled such that the expectation of the stochastic gradient is equal to the true gradient. the sampled graph and stochastic gradient are denoted by $G_s^{(t)}(V_s^{(t)}, E_s^{(t)})$ and $\tilde{g}_{i,t}(X, \zeta)$, respectively. Therefore, we have

$$\mathbb{E}_\zeta \left[\tilde{g}_{i,t}(X, \zeta)\right] = g_{i,t}.$$

Note that

$$\mathbb{E}_\zeta \left[-\frac{1}{p(i,j)} \frac{\partial \mathcal{D}_{i,j}(A_{ij}^{(t)}, f_t(X_{.,i}, X_{.,j}))}{\partial X_{.i}} \right] = \sum_{j=1}^{n_t} \frac{\partial \mathcal{D}_{i,j}(A_{ij}^{(t)}, f_t(X_{.,i}, X_{.,j}))}{\partial X_{.i}},$$

where $p(i,j)$ is population distribution of drawing a node pair (i,j) at random from $n_t(n_t - 1)/2$ pairs. This implies

$$\mathbb{E}_\zeta \left[-\frac{1}{|V_s^{(t)}|} \sum_{j \in V_s^{(t)}} \frac{1}{p(i,j)} \frac{\partial \mathcal{D}_{i,j}(A_{ij}^{(t)}, f_t(X_{.,i}, X_{.,j}))}{\partial X_{.i}} \right] = g_{i,t},$$

or equivalently

$$\tilde{g}_{i,t}(X, \zeta) = -\frac{1}{|V_s^{(t)}|} \sum_{j \in V_s^{(t)}} \frac{1}{p(i,j)} \frac{\partial \mathcal{D}_{i,j}(A_{ij}^{(t)}, f_t(X_{.,i}, X_{.,j}))}{\partial X_{.i}}.$$

The population distribution depends on the sampling method and there are many strategies that are based on two basic steps; node sampling and edge sampling. These steps are used in the following sub-sampling methods.

4.1. Node Sampling. In node sampling, a node is chosen uniformly at random from $V^{(t)}$ and thus each node is sampled with a probability of $1/n_t$. Then, the sampled subgraph $G_s^{(t)}$ is constructed to be the induced subgraph over the sampled node set $V_s^{(t)}$. Each pair $(i,j) \in\in E_s^{(t)}$ is associated with both nodes i and j which implies $p(i,j) = 2/n_t$.

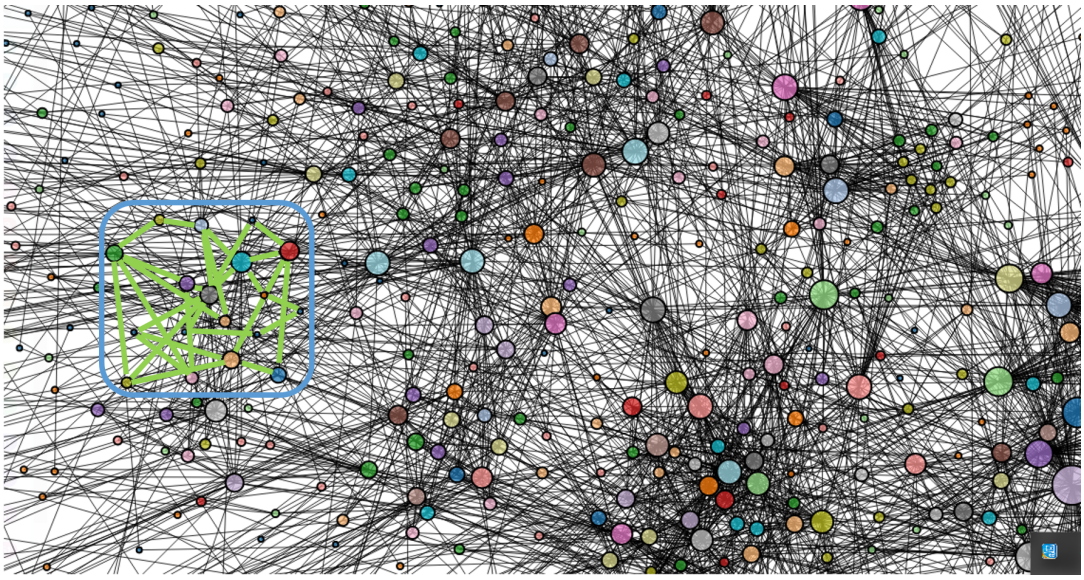


FIGURE 4.1. Sub-sampling the network.

4.2. Pair Sampling. Node pairs $(i, j) \in V_s^{(t)} \times V_s^{(t)}$ are chosen independently and uniformly at random from $G^{(t)}$ for inclusion in the sampled graph $G_s^{(t)}$. In this method the node pairs are sampled with a probability $p(i, j) = \frac{2}{n_t(n_t-1)}$.

4.3. Edge Sampling. In classic edge sampling, the sampled graph $G_s^{(t)}$ is constructed with edges that are chosen independently and uniformly at random from $E^{(t)}$. Since, real-world networks are often sparse, with links accounting for less than 0.1 of all node pairs, it may help to focus on the selection of edges rather than nodes to populate the sample. Sampling nodes set $V_s^{(t)}$ is constructed by including both incident nodes on the sampled edges. The resulting subgraph is partially induced, which means $E_s^{(t)}$ is constructed by the edges chosen during the random edge selection process. Each pair $(i, j) \in\in E_s^{(t)}$ is associated with both nodes i and j which implies link(i, j) is sampled with a probability ‘

$$p((i, j) \in E_s^{(t)}) = \frac{1}{|E^{(t)}|}$$

and non-link(i,j) is sampled with probability

$$p((i,j) \notin E_s^{(t)} \text{ and } i \in V_s^{(t)}, j \in V_s^{(t)}) = \frac{2}{|V_s^{(t)}|(|V_s^{(t)}|-1) - 2|E_s^{(t)}|} \times P(i \in V_s^{(t)}) P(j \in V_s^{(t)})$$
$$= \frac{2}{|V_s^{(t)}|(|V_s^{(t)}|-1) - 2|E_s^{(t)}|} \times \frac{\deg_{t,i}}{2|E^{(t)}|} \times \frac{\deg_{t,j}}{2|E^{(t)}|},$$

where $\deg_{t,i}$ represent the degree of node i in an un-directed graph $G^{(t)}$.

4.4. Induced Edge Sampling. Similar to edge sampling, edges are chosen independently and uniformly at random from $E^{(t)}$ and sampling nodes set $V_s^{(t)}$ is constructed by including both incident nodes on the sampled edges. The resulting subgraph is the induced subgraph over the sampled node set $V_s^{(t)}$. Each pair $(i,j) \in\in E_s^{(t)}$ is associated with both nodes i and j which implies link(i,j) is sampled with a probability

$$p((i,j) \in E_s^{(t)}) = \frac{1}{|E^{(t)}|}$$

and non-link(i,j) is sampled with probability

$$p((i,j) \notin E^{(t)} \text{ and } i \in V_s^{(t)}, j \in V_s^{(t)}) = \frac{2}{|V_s^{(t)}|(|V_s^{(t)}|-1) - 2|E_s^{(t)}|} \times P(i \in V_s^{(t)}) P(j \in V_s^{(t)})$$
$$= \frac{2}{|V_s^{(t)}|(|V_s^{(t)}|-1) - 2|E_s^{(t)}|} \times \frac{\deg_{t,i}}{2|E^{(t)}|} \times \frac{\deg_{t,j}}{2|E^{(t)}|}.$$

Note that induced edge sampling achieves a higher $|E_s(t)|$ than edge sampling method.

5. Online Stochastic Randomized Block Coordinate Descent

Block-coordinate descent (BCD) methods and their variants have been of interests for solving problems with $\mathbb{X}$ given in the block structure form. Moreover, the nodes are approximately following the opinion dynamics.

In this section, the variants of Online Stochastic BCD (OSR-BCD) algorithms for solving (3.1), followed by their convergence analysis, are presented. Assume we have n_t blocks and at each iteration of OSR-BCD, a subset $V_s^{(t)}$ of blocks are picked according to the network sampling methods where $V_s^{(t)}$ is the nodes set of sub-sampled graph. Then, the state of each block in $V_s^{(t)}$ is updated based on gradient descent:

(5.1)
$$Y_{.i}^{(t+1)} = \arg\min_{x\in\mathbb{R}^k} \left\langle \tilde{g}_{i,t}(X^{(t)},\zeta), x \right\rangle + \frac{1}{2(1-\eta_t)}||x-\tilde{h}_{i,t}(X^{(t)},\zeta)||_2^2 + \frac{1}{2\eta_t}||x-X_{.i}^{(t)}||_2^2,$$

(5.2)
$$X_{.i}^{(t+1)} = \arg\min_{x\in\mathbb{X}} \mathcal{B}_\phi\left(x, Y_{.i}^{(t+1)}\right)$$

where $D(u) = \frac{1}{2}||u||_2^2$, $\eta_t \in (0,1)$ is a non-increasing step size,

$$\tilde{g}_{i,t}(X,\zeta) = -\frac{1}{|V_s^{(t)}|}\sum_{j\in V_s^{(t)}} \frac{1}{p(i,j)} \frac{\partial \mathcal{D}_{i,j}(A_{ij}^{(t)}, f_t(X_{.,i}, X_{.,j}))}{\partial X_{.i}},$$

$$\tilde{h}_{i,t}(X^{(t)},\zeta) = \frac{1}{|N_s^{(t)}(i)|}\sum_{j\in N_s^{(t)}(i)} \frac{1}{p(i,j)} h((X_{.j}^{(t)}, X_{.i}^{(0)}),$$

and $p(i,j)$ is the population distribution of drawing a node pair (i,j) at random from all possible pairs. Note that, (5.1) captures the notion that each node wants to approximately follow the opinion dynamics but it also wants to move in a direction that decreases the loss of our hypothesis on the most recently seen example, where the loss is approximated by a linear function.

The OSR-BCD is presented in Algorithm 6.

Algorithm 6: Online Stochastic Randomized Block Coordinate Descent (OSR-BCD)

Input: $\{\eta_t\}_{t=1}^T$
Initialize $X_{.i}^{(0)}$ randomly for $\forall i = 1,...,n$
for $t = 1$ **to** T **do**
  $f_t(X)$ and $A^{(t)}$ are revealed
  Sample subgraph $G_s^{(t)}(V_s^{(t)}, E_s^{(t)})$
  foreach *Agent* $i \in V_s^{(t)}$ **do**
    Compute stochastic subgradient $\tilde{g}_{i,t}(X^{(t)},\zeta) \in \partial f_t(X^{(t)})/\partial X_i$ and opinion dynamics $\tilde{h}_{i,t}(X^{(t)},\zeta)$
    $Y_{.i}^{(t+1)} = X_{.i}^{(t)} - \eta_t(1-\eta_t)\left(\frac{1}{1-\eta_t}(X_{.i}^{(t)} - \tilde{h}_{i,t}(X^{(t)},\zeta)) + \tilde{g}_{i,t}(X^{(t)},\zeta)\right)$
    $X_{.i}^{(t+1)} = \arg\min_{x\in\mathbb{X}} \mathcal{B}_\phi(x, Y_{.i}^{(t+1)})$
  end
end

Next, the regret analysis is presented based on the Stochastic Block Mirror Descent method [**142**] and Online Randomized Block Coordinate Descent [**143**].

5.1. Convergence Analysis. Now, the regret analysis will be presented, where the intermediate results required for its proof are presented in the following lemma.

LEMMA 26. *For any positive and non-increasing sequence η_t and $X^* \in \mathbb{X}$,*

$$\left\langle \frac{1}{1-\eta_t}\left(X_{.i}^{(t)} - \tilde{h}_{i,t}(X,\zeta)\right) + \tilde{g}_{i,t}(X,\zeta), X_{.i}^{(t)} - X_{.i}^{*}\right\rangle \leq \frac{1}{2\eta_t(1-\eta_t)}\left(||X_{.i}^{(t)} - X_{.i}^{*}||_2^2 - ||X_{.i}^{(t+1)} - X_{.i}^{*}||_2^2\right) + \frac{\eta_t}{2(1-\eta_t)}\left(H_i^2 + D^2\right) + \frac{\eta_t(1-\eta_t)}{2}L_i^2,$$

where the sequence of $X_{.i}^{(t)}$ is generated by (5.1) *and* (5.2).

PROOF. Based on (5.1), we have

$$Y_{.i}^{(t+1)} = X_{.i}^{(t)} - \eta_t(1-\eta_t)\left(\frac{1}{1-\eta_t}(X_{.i}^{(t)} - \tilde{h}_{i,t}(X^{(t)},\zeta)) + \tilde{g}_{i,t}(X^{(t)},\zeta)\right)$$

which implies

$$Y_{.i}^{(t+1)} - X_{.i}^{*} = X_{.i}^{(t)} - X_{.i}^{*} - \eta_t(1-\eta_t)\left(\frac{1}{1-\eta_t}(X_{.i}^{(t)} - \tilde{h}_{i,t}(X^{(t)},\zeta)) + \tilde{g}_{i,t}(X^{(t)},\zeta)\right)$$

and

$$||Y_{.i}^{(t+1)} - X_{.i}^{*}||_2^2 = ||X_{.i}^{(t)} - X_{.i}^{*}||_2^2 + \eta_t^2(1-\eta_t)^2||\frac{1}{1-\eta_t}(X_{.i}^{(t)} - \tilde{h}_{i,t}(X^{(t)},\zeta)) + \tilde{g}_{i,t}(X^{(t)},\zeta)||_2^2 - 2\eta_t(1-\eta_t)\left\langle \frac{1}{1-\eta_t}\left(X_{.i}^{(t)} - \tilde{h}_{i,t}(X^{(t)},\zeta)\right) + \tilde{g}_{i,t}(X^{(t)},\zeta), X_{.i}^{(t)} - X_{.i}^{*}\right\rangle.$$

Note that $\mathcal{B}_\phi\left(x, Y_{.i}^{(t+1)}\right) = ||Y_{.i}^{(t+1)} - X_{.i}^*||_2^2$ if we set $\phi(x) = \frac{1}{2}||x||_2^2$. Thus based on (5.2) and the Pythagorean Theorem for Bregman divergences **[144]**, we have:

$$\left\langle \frac{1}{1-\eta_t}\left(X_{.i}^{(t)} - \tilde{h}_{i,t}(X^{(t)},\zeta)\right) + \tilde{g}_{i,t}(X^{(t)},\zeta), X_{.i}^{(t)} - X_{.i}^* \right\rangle \leq$$
$$\frac{1}{2\eta_t(1-\eta_t)}\left(||X_{.i}^{(t)} - X_{.i}^*||_2^2 - ||X_{.i}^{(t+1)} - X_{.i}^*||_2^2\right) +$$
$$\frac{\eta_t(1-\eta_t)}{2}||\frac{1}{1-\eta_t}(X_{.i}^{(t)} - \tilde{h}_{i,t}(X^{(t)},\zeta)) + \tilde{g}_{i,t}(X^{(t)},\zeta)||_2^2,$$

and the statement of the lemma follows from Assumptions 3 and 4. □

This lemma is a key building block for the proof of the convergence rate of OSR-BCD. Subsequently, building on these results, a sub-linear regret bound for OSR-BCD can be established as formalized by the following result.

THEOREM 27. *For every $X^* \in \mathbb{X}$, the sequence $X_{.i}^{(t)}$, generated by OSR-BCD described in* (5.1) *and* (5.2) *with $\eta_t = k/\sqrt{t}$ for an arbitrary $k \in (0,1)$, we have*

$$R_T \leq \mathcal{O}(\sqrt{T}).$$

PROOF. In the stochastic setting, taking the expectation over ζ and $i \in \{1,\ldots,n_t\}$, conditioned on $X^{(t)}$, we have

$$\mathbb{E}\left[\left\langle \frac{1}{1-\eta_t}\left(X_{.i}^{(t)} - \tilde{h}_{i,t}(X^{(t)},\zeta)\right) + \tilde{g}_{i,t}(X^{(t)},\zeta), X_{.i}^{(t)} - X_{.i}^* \right\rangle\right] =$$
$$\mathbb{E}_\zeta\left[\mathbb{E}_i\left[\left\langle \frac{1}{1-\eta_t}\left(X_{.i}^{(t)} - \tilde{h}_{i,t}(X^{(t)},\zeta)\right), X_{.i}^{(t)} - X_{.i}^* \right\rangle\right]\right] + \mathbb{E}_\zeta\left[\mathbb{E}_i\left[\left\langle \tilde{g}_{i,t}(X^{(t)},\zeta), X_{.i}^{(t)} - X_{.i}^* \right\rangle\right]\right],$$

where

$$\mathbb{E}_i\left[\left\langle \frac{1}{1-\eta_t}\left(X_{.i}^{(t)} - \tilde{h}_{i,t}(X^{(t)},\zeta)\right), X_{.i}^{(t)} - X_{.i}^* \right\rangle\right] =$$
$$\frac{|V_s^{(t)}|}{n_t}\sum_{i=1}^{n_t}\mathbb{E}_\zeta\left[\left\langle \frac{1}{1-\eta_t}\left(X_{.i}^{(t)} - \tilde{h}_{i,t}(X^{(t),}\zeta)\right), X_{.i}^{(t)} - X_{.i}^* \right\rangle\right] =$$
$$\frac{|V_s^{(t)}|}{n_t}\sum_{i=1}^{n_t}\left\langle \frac{1}{1-\eta_t}\left(X_{.i}^{(t)} - h_{i,t}(X^{(t)})\right), X_{.i}^{(t)} - X_{.i}^* \right\rangle,$$

and

$$\mathbb{E}_i\left[\left\langle \tilde{g}_{i,t}(X^{(t)},\zeta), X_{.i}^{(t)} - X_{.i}^{*}\right\rangle\right] =$$
$$\frac{|V_s^{(t)}|}{n_t}\sum_{i=1}^{n_t}\mathbb{E}_\zeta\left[\left\langle \tilde{g}_{i,t}(X^{(t)},\zeta), X_{.i}^{(t)} - X_{.i}^{*}\right\rangle\right] =$$
$$\frac{|V_s^{(t)}|}{n_t}\sum_{i=1}^{n_t}\left\langle g_{i,t}(X^{(t)}), X_{.i}^{(t)} - X_{.i}^{*}\right\rangle.$$

Thus, the above results and Lemma (26) imply

$$\sum_{i=1}^{n_t}\left\langle g_{i,t}(X^{(t)}) + \frac{1}{1-\eta_t}\left(X_{.i}^{(t)} - h_{i,t}(X^{(t)})\right), X_{.i}^{(t)} - X_{.i}^{*}\right\rangle \leq$$
$$\frac{n_t}{|V_s^{(t)}|}\left\{\frac{1}{2\eta_t(1-\eta_t)}\left(||X_{.i}^{(t)} - X_{.i}^{*}||_2^2 - ||X_{.i}^{(t+1)} - X_{.i}^{*}||_2^2\right) +\right.$$
$$\left.\frac{\eta_t}{2(1-\eta_t)}\left(H_i^2 + D^2\right) + \frac{\eta_t(1-\eta_t)}{2}L_i^2\right\}. \tag{5.3}$$

Since $\mathcal{D}(A^{(t)}, f_t(X))$ and $\frac{1}{2}\sum_{i=1}^{n_t}\frac{1}{1-\eta_t}||X_{.i} - \sum_{j\in\{i,N^{(t)}(i)\}}h_j((X_{.j}^{(t)}, X_{.i}^{(0)})||_2^2$ are convex, we have

$$\mathcal{D}(A^{(t)}, f_t(X)) - \mathcal{D}(A^{(t)}, f_t(X^*)) \leq \operatorname{tr}\left(\nabla\mathcal{D}(A^{(t)}, f_t(X))^T(X - X^*)\right)$$
$$\leq \sum_{i=1}^{n_t}\left\langle g_{i,t}(X^{(t)}), X_{.i}^{(t)} - X_{.i}^{*}\right\rangle,$$

and

$$\frac{1}{2(1-\eta_t)}\sum_{i=1}^{n_t}\left(||X_{.i} - \sum_{j\in\{i,N^{(t)}(i)\}}h_j((X_{.j}^{(t)}, X_{.i}^{(0)})||_2^2 - ||X_{.i}^{*} - \sum_{j\in\{i,N^{(t)}(i)\}}h_j((X_{.j}^{(t)}, X_{.i}^{(0)})||_2^2\right) \leq$$
$$\frac{1}{1-\eta_t}\sum_{i=1}^{n_t}\left\langle\left(X_{.i}^{(t)} - h_{i,t}(X^{(t)})\right), X_{.i}^{(t)} - X_{.i}^{*}\right\rangle.$$

Therefore, based on (5.3), the regret is bounded as

$$R_T \leq \sum_{t=1}^{T} \frac{n_t}{|V_s^{(t)}|} \left\{ \frac{||X_{.i}^{(t)} - X_{.i}^*||_2^2 - ||X_{.i}^{(t+1)} - X_{.i}^*||_2^2}{2\eta_t(1-\eta_t)} + \frac{\eta_t}{2(1-\eta_t)} \left(H_i^2 + D^2\right) + \frac{\eta_t(1-\eta_t)}{2} L_i^2 \right\}$$

$$\leq \frac{N}{2|V_s^{(t)}|} \left(\frac{||X_{.i}^{(1)} - X_{.i}^*||_2^2}{\eta_1(1-\eta_1)} - \frac{||X_{.i}^{(T+1)} - X_{.i}^*||_2^2}{2\eta_{T+1}(1-\eta_{T+1})} + \left(H_i^2 + D^2\right) \sum_{t=1}^{T} \frac{\eta_t}{1-\eta_t} + L_i^2 \sum_{t=1}^{T} \eta_t(1-\eta_t) \right)$$

(5.4)

$$\leq \frac{N}{2|V_s^{(t)}|} \left(\frac{D^2}{\eta_1(1-\eta_1)} + \left(H_i^2 + D^2\right) \sum_{t=1}^{T} \frac{\eta_t}{1-\eta_t} + L_i^2 \sum_{t=1}^{T} \eta_t(1-\eta_t) \right).$$

If we set $\eta_t = k/\sqrt{t}$, it follows from the integral test that

$$\sum_{t=1}^{T} \frac{\eta_t}{1-\eta_t} \leq 2k \left(k \log \left(\sqrt{T} - k \right) + \sqrt{T} \right) + \frac{k}{1-k} - 2k \left(k \log (1-k) + 1 \right),$$

$$\sum_{t=1}^{T} \eta_t(1-\eta_t) \leq k \left(2\sqrt{T} - k \log (T) \right) - k - k^2. \tag{5.5}$$

Combining (5.4) and (5.5) , the regret can thereby be bounded as:

(5.6)

$$R_T \leq \frac{N}{2|V_s^{(t)}|} \left(\frac{D^2}{k(1-k)} + 2\left(H^2 + D^2\right) k \left(k \log \left(\sqrt{T} - k \right) + \sqrt{T} \right) + L^2 k \left(2\sqrt{T} - k \log (T) \right) \right),$$

where $L = \max_{i \in [n_t]} L_i$, $H = \max_{i \in [n_t]} H_i$, and $N = \max_{t \in [T]} n_t$. □

The result of Theorem 27 highlights the effect of network sampling on regret where $\frac{|V_s|}{N}$ represents the fraction of nodes observed at each iteration.

6. Community Detection in Time-Varying Social Networks

An example of state-dependent graphs is social networks. We assume the interactions among people in social networks is presented by a time-varying graph $G^{(t)}(V^{(t)}, E^{(t)})$, where nodes represent social agents and links represent the interaction or information exchange among nodes. There are k communities or opinions and each node can be associated with multiple communities. Thus, the community membership vector for each node i is $C_i \in [0,1]^k$ which constructs columns of matrix $C \in [0,1]^{k \times n_t}$ and $\sum_{j=1}^{k} C_{ij} = 1$. For

example in the case of political elections in US, C_i can demonstrate the preference of agent i towards the two political parties and independent candidates in the form of a three dimensional probability vector.

The community interaction matrix is represented by $B \in [0,1]^{k\times k}$. This matrix represent the relationship between communities with elements providing the probability of interaction between two communities or political parties. The relationship between two communities can be observed from the cross community interactions of the people inside those groups. In other words, $C_i^T BC_j$ represents the probability of existing a link between node i and j. For example, if nodes i and j both share a similar point of view and are supporting a particular political party K, they interact with each other if the Kth diagonal element of B is close to 1. This means the inner community interaction in community K is strong. On the contrary, if the Kth diagonal element of B is small, the people with similar political opinion do not interact with each other as much as they argue with opponent parties.

Finally, the time varying adjacency matrix can be approximated as $A^{(t)} \approx C^T BC$. In other words, the adjacency matrix is state dependent and the state of each node i is its community membership vector, i.e., $C_i \in [0,1]^k$. The cost function can be presented as:

$$\begin{aligned} \min_{B,C} \sum_{t=1}^{T} ||A^{(t)} - C^T BC||_F^2 \\ s.t.\ C_i \in \Delta^{k-1} \text{ for all } i \in [n_t], \\ A^{(t)} - C^T BC \leq 0 \end{aligned}$$

where Δ^{k-1} is the standard k-simplex. Moreover, the state of the nodes are affected by the topology of the network which is the topic of opinion or consensus dynamics. A simple model for opinion dynamics is based on the Friedkin-Johnsen model and can be expressed as

$$C^{(t+1)} = \mu C^{(0)} + (I - \mu) C^{(t)} A^{(t)}. \tag{6.1}$$

An alternative model is the bounded confidence model in which the state of nodes are adjusted according to

$$C_i^{(t+1)} = C_i^{(t)} + \mu(C_j^{(t)} - C_i^{(t)}), \tag{6.2}$$

if $||(C_j - C_i)|| < d$ and $j \in \mathcal{N}_i$. In this model, d is the confidence bound and μ is a convergence rate whose value may range from 0 to 0.5. Thus we have :

$$C^{(t+1)} = C^{(t)} - \mu C^{(t)} \mathcal{L}^{(t)},$$

where $\mathcal{L}^{(t)}$ is the time varying Laplacian matrix for the network and $\mathcal{L} = D - A$.

Note that

$$\begin{aligned} ||A^{(t)} - C^T B^{(t)} C||_F^2 &= \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} |A_{ij}^{(t)} - C_i^T B^{(t)} C_j)|^2, \\ ||C - \mu C^{(0)} - (I - \mu) C^{(t)} A^{(t)}||_F^2 &= \sum_{i=1}^{n_t} ||C_i - \mu C_i^{(0)} - (1 - \mu) \sum_{j \in \mathcal{N}_i} C_j^{(t)} A_{ji}^{(t)}||_2^2 \end{aligned}$$

Therefore, we can extend the aforementioned optimization problem to the following problem where we consider the state dynamics of the node:

$$\begin{aligned} \min_{B, \hat{C}} \sum_{t=1}^{T} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} |A_{ij}^{(t)} - C_i^T B C_j)|^2 + \frac{1}{1 - \eta_t} \sum_{i=1}^{n_t} \mathcal{B}_\phi(C_i, \sum_{j \in \{i, N(i)\}} h_i(C_j^{(t)})) \\ s.t.\ C_i \in \Delta_k \text{ for all } i \in [n_t], \\ A^{(t)} - C^T B C \leq 0 \end{aligned}$$

If we set $\phi(x) = \frac{1}{2}||x||_2^2$, then

$$\mathcal{B}_\phi(C_i, \sum_{j \in \{i, N(i)\}} h_i(C_j^{(t)})) = \frac{1}{2}||C_i - \sum_{j \in \{i, N(i)\}} h_i(C_j^{(t)})||_2^2.$$

Since both $C^{(t)}$ and $B^{(t)}$ are not known, the cost function is not convex. Moreover, solving the aforementioned optimization problem is NP hard. One approach to solve this problem is alternating least squares in which the algorithm alternate between minimizing

the cost function with respect to C and B and it converges to a local optima. In this approach we can consider the cost function a time-varying convex function over C, in which $B^{(t)}$ is not known ahead of time. Therefore, we can either solve for $B^{(t)}$ at each time step or assume it will be revealed after we commit to a decision.

Note that at each iteration, we sample a subset of nodes and edges and matrix $A^{(t)}$ is not completely known ahead of time. Therefore, similar to coordinate descent for NMF, we consider the known elements of the matrix $A^{(t)}$ in the coordinated descent steps. In other words, the corresponding gradients are as follows:

$$\tilde{g}_{i,t}(C^{(t)},\zeta) = \sum_{i,j\in V_s^{(t)}} -B^{(t)}C_j^{(t)}\left(A_{ij}^{(t)} - \left(C_i^{(t)}\right)^T B^{(t)}C_j^{(t)}\right),$$

and

$$\tilde{h}_{i,t}(C^{(t)},\zeta) = \frac{1}{|N_s^{(t)}(i)|}\sum_{j\in\{i,N_s^{(t)}(i)\}}\frac{1}{p(i,j)}h_i(C_{.j}^{(t)}),$$

Thus, the OSR-BCD steps for the above optimization problem is

$$\hat{C}_i^{(t+1)} \quad = \quad C_i^{(t)} - \eta_t(1-\eta_t)\left(\frac{1}{1-\eta_t}(C_i^{(t)} - \tilde{h}_{i,t}(C^{(t)},\zeta)) - (1-\eta_t)\tilde{g}_{i,t}(C^{(t)},\zeta)\right),$$

followed by a projection step onto the simplex Δ^{k-1}, i.e.,

$$C_{ij}^{(t+1)} = \arg\min_{x\in\mathbb{X}} \mathcal{B}_\phi(x,\hat{C}_i^{(t+1)}),$$

which is based on the linear time algorithm proposed in **[145]** and presented in Algorithm 7. Note that the prediction step for C_i can be done for sampled nodes in parallel.

Assuming $B^{(t)}$ is not observed, the community interaction matrix can be calculated as

$$B^{(t+1)} = C^{(t)}A^{(t)}\left(C^{(t)}\right)^T.$$

However, we sample a few nodes at time step t and do not observe all the elements in $A^{(t)}$. Therefore, based on stochastic gradient descent approach, matrix $B^{(t+1)}$ is estimated as follows:

Algorithm 7: Projection onto the simplex

Input: $\hat{x} \in \mathbb{R}^k$
Initialize $U = [k],\ s = 0,\ \rho = 0$
while $U \neq \emptyset$ **do**
  Pick $r \in U$ at random
  $G = \{j \in U | \hat{x}_j \geq \hat{x}_r\}$
  $L = \{j \in U | \hat{x}_j < \hat{x}_r\}$
  Calculate $\Delta\rho = |G|$ and $\Delta s = \sum_{j \in G} \hat{x}_j$
  if $s + \Delta s - (\rho + \Delta\rho)\hat{x}_r < 1$ **then**
    $s = s + \Delta s$
    $\rho = \rho + \Delta\rho$
    $U = L$
  else
    $U = G \setminus \{r\}$
  end
end
$\theta = (s-1)/\rho$
Output x such that $x_j = \max\{\hat{x}_j - \theta, 0\}$

$$\begin{aligned} \hat{B}^{(t+1)} &= B^{(t)} - \alpha_t \tilde{\nabla}_B ||A^{(t)} - \left(C^{(t)}\right)^T B C^{(t)}||_F^2, \\ B^{(t+1)} &= \arg\min_{B \in \mathcal{M}} \mathcal{B}_\phi(B, \hat{B}^{(t+1)}) \end{aligned}$$

where $\mathcal{M}$ is a convex polytope of matrices, i.e.,

$$\mathcal{M} = \{B \in \mathbb{R}^{k \times k} | 0 \leq B \leq \mathbf{1}\mathbf{1}^T,\ A^{(t)} - \left(C^{(t)}\right)^T B C^{(t)} \leq 0\},$$

$$\begin{aligned} \tilde{\nabla}_B \left(\frac{1}{2}||A^{(t)} - C^T B C||_F^2\right) = -\frac{2}{|V_s^{(t)}|(|V_s^{(t)}| - 1)} \sum_{i \in V_s^{(t)}} \sum_{j \in V_s^{(t)}} \frac{1}{p(i,j)} C_i^{(t)} \left(C_j^{(t)}\right)^T \\ \left(A_{ij}^{(t)} - \left(C_i^{(t)}\right)^T B^{(t)} C_j^{(t)}\right), \\ \mathbb{E}_\zeta \left[\tilde{\nabla}_B \left(\frac{1}{2}||A^{(t)} - C^T B C||_F^2\right)\right] = \nabla_B \left(\frac{1}{2}||A^{(t)} - C^T B C||_F^2\right), \end{aligned}$$

and α_t is a non-increasing step size.

7. Simulation Results

The goal is to identify the political ideology of more than 18,500 Twitter users towards political parties and predict the re-tweets. The Twitter network were obtained from [**128, 129**] and are undirected and unweighted. Nodes are Twitter users and edges represent whether the users have retweeted each other. The dataset includes the time-stamp of the edge. Thus, snapshots of the time-varying graph were created for each hour during the six weeks leading up to the 2010 U.S. congressional midterm elections. The aggregated graph over time has the following properties:

Edges	Density	Maximum degree	Minimum degree	Average degree	Assortativity
$61,200$	0.000358563	1000	1	6	-0.0560735

TABLE 1. Properties of the aggregated graph for Twitter network.

Note that the assortativity is a preference for a network's nodes to attach to others that are similar in terms of a node's degree. Moreover, the temporal properties of the graph as observed by the OSR-BCD is presented in Figure 7.1.

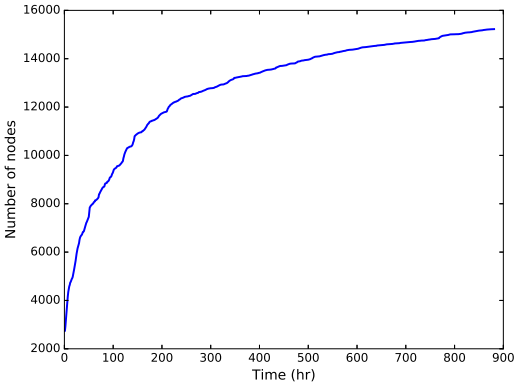
Based on the graph properties we can consider the following parameters to be implemented in the OSR-BDC and theoretical upper bound as presented in Theorem 27:

H	L	k	D	$\frac{N}{\lvert V_s^{(t)}\rvert}$
6	1000	1	2	0.25

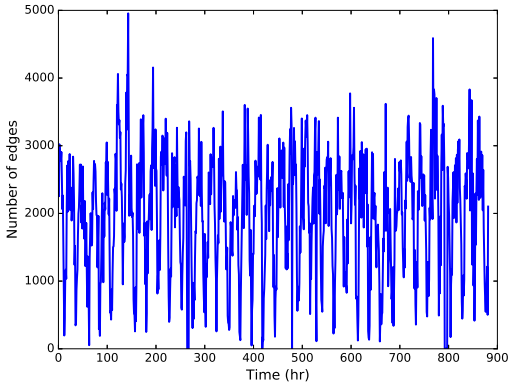
TABLE 2. Parameters of OSR-BCD algorithm.

Due to the large size of the network, a subset of 1000 nodes have been selected uniformly at random to generate the validation graph $G_v^{(t)}$ and regret and accuracy have been calculated over this graph.. In addition several opinion dynamics model were implemented and their results are presented along the theoretical bounds and the model without opinion dynamics in Figure 7.2.

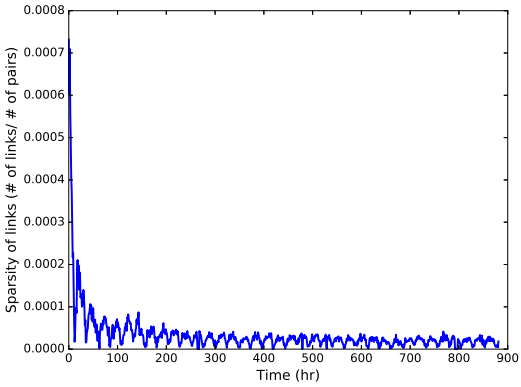
Note that regret is sub linear over time which is in agreement with our theoretical results ($R_T \leq \mathcal{O}(\sqrt{T})$). Moreover, the simple linear opinion dynamics such as (6.1) and (6.2) does not improve the performance of online estimations algorithm. However, by using



(A) Number of nodes observed by the OSR-BCD in political Twitter dataset.



(B) Number of edges observed by the OSR-BCD in political Twitter dataset.



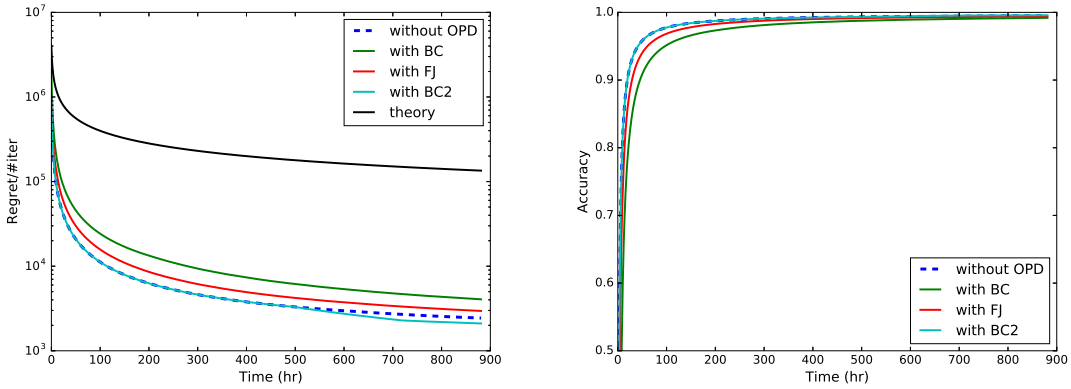
(C) Number of edges observed with respect to the number of pairs in political Twitter dataset.

FIGURE 7.1. Characteristics of the political Twitter network.

a more complex opinion dynamics we achieved a slightly superior results with respect to the scenario without considering any local dynamics.

In order to predict the result of election the number users in three communities republican, democrat and independent, has been calculated over the whole graph and presented in Figure 7.3.

Identifying the types of each community requires sentiment analysis based on the content of the retweet which is out of scope of this work. However, we can expect that



(A) Average regret over time. (B) Accuracy of predictions over time.

FIGURE 7.2. Performance of OSR-BCD

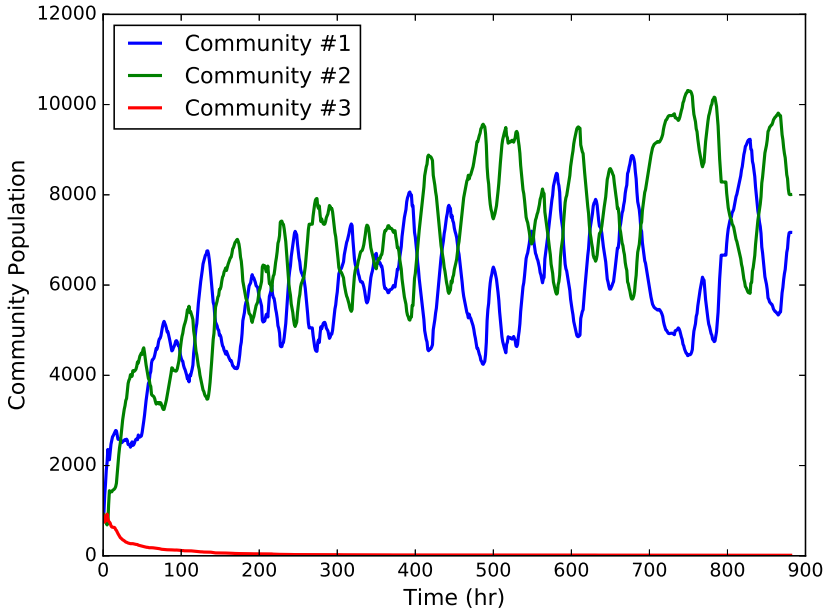


FIGURE 7.3. Communities populations over time where the communities refer to the supporters of republican, democrat, and independent parties.

community 3 corresponds to the supporters of independent politicians who are much lees the the other two political parties.

8. Remarks

The goal of this chapter is to provide a scalable method to analyze large time-varying networks. A general framework for state-dependent network was proposed and the good performance of the online estimation algorithm was demonstrated through regret analysis and empirical results. One application of such approach is opinion or community detection in online social networks where the opinion of users are changing due to their interaction with other users. The empirical results do not show a significant improvement over the case without any local dynamics, where simple linear opinion dynamics models are implemented. However, in real world the interaction among users are not linear and one direction for future research is implementing a more complex opinion dynamics model. In the extension of this work we develop a method to estimate the model parameters for opinion dynamics. In addition, the online estimation algorithm can be applied on many other large time-varying networks such as sensor networks, genomes, and brain networks. Another future direction for this work is applying this online algorithm on the aforementioned networks and other social networks.

Part 3

Final Remarks and Road Ahead

CHAPTER 7

Conclusions and Future Work

This dissertation sets out to investigate autonomous decision making in uncertain environments for single and multiple agent systems. Moreover, we examined the applications of this approach for smart energy systems, long endurance UAVs, distributed aerospace systems in uncertain environment, and social networks. In this final chapter, we will review the research contributions presented in the dissertation, and discuss directions for future work.

1. Contributions

The following are the main research contributions of this dissertation. (Some minor contributions are omitted.)

Energy management of complex engineered systems under uncertainties that improves the flight endurance of UAVs by harvesting renewable energy and an adaptive power allocation among electric components of the vehicle (Chapter 1 and Chapter 2).

Publications:

[1] Saghar Hosseini, Ran Dai, Mehran Mesbahi, "Optimal Path Planning and Power Allocation for a Long Endurance Solar-Powered UAV," In Proc. of the American Control Conference, 2013

[2] Ran Dai, Unsik Lee, Saghar Hosseini, Mehran Mesbahi, "Optimal Path Planning for Solar-Powered UAVs based on Unit Quaternions," In Proc. of IEEE Conference on Decision and Control, 2012

[3] Saghar Hosseini and Mehran Mesbahi, "Energy Aware Aerial Surveillance for a Long Endurance Solar-Powered UAV,". In Proc. of the AIAA Guidance, Navigation and Control Conference, 2013

[4] Saghar Hosseini and Mehran Mesbahi, "Energy Aware Aerial Surveillance for a Long Endurance Solar-Powered UAV,". AIAA Journal of Guidance, Control, and Dynamics, 2015 (submitted)

Moreover, the online energy management in smart cooling systems allows customers to optimize electricity cost by adequately scheduling the operations of the cooling load without an adverse effect on the entire system considering the fluctuating prices and customers preferences (Chapter 3).

[5] Saghar Hosseini, Ran Dai, and Mehran Mesbahi, " Power Management of Cooling Systems with Dynamic Pricing". In Proc. of the American Control Conference, 2014

A new distributed optimization scheme over a network of agents in the presence of cost uncertainties and over switching communication topologies was proposed. This work can be applied in the context of a range of applications such as mobile sensor networks where the network is susceptible to unknown errors, jamming, link failure, and a varying network topology (Chapter 4).

Publications:

[6] Saghar Hosseini, Airlie Chapman, and Mehran Mesbahi, "Online Distributed Optimization via Dual Averaging," In Proc. of IEEE Conference on Decision and Control, 2013

[7] Saghar Hosseini, Airlie Chapman, and Mehran Mesbahi, "Online Distributed Optimization on Dynamic Networks," IEEE Transactions on Automatic Control, 2017

A new problem set up was introduced with a network of agents, where each agent optimizes the global objective function with access to its privately known local objective functions and linear constraints. This formulation has many applications in networked systems such as robotics and computer networks (Chapter 5).

Publications:

[8] Saghar Hosseini, Airlie Chapman, and Mehran Mesbahi, "Online Distributed ADMM via Dual Averaging," In Proc. of IEEE Conference on Decision and Control, 2014

[9] Saghar Hosseini, Airlie Chapman, and Mehran Mesbahi, "Online Distributed ADMM on Networks: Social Regret, Network Effect, and Condition Measures," Automatica, 2016 (submitted)

Large state dependent networks were analyzed and an online estimation algorithms were proposed. This work has many applications including opinion detection in online social networks (Chapter 6).

Publications:

[10] Saghar Hosseini and Mehran Mesbahi, "Online Estimation on State-dependent Social Networks, " In Proc. of IEEE Conference on Decision and Control, 2016 (in preparation)

[11] Saghar Hosseini and Mehran Mesbahi, "Online Estimation of Large State-Dependent and Time-Varying Networks with Applications to Social Graphs," IEEE Transactions on Automatic Control, 2016 (in preparation)

2. Ongoing and Future Research

In my future research, I intend to continue working on the development, analysis, and implementation of distributed learning algorithms. Topology design is of critical importance for large-scale networks and cloud infrastructures represents a novel future research area in distributed computational systems. In addition, many distributed learning algorithms have been developed when operating around nominal behavior which can be well modeled and validated. Challenges arise when the communications among computing units are asynchronous and noisy. An intriguing research area would be to identify underlying communication network topologies that are robust against non-uniform operations of computing units and facilitates fast convergence of the distributed algorithm. In this direction, an examination of graph signatures of existing distributed algorithms is a logical first step. The next step is modeling of the asynchronous communications and non-homogeneous network systems such that each node represents a system with different storage and computation capabilities. The calculation of algorithms performance measures in such systems could then be used to design an adaptive communication topology in which the communication links are autonomously and adaptively scheduled.

Another exciting direction for learning in complex systems is the combination of data driven decision making and dynamics model. Machine learning methods are mainly focused on data driven decision making while the control community is mainly focused on the dynamics of the systems. These two directions are both very valuable in solving difficult problems and exploring the combination of these strong approaches represents a new future research area in dynamic systems and control theory.

Bibliography

[1] J. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[2] C. T. A. O. M. Kamgarpour, M. Soler and J. Lygeros, "Hybrid optimal control for aircraft trajectory design with a variable sequence of modes," *Proceedings of the IFAC World Congress*, pp. 7238–7243, 2011.

[3] J. Meyer, J. du Plessis, P. Ellis, and W. Clark, "Design considerations for a low altitude long endurance solar powered unmanned aerial vehicle," *AFRICON 2007*, pp. 1–7, 2007.

[4] R. Capata, L. Marino, and E. Sciubba, "A hybrid propulsion system for a high-endurance UAV: Configuration selection and aerodynamic study," *ASME 2011 International Mechanical Engineering Congress and Exposition*, vol. 11, pp. 767–774, 2011.

[5] S. Chiesa, S. Farfaglia, M. Fioriti, and N. Viola, "Design of all electric secondary power system for future advanced medium altitude long endurance unmanned aerial vehicles," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 226, no. 10, pp. 1255–1270, 2012.

[6] V. S. A. H. Eleftherios I. Amoiralis, Marina A. Tsili, "Energy efficiency optimization in uavs: a review," *Materials Science Forum*, vol. 792, pp. 281–286, 2014.

[7] J. Langelaan, "Long distance/duration trajectory optimization for small UAVs," *AIAA Guidance, Navigation and Control Conference and Exhibit*, vol. 1, pp. 1–14, 2007.

[8] Z. Akos, M. Nagy, S. Leven, and T. Vicsek, "Thermal soaring flight of birds and unmanned aerial vehicles," *Bioinspiration & Biomimetics*, vol. 5, no. 4, pp. 45003–45015, 2010.

[9] W. Kagabo and J. Kolodziej, "Trajectory determination for energy efficient autonomous soaring," *American Control Conference*, vol. 1, pp. 4655–4660, 2011.

[10] A. Chakrabarty and J. W. Langelaan, "Energy-based long-range path planning for soaring-capable unmanned aerial vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 1002–1015, 2011.

[11] T. C. Flanzer, G. C. Bower, and I. M. Kroo, "Robust trajectory optimization for dynamic soaring," *AIAA Guidance, Navigation, and Control Conference*, vol. 1, pp. 1–22, 2012.

[12] V. Bonnin and C. C. Toomer, "Energy-harvesting mechanisms for UAV flight by dynamic soaring," *AIAA Atmospheric Flight Mechanics Conference*, 2013.

[13] R. Bencatel, P. Kabamba, and A. Girard, "Perpetual dynamic soaring in linear wind shear," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 5, pp. 1712–1716, 2014.

[14] F. Fazelpour, M. Vafaeipour, O. Rahbari, and R. Shirmohammadi, "Considerable parameters of using PV cells for solar-powered aircrafts," *Renewable and Sustainable Energy Reviews*, vol. 22, pp. 81–91, 2013.

[15] A. T. Klesh and P. T. Kabamba, "Energy-optimal path planning for solar-powered aircraft in level flight," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, vol. 1, pp. 1–17, 2007.

[16] A. T. Klesh and P. T. Kabamba, "Solar-powered aircraft: Energy-optimal path planning and perpetual endurance," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 4, pp. 1320–1329, 2009.

[17] S. C. Spangelo, E. G. Gilberty, A. T. Klesh, P. T. Kabamba, and A. R. Girard, "Periodic energy-optimal path planning for solar-powered aircraft," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, vol. 1, pp. 1–15, 2009.

[18] G. Sachs, J. Lenz, and F. Holzapfel, "Unlimited endurance performance of solar UAVs with minimal or zero electric energy storage," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, vol. 1, pp. 1–13, 2009.

[19] S. Hosseini, R. Dai, and M. Mesbahi, "Optimal path planning and power allocation for a long endurance solar-powered UAV," *American Control Conference*, vol. 1, pp. 2588–2593, 2013.

[20] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero, "Cooperative large area surveillance with a team of aerial mobile robots for long endurance missions," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 70, no. 1-4, pp. 329–345, 2013.

[21] L. Doitsidis, S. Weiss, A. Renzaglia, M. W. Achtelik, E. Kosmatopoulos, R. Siegwart, and D. Scaramuzza, "Optimal surveillance coverage for teams of micro aerial vehicles in GPS-denied environments using onboard vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 173–188, 2012.

[22] R. Beard, T. McLain, D. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature uavs," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1306–1324, 2006.

[23] K. Alexis, G. Nikolakopoulos, A. Tzes, and L. Dritsas, "Coordination of helicopter uavs for aerial forest-fire surveillance," *Applications of Intelligent Control to Engineering Systems*, Springer Netherlands, vol. 39, pp. 169–193, 2009.

[24] M. Flint, M. Polycarpou, and E. Fernandez-Gaucherand, "Cooperative control for multiple autonomous UAV's searching for targets," *IEEE Conference on Decision and Control*, vol. 3, pp. 2823–2828, 2002.

[25] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. S. G. Lee, "Deployment of mobile robots with energy and timing constraints," *IEEE Transactions on Robotics*, vol. 22, no. 3, pp. 507–522, 2006.

[26] N. Nigam, S. Bieniawski, I. Kroo, J. Vian, and S. Member, "Control of multiple UAVs for persistent surveillance : Algorithm and flight test results," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1236–1251, 2012.

[27] P. Sujit and D. Ghose, "Search using multiple UAVs with flight time constraints," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 2, pp. 491–509, 2004.

[28] S. Vasisht and M. Mesbahi, "Trajectory design and coverage control for solar-powered uavs," *AIAA Guidance, Navigation, and Control Conference*, vol. 1, pp. 1–13, 2015.

[29] I. Reda and A. Andreas, "Solar position algorithm for solar radiation applications," *Solar Energy*, vol. 76, no. 5, pp. 577–589, 2004.

[30] S. J. Moura, D. S. Callaway, H. K. Fathy, and J. L. Stein, "Tradeoffs between battery energy capacity and stochastic optimal power management in plug-in hybrid electric vehicles," *Journal of Power Sources*, vol. 195, no. 9, pp. 2979–2988, 2010.

[31] The Rechargeable Battery Company, The Rechargeable Battery Company, Available: Http://sionpower.com/technology.html.

[32] R. Dai and J. E. Cochran, "Three-dimensional trajectory optimization in constrained airspace," *Journal of Aircraft*, vol. 46, no. 2, pp. 627–634, 2009.

[33] D. C. U.S. Government Printing Office Washington, "U.S. Standard Atmosphere," Tech. Rep., 1976.

[34] K. Holmstrom A.O. Goran and M. M. Edvall, *User's Guide For TOMLAB/SNOPT*, 2005. Available: http://tomlab.biz

[35] S. E. Gano, V. M. Perez, and J. E. Renaud, "Development and verification of a MATLAB driver for the SNOPT optimization software," in *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2001.

[36] H. Allcott, "Rethinking real-time electricity pricing," *Resource and Energy Economics*, vol. 33, no. 4, pp. 820–842, 2011.

[37] C. Aubin, D. Fougere, E. Husson, and M. Ivaldi, "Real-time pricing of electricity for residential customers: Econometric analysis of an experiment," *Journal of Applied Econometrics*, vol. 10, no. S1, pp. S171–S191, 1995.

[38] J. C. Laurent, G. Desaulniers, R. P. Malhamé, and F. Soumis, "A column generation method for optimal load management via control of electric water heaters," *IEEE Transactions on Power Systems*, vol. 10, no. 3, pp. 1389–1400, 1995.

[39] N. Lu, D. P. Chassin, and S. E. Widergren, "Modeling uncertainties in aggregated thermostatically controlled loads using a state queueing model," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 725–733, 2005.

[40] C. Perfumo, E. Kofman, J. H. Braslavsky, and J. K. Ward, "Load management: Model-based control of aggregate power for populations of thermostatically controlled loads," *Energy Conversion and Management*, vol. 55, pp. 36–48, 2012.

[41] S. Katipamula and L. NING, "Evaluation of residential hvac control strategies for demand response programs," *ASHRAE transactions*, pp. 535–546, 2006.

[42] D. L. Ha, F. F. De Lamotte, and Q. H. Huynh, "Real-time dynamic multilevel optimization for demand-side load management,"*IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 945–949, 2007.

[43] A. I. P. Co. (2010) Real-time pricing (rtp) for residential consumers. Available: http://www.ameren.com/sites/aiu/MyHome/IlChoiceResidential /Pages/RealTimePricing.aspx

[44] C. E. Co. (2010) The comed residential real-time pricing program, guide to real-time pricing 2010. Available: http://www.thewattspot.com/pdf/RRTPGuide200903.pdf

[45] M. A. Ortega-Vazquez and D. S. Kirschen, "Assessing the impact of wind power generation on operating costs," *IEEE Transactions on Smart Grids*, vol. 1, no. 3, pp. 295–301, 2010.

[46] D. S. Callaway, "Tapping the energy storage potential in electric loads to deliver load following and regulation, with application to wind energy," *Energy Conversion and Management*, vol. 50, no. 5, pp. 1389–1400, 2009.

[47] K. Strunz and E. Kristina Brock, "Stochastic energy source access management: infrastructure-integrative modular plant for sustainable hydrogen-electric co-generation," *International journal of hydrogen energy*, vol. 31, no. 9, pp. 1129–1141, 2006.

[48] S. Abedi, A. Alimardani, G. Gharehpetian, G. Riahy, and S. Hosseinian, "A comprehensive method for optimal power management and design of hybrid res-based autonomous energy systems," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 3, pp. 1577–1587, 2012.

[49] I. Raykhel and D. Ventura, "Real-time automatic price prediction for ebay online trading," *Innovative Applications of Artificial Intelligence Conference* , pp. 135–140, 2009.

[50] H. B. McMahan and A. Blum, "Online geometric optimization in the bandit setting against an adaptive adversary," in *Learning theory*. Springer, pp. 109–123, 2004.

[51] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," *International Conference on Machine Learning*, 2003.

[52] I. CPLEX, "11.0 user's manual," *ILOG SA, Gentilly, France*, 2007.

[53] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Cooperative convex optimization in networked systems: Augmented lagrangian algorithms with directed gossip communication," *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3889–3902, 2011.

[54] D. Mosk-Aoyama, T. Roughgarden, and D. Shah, "Fully distributed algorithms for convex optimization problems," *Distributed Computing*, vol. 4731, pp. 492–493, 2007.

[55] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, 2006.

[56] K. Chi and X. Jiang, "Topology design of network-coding-based multicast networks," *IEEE Transaction on Parallel and Distributed Systems*, vol. 19, no. 5, pp. 627–640, 2008.

[57] R. Dai and M. Mesbahi, "Optimal topology design for dynamic networks," *IEEE Conference on Decision and Control and European Control Conference*, pp. 1280–1285, 2011.

[58] I. Lobel, A. Ozdaglar, and D. Feijer, "Distributed multi-agent optimization with state-dependent communication," *Mathematical programming*, vol. 129, pp. 255–284, 2011.

[59] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, 2012.

[60] S. Lee and A. Nedic, "Distributed random projection algorithm for convex optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, pp. 221–229, 2013.

[61] S. Ram, A. Nedic, and V. Veeravalli, "Incremental stochastic subgradient algorithms for convex optimization," *SIAM Journal on Optimization*, pp. 1–30, 2009.

[62] A. Agarwal and J. Duchi, "Distributed delayed stochastic optimization," *IEEE Conference on Decision and Control*, pp. 5451–5452, 2012.

[63] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *International Conference on Machine Learning*, pp. 421–422, 2003.

[64] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *Journal of Machine Learning Research*, vol. 11, pp. 2543–2596, 2010.

[65] M. Raginsky, N. Kiarashi, and R. Willett, "Decentralized online convex programming with local information," in *American Control Conference*, pp. 5363–5369, 2011.

[66] A. Ghosh and S. Boyd, "Growing well-connected graphs," *IEEE Conference on Decision and Control*, pp. 6605–6611, 2006.

[67] D. Zelazo and M. Mesbahi, "Edge agreement: graph-theoretic performance bounds and passivity analysis," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 544–555, 2011.

[68] Y. Wan, S. Roy, and A. Saberi, "Network design problems for controlling virus spread," *IEEE Conference on Decision and Control*, pp. 3925–3932, 2007.

[69] A. Chapman and M. Mesbahi, "Semi-autonomous consensus: network measures and adaptive trees," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 19–31, 2013.

[70] A. Chapman, E. Schoof, and M. Mesbahi, "Distributed online topology design for disturbance rejection," *IEEE Conference on Decision and Control*, pp. 817–822, 2013.

[71] S. Aldosari and J. Moura, "Topology of sensor networks in distributed detection," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1061–1064, 2006.

[72] S. Kar and J. Moura, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3315–3326, 2008.

[73] A. Laszka, L. Buttyán, and D. Szeszlér, "Designing robust network topologies for wireless sensor networks in adversarial environments," *Pervasive and Mobile Computing*, vol. 9, no. 4, pp. 546–563, 2013.

[74] N. Littlestone and M. Warmuth, "The weighted majority algorithm," *Information and Computation*, vol. 108, pp. 212–261, 1994.

[75] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[76] C. W. Wu, "On bounds of extremal eigenvalues of irreducible and m-reducible matrices," *Linear Algebra and its Applications*, vol. 402, pp. 29–45, 2005.

[77] J. Anthonisse and H. Tijms, "Exponential convergence of products of stochastic matrices," *Journal of Mathematical Analysis and Applications*, vol. 59, no. 2, pp. 360–364, 1977.

[78] A. Jadbabaie and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[79] L. Xiao, S. Boyd, and S. Lall, "A space-time diffusion scheme for peer-to-peer least-squares estimation," *International Conference on Information Processing in Sensor Networks*, pp. 168–176, 2006.

[80] B. Bollobás, *Modern Graph Theory*. New York: Springer, 1998.

[81] C. Godsil and G. Royle, *Algebraic Graph Theory*. Springer, 2001.

[82] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed ADMM via dual averaging," *IEEE Conference on Decision and Control*, pp. 904–909, 2014.

[83] I. Necoara, V. Nedelcu, and I. Dumitrache, "Parallel and distributed optimization methods for estimation and control in networks," *Journal of Process Control*, vol. 21, no. 5, pp. 756–766, 2011.

[84] A. Dominguez Garcia, S. Cady, and C. Hadjicostis, "Decentralized optimal dispatch of distributed energy resources,"*IEEE Conference on Decision and Control*, pp. 3688–3693, 2012.

[85] P. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM Journal on Numerical Analysis*, vol. 16, no. 6, pp. 964–979, 1979.

[86] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," *International Conference on Machine Learning*, pp. 421–422, 2003.

[87] H. Ouyang, N. He, and A. Gray, "Stochastic ADMM for nonsmooth optimization," *arXiv preprint arXiv:1211.0632*, pp. 1–11, 2012.

[88] H. Wang and A. Banerjee, "Online alternating direction method," *International Conference on Machine Learning*, no. 1, pp. 1119–1126, 2012.

[89] T. Suzuki, "Dual averaging and proximal gradient descent for online alternating direction multiplier method," *International Conference on Machine Learning*, vol. 28, pp. 392–400, 2013.

[90] S. Boyd, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[91] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," *IEEE Conference on Decision and Control*, pp. 5445–5450, 2012.

[92] F. Iutzeler and P. Bianchi, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," *IEEE Conference on Decision and Control*, pp. 3671–3676 2013.

[93] E. Wei and A. Ozdaglar, "On the $O(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers," *IEEE Global Conference on Signal and Information Processing*, pp. 551–554, 2013.

[94] F. Yan, S. Sundaram, S. V. N. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, pp. 1041–4347, 2013.

[95] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed optimization via dual averaging," *IEEE Conference on Decision and Control*, pp. 1484–1489, 2013.

[96] A. Koppel, F. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8292–8296, 2014.

[97] S. S. Kia, J. Cortés, S. Martínez, "Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication," *Automatica*, vol. 55, pp. 254–264, 2015.

[98] P. Lin, W. Ren, Y. Song, "Distributed multi-agent optimization subject to nonidentical constraints and communication delays," *Automatica*, vol. 65, pp. 120–131, 2016.

[99] A. G. G. B. Schizas, Ioannis D. Ribeiro, "Consensus in ad hoc WSNs with noisy links - Part I: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, pp. 350–364, 2008.

[100] Z.J. Towfic, A.H. Sayed, "Stability and performance limits of adaptive primal-dual networks," *IEEE Transactions on in Signal Processing*, vol.63, no.11, pp.2888-2903, 2015.

[101] J. Mota, J. Xavier, P. Aguiar, and M. Puschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2718–2723, 2013.

[102] W. Deng, M. Lai, and W. Yin, "On the $O(1/k)$ convergence and parallelization of the alternating direction method of multipliers," *arXiv preprint arXiv:1312.3040*, pp. 1–23, 2013.

[103] B. He and X. Yuan, "On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method," *SIAM Journal on Numerical Analysis*, vol. 50, no. 2, pp. 700–709, 2012.

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

[105] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[106] D. Bertsekas, A. Nedic, and A. Ozdaglar *Convex Analysis and Optimization*. Athena Scientific, 2003.

[107] A. Nedic and A. Ozdaglar, "Subgradient methods for saddle-point problems," *Journal of Optimization Theory and Applications*, vol. 142, no. 1, pp. 205–228, 2009.

[108] T. Suzuki, "Stochastic dual coordinate ascent with alternating direction multiplier method," *International Conference on Machine Learning*, pp. 736–744, 2014.

[109] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer New York, vol. 1 2003.

[110] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Mathematical Programming*, vol. 120, no. 1, pp. 221–259, 2007.

[111] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, pp. 48–61, 2009.

[112] S. Sundhar Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, 2010.

[113] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Transactions on Automatic Control*, pp. 1291–1306, 2011.

[114] D. Bertsekas, "Incremental proximal methods for large scale convex optimization," *Mathematical Programming*, vol. 129, no. 2, pp. 163–195, 2011.

[115] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, 1979.

[116] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[117] T. Gross and B. Blasius, "Adaptive coevolutionary networks: a review," *Journal of The Royal Society Interface*, vol. 5, no. 20, pp. 259–271, 2008.

[118] M. T. Horstmann, S. Bialonski, N. Noennig, H. Mai, J. Prusseit, J. Wellmer, H. Hinrichs, and K. Lehnertz, "State dependent properties of epileptic brain networks: Comparative graph-theoretical analyses of simultaneously recorded EEG and MEG," *Clinical Neurophysiology*, vol. 121, no. 2, pp. 172–185, 2010.

[119] I. Rajapakse, M. Groudine, and M. Mesbahi, "Dynamics and control of state-dependent networks for probing genomic organization," *Proceedings of the National Academy of Sciences*, vol. 108, no. 42, pp. 17 257–17 262, 2011.

[120] D. Silvestre, P. Rosa, P. Hespanha, and C. Silvestre, "Finite-time convergence policies in state-dependent social networks," *American Control Conference*, pp. 3422–3427, 2015.

[121] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. NJ: Princeton University Press, 2010.

[122] R. Hegselmann and U. Krause, "Opinion dynamics and bounded confidence," *Simulation*, vol. 5, no. 3, 2002.

[123] V. D. Blondel, J. M. Hendrickx, and J. N. Tsitsiklis, "On Krause's multi-agent consensus model with state-dependent connectivity," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2586–2597, 2009.

[124] A. Daron Acemoglu, Ozdaglar, "Opinion dynamics and learning in social networks," in *Dynamic Games and Applications*, vol. 1, no. 1, pp. 3–49, 2011.

[125] A. Mirtabatabaei and F. Bullo, "Opinion dynamics in heterogeneous networks: Convergence conjectures and theorems," *SIAM Journal on Control and Optimization*, vol. 50, no. 5, pp. 2763–2785, 2012.

[126] W. Quattrociocchi, G. Caldarelli, and A. Scala, "Opinion dynamics on interacting networks: media competition and social influence," *Scientific Reports*, vol. 4, pp. 1–7, 2014.

[127] S. Motsch and E. Tadmor, "Heterophilious dynamics enhances consensus," *SIAM Review*, vol. 56, no. 4, pp. 577–621, 2014.

[128] M. Conover, J. Ratkiewicz, and M. Francisco, "Political polarization on twitter." *International AAAI Conference on Weblogs and Social Media*, vol. 133, no. 26, pp. 89–96, 2011.

[129] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. Available: http://networkrepository.com

[130] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: the state of the art and comparative study," *ACM Computing Surveys*, vol. 45, no. 4, pp. 1–35, 2013.

[131] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[132] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *Advances in Neural Information Processing Systems*, vol. 9, no. 2008, pp. 33–40, 2009.

[133] P. K. Gopalan and D. M. Blei, "Efficient discovery of overlapping communities in massive networks." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 110, no. 36, pp. 14 534–9, 2013.

[134] L. Tang, H. Liu, J. Zhang, and Z. Nazeri, "Community evolution in dynamic multi-mode networks," *ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 2008.

[135] E. P. Xing, W. Fu, and L. Song, *A state-space mixed membership blockmodel for dynamic network tomography*, vol. 4, no. 2, 2010.

[136] Q. Ho, L. Song, and E. P. Xing, "Evolving cluster mixed-membership blockmodel for time-varying networks," *International Conference on Artificial Intelligence and Statistics*, vol. 15, 2011.

[137] K. S. Xu and A. O. Hero, "Dynamic stochastic blockmodels for time-evolving social networks," *IEEE Journal on Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 552–562, 2014.

[138] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2011.

[139] S. Bubeck, "Introduction to online optimization," *Lecture Notes*, 2011.

[140] E. Hazan, "The Convex Optimization Approach to Regret Minimization," *Optimization for machine learning*, 2011.

[141] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, pp. 169–192, 2007.

[142] C. J. P. Newton, "Stochastic block mirror descent methods for nonsmooth and stochastic optimization," *SIAM Journal on Optimization*, vol. 37, no. 2, pp. 215–238, 2015.

[143] H. Wang and A. Banerjee, "Randomized block coordinate descent for online and stochastic optimization," *arXiv preprint*, 2014. Available: http://arxiv.org/abs/1407.0107

[144] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[145] J. C. Duchi, S. Shalev-Shwartz, and Y. Singer,T. Chandra, "Efficient projections onto the L1-ball for learning in high dimensions," *International conference on Machine learning*, pp. 272–279, 2008.